

GAME OVER



INSERT COIN

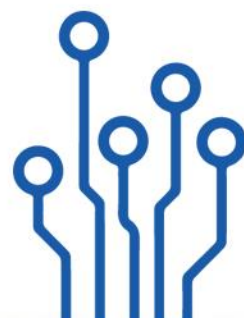


SAÉ 2

CONSOLE RETROGAMING



**Jeu
PONG**



MICHAËL BOTTIN – IUT DE RENNES – DPT GEII



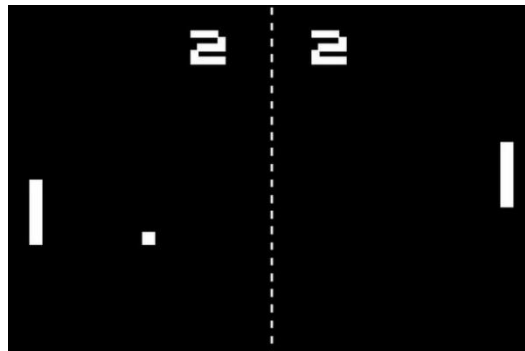
INTRODUCTION ET HISTORIQUE

Bien qu'il y ait eu quelques prémices, on considère souvent le jeu Pong comme l'ancêtre de tous les jeux vidéo. En tout cas, c'est le premier jeu vidéo à devenir populaire ! Il sort en 1972 sur borne d'arcade et il est développé par cette toute nouvelle entreprise appelée Atari.

Le principe est simple : deux joueurs s'affrontent dans un match ressemblant à du tennis en vue de dessus. Chacun dispose d'une raquette et doit renvoyer la balle d'un bout à l'autre de l'écran. Celui qui n'arrive pas à renvoyer la balle avec sa raquette perd le point.



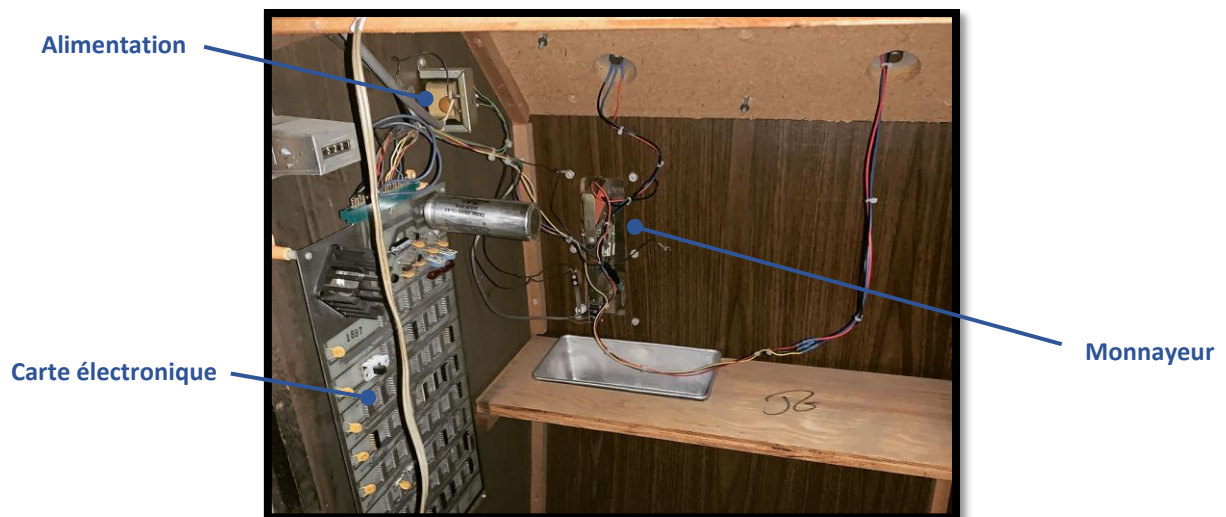
Ted Dabney (1937-2018), Nolan Bushnell (1943-) fondateurs d'Atari
Fred Marincic chef financier et Allan Alcorn (1948-) ingénieur créateur du jeu Pong.



Visuel de la forme initiale du jeu Pong de 1972

Pour information, toute la carte électronique de la borne d'arcade était réalisée avec des portes logiques. Un téléviseur (analogique bien sûr !) était branché à celle-ci :





Voici une vue rapprochée de l'intérieur de la borne



Et enfin, une vue de la carte électronique elle-même
qui comporte plus de 60 circuits intégrés logiques.

La carte faisait tout de même environ 35cm x 25cm !

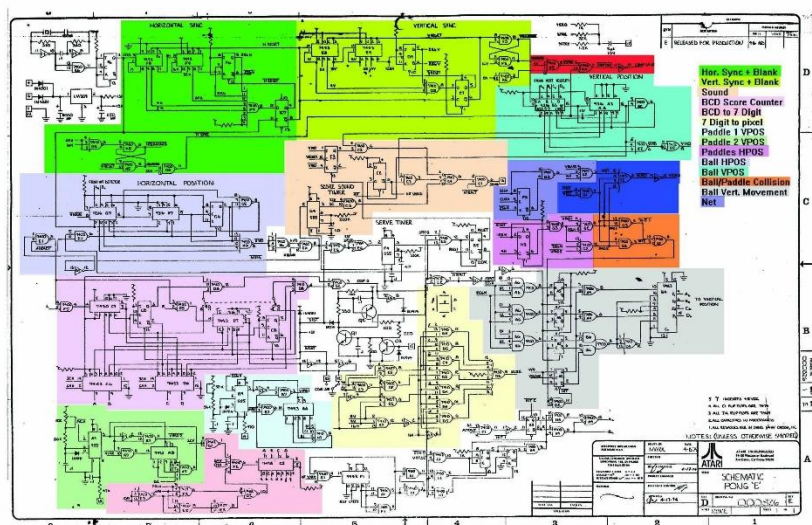


Schéma de la carte ci-dessus.

En 1975, Atari sortira sa console de salon Pong :



Cette console utilisait aussi des portes logiques et donc de la programmation câblée, ce qui faisait que ces consoles ne faisaient tourner qu'un seul jeu. Avec l'avènement des microprocesseurs dans les années 70, on verra fleurir par la suite des consoles plus souples permettant de changer de jeu grâce à des cartouches.

En tout cas, il était logique qu'en créant une mini-console de rétrogaming, on rende hommage à ces précurseurs en reproduisant en version portable sur un écran couleurs et en langage Python le jeu Pong.

CONSEIL AVANT DE DÉMARRER



Vous le savez maintenant, dans un microcontrôleur, la mémoire de stockage FLASH est précieuse car elle est souvent en quantité limitée. Il est donc important de libérer de l'espace de stockage pour ne pas être gêné par la suite :

- Faites tout d'abord une copie sécurisée (dans votre espace H :) des fichiers du jeu précédent car ils seront nécessaires **lors de l'évaluation**.
- Ensuite, supprimer les fichiers et dossiers contenus dans :
 - Le dossier 'audio'
 - Le dossier 'images'
 - Le dossier 'lib'

DÉROULEMENT ET MÉCANIQUE DU JEU 'PONG'

Voici la séquence des quatre écrans qui composent le jeu :



ÉCRAN TITRE :

- On affiche l'image de présentation du jeu.
- La guirlande lumineuse est animée
- Une courte musique accompagne cette image.
- Le bouton 'start' clignote en attendant un appui sur le bouton 'start' du joystick.
- Un son accompagne l'appui sur 'start'.



ÉCRAN DES REGLES :

- On affiche l'image du sapin.
- Les labels des règles du jeu s'affichent.
- Le bouton 'start' clignote en attendant un appui sur le bouton 'start' du joystick.
- Un son accompagne l'appui sur 'start'.



ÉCRAN DE JEU :

- Chaque raquette peut se déplacer verticalement mais pas horizontalement.
- La balle se déplace toujours en diagonale.
- Chaque joueur doit la renvoyer vers le camp adverse.
- S'il n'y parvient pas, son adversaire marque un point.
- Le premier joueur qui arrive à 10 point gagne la partie.



ÉCRAN DE FIN :

- On affiche l'image de fin.
- Le score final des deux joueurs apparaît.
- Un appui sur le bouton 'reset' relance une nouvelle partie.

RÉCUPÉRATION DES FICHIERS NÉCESSAIRES ET MISE EN PLACE

Vérifiez que vous avez assez de place mémoire pour copier les nouveaux fichiers. Vérifier pour cela que les dossiers '**audio**', '**images**' et '**lib**' soient vides.

Téléchargez ensuite le fichier '**Fichiers_jeu_pong.zip**' disponible sur Moodle. Dézippez-le et comme on l'a vu dans le tutoriel, copiez l'intégralité des fichiers et dossiers vers le lecteur de disque '**CIRCUITPY**' correspondant à votre module Raspberry Pico 2.

Des fichiers image, des fichiers audios et des bibliothèques sont maintenant disponibles pour l'écriture de votre code CircuitPython. Nous allons les parcourir dans la suite.

UTILISATION DES DIFFÉRENTES RESSOURCES DANS LE CODE

La démarche est et restera toujours la même tout au long de cette SAÉ pour accéder aux différentes ressources du jeu.

Tout d'abord, dans votre code, vous devez importer la bibliothèque des ressources :

Import Ressources_pong

Ensuite, vous devez instancier (une unique fois !) la classe d'objets qu'elle contient :

Pong = Ressources_pong.pong()

Une fois cet objet 'Pong' créé, vous pouvez accéder à n'importe quel élément graphique ou sonore grâce à cet objet.

Exemples :

- **Pong.bouton_start** permet d'accéder à l'image du bouton 'start'
- **Pong.bruitage_intro** permet d'accéder à l'audio dédié à l'écran de titre
- (...)

RESSOURCES DE L'ÉCRAN TITRE



- 'Pong.ecran_titre_christmas' :



Image de taille 160x128 pixels qui servira de fond lors de l'écran d'introduction du jeu.

- 'Pong.bouton_start' :



Image d'un bouton 'start' de 80x18 pixels qui doit clignoter pour indiquer au joueur qu'il doit appuyer sur le bouton 'start' du joystick.

- 'Pong.bruitage_intro' : petite musique d'introduction qui doit être jouée dès l'affichage de l'écran de titre.
- 'Pong.guirlande' : image d'une guirlande lumineuse de 160x34 pixels. Elle occupe toute la largeur de l'écran. C'est aussi une feuille d'animation qui comporte deux images-clés. Le changement entre ces deux images donne l'illusion d'une animation lumineuse de la guirlande.



Image
ressource



Index
d'animation

RESSOURCES DE L'ÉCRAN DES REGLES



- 'Pong.bouton_start' :



Image d'un bouton 'start' de 80x18 pixels qui doit clignoter pour indiquer au joueur qu'il doit appuyer sur le bouton 'start' du joystick.

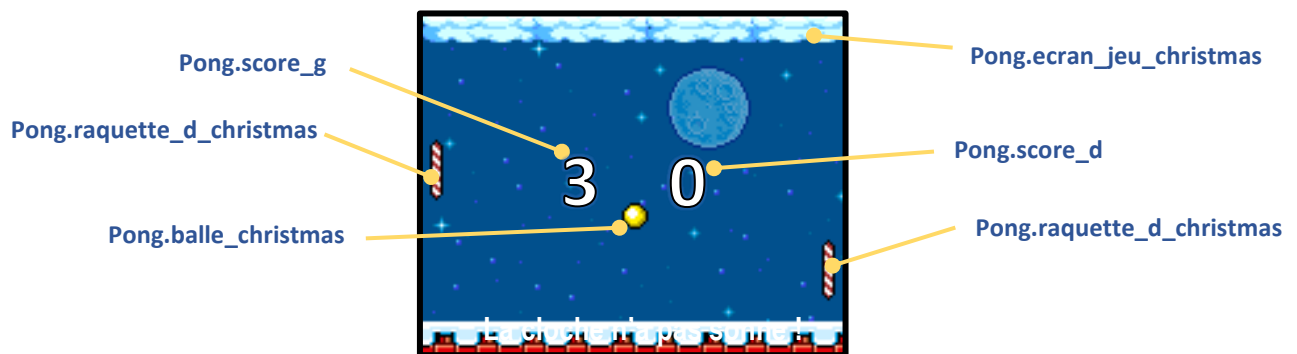
- 'Pong.ecran_regles_christmas' :



Image de taille 160x128 pixels qui servira de fond lors de l'écran des règles du jeu.

- 'Pong.label1_regles' et 'Pong.label2_regles' : zones de texte à positionner correctement sur l'écran.
- 'Pong.label3_regles' : zones de texte à positionner correctement sur l'écran. Vous devrez également modifier le texte pour qu'il corresponde aux règles de l'image de la page précédente.

RESSOURCES DE L'ÉCRAN DE JEU



- 'Pong.ecran_jeu_christmas' :



Image de taille 160x128 pixels qui servira de fond lors du jeu.

ATTENTION ! Du à la présence de nuages dans la partie supérieur et du muret couvert de neige dans la partie inférieure, l'aire de jeu sera plus petite que la taille réelle de l'écran.

Pour faciliter votre code, vous disposez de **CONSTANTES** délimitant cette aire de jeu :

- **'Pong.bord_superieur'**
 - **'Pong.bord_inferieur'**
 - **'Pong.bord_gauche'**
 - **'Pong.bord_droit'**
- **'Pong.score_g'** et **'Pong.score_d'** : images affichant le score du joueur. Il ne s'agit pas ici d'une zone de texte comme dans le tutoriel mais bien d'une image.



Image
ressource



Index
d'animation

Il s'agit finalement d'une feuille d'animation de 80x9 pixels. Le score à afficher correspond à l'index de l'image-clé de 8x9 pixels.

- **'Pong.raquette_g_christmas'** et **'Pong.raquette_d_christmas'** :



Cette image est celle de chaque raquette (gauche et droite). Elle fait 5x23 pixels.

Chacune de ces deux raquettes ne peut se déplacer qu'en vertical (axe y) dans la zone réduite de l'aire de jeu entre le bord supérieur (**'Pong.bord_superieur'**) et le bord inférieur (**'Pong.bord_inferieur'**).

Pour vous aider, vous disposez également de **CONSTANTES** :

- **'Pong.largeur_raquette'**
- **'Pong.hauteur_raquette'**

- **'Pong.balle_christmas'** :



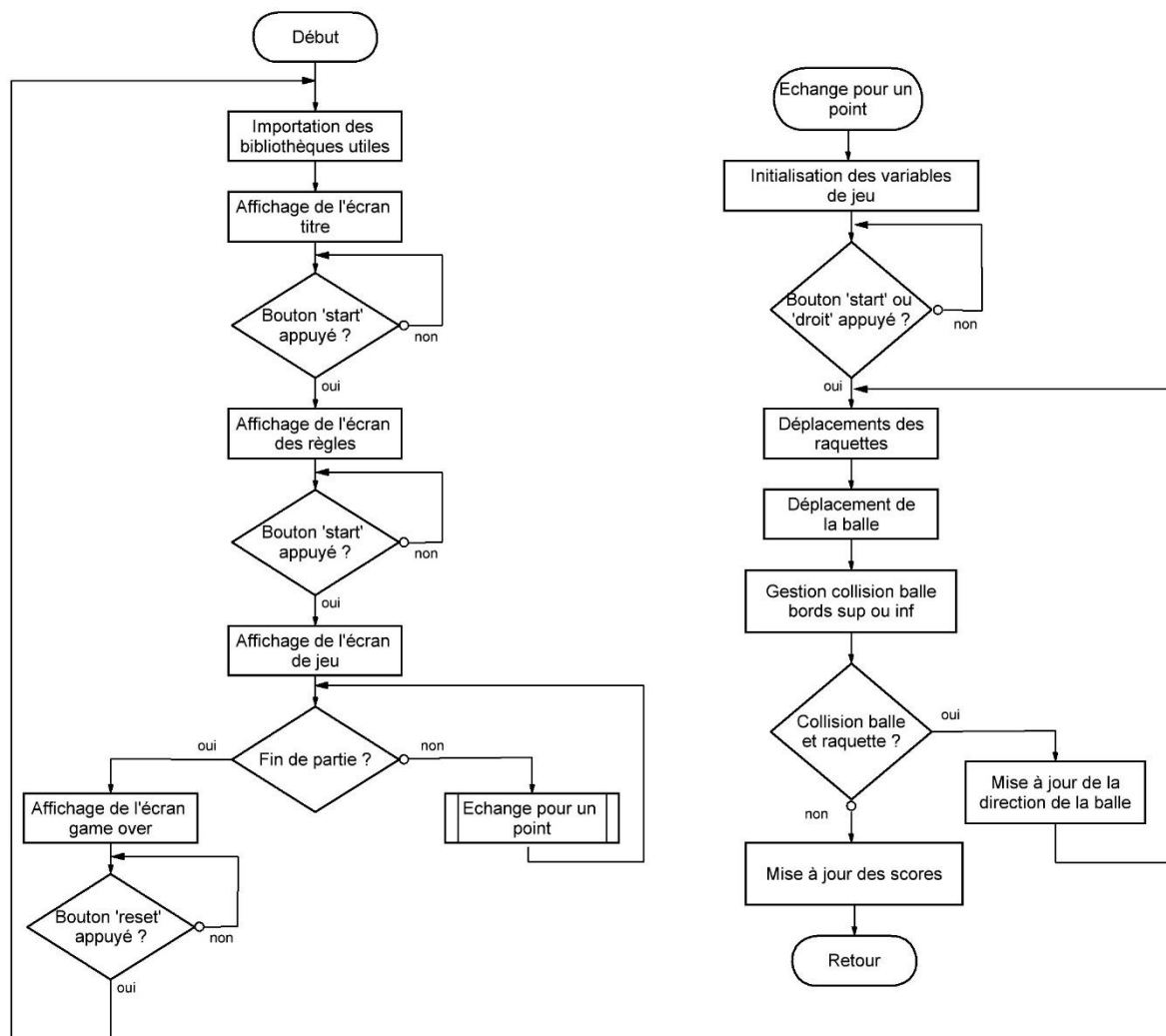
Il s'agit d'une image de 10x10 pixels. Sa taille est disponible via une **CONSTANTE** dans le code : **'Pong.diametre_balle'**

- **'Pong.bruitage_rebond_murs'** et **'Pong.bruitage_rebond_raquette'** : ces bruitages sont joués durant la phase de jeu respectivement lorsque la balle rebondit sur un mur (nuages dans le haut ou muret enneigé dans le bas) ou sur les raquettes des joueurs.

MÉCANIQUE DE BASE DU JEU

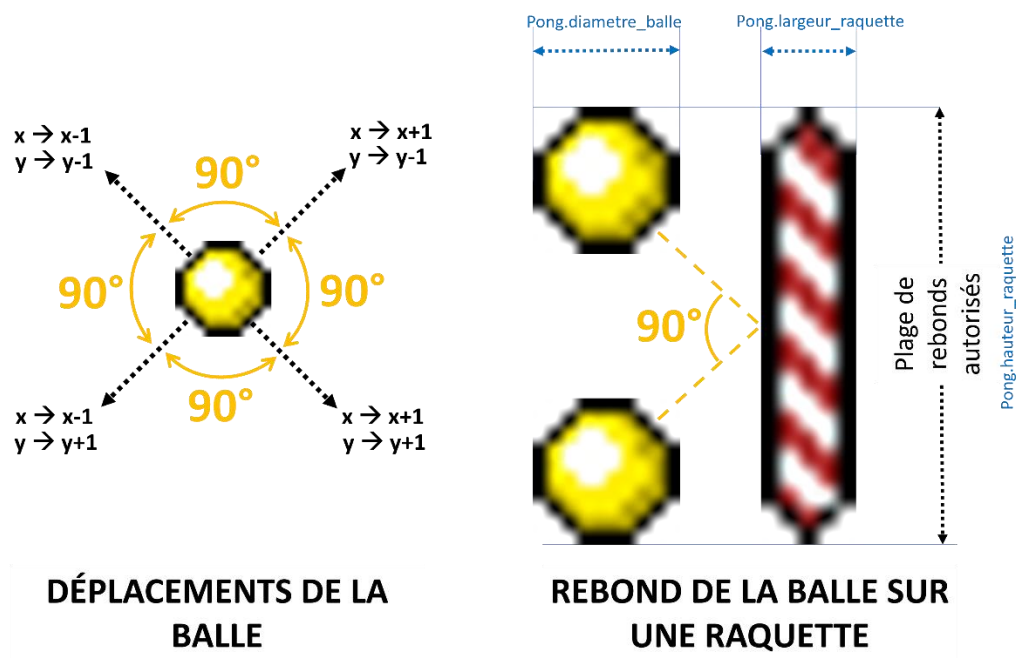
Vous allez vous limiter dans cette première version à une solution simple ! On la fera évoluer par la suite.

L'algorithme suivant devrait vous aider :

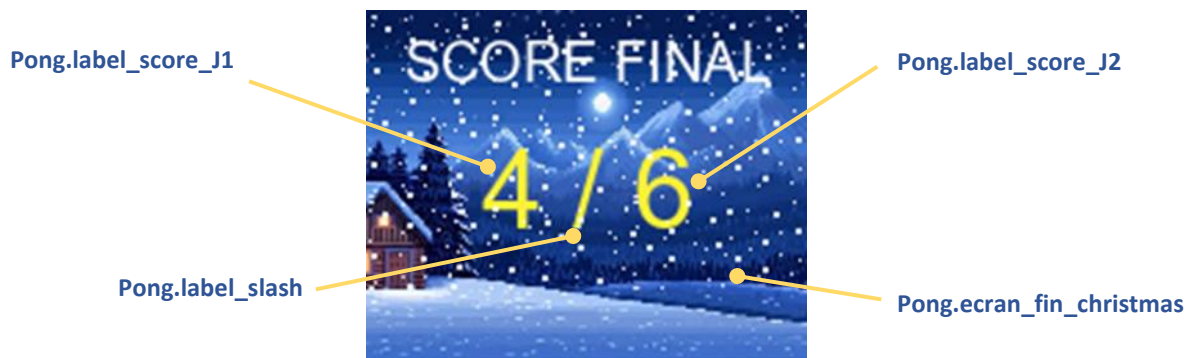


Mais celui-ci peut ne pas être suffisant. Voici donc quelques informations supplémentaires :

- Dans cette version, la balle ne peut se déplacer qu'en diagonale, y compris lorsqu'elle rebondit sur un bord ou sur une raquette (cf. graphique de la page suivante).
- Les vitesses de déplacement des raquettes et de la balle sont constantes. Vous pouvez, dans cette version, utiliser un 'time.sleep()' dans la boucle infinie pour contrôler la vitesse du jeu.
- Si la balle dépasse le bord intérieur de la raquette sans rebondir, elle est perdue.
- Pour que votre code soit lisible et portable, vous devez utiliser les CONSTANTES (largeur et hauteur de raquette, diamètre de balle, bord supérieur et inférieur...)
- Vous devrez utiliser des variables booléennes (ex : balle_perdue) pour effectuer vos tests de perte de point ou de fin de partie...
- Le score n'est pas affiché en permanence, mais uniquement lorsqu'un point est marqué. Il reste alors affiché pendant une seconde par exemple.



RESSOURCES DE L'ÉCRAN DE FIN



- 'Pong.ecran_fin_christmas' :



Image de taille 160x128 pixels qui servira de fond pour l'écran de fin de jeu.

- 'Pong.label_score_J1', 'Pong.label_slash' et 'Pong_label_score_J2' : zones de texte pour afficher le score final des deux joueurs.
- 'Pong.bruitage_game_over' : son à jouer pour clôturer la partie.

ÉTAPES D'ÉCRITURE DU CODE : AIDE

Voici également les étapes que je vous conseille pour éviter d'aborder le problème présenté par cet algorithme dans son ensemble dès le début :

- **Version Alpha 0** : on se concentre sur la phase de jeu
 - Uniquement l'écran de jeu
 - Rebond de la balle sur tous les bords de la zone de jeu
 - Aucun son
- **Version Alpha 1** :
 - Ajout du déplacement indépendant des raquettes
- **Version Alpha 2** :
 - Gestion des collisions de la balle sur les raquettes + perte de balle si absence de collision
- **Version Beta** :
 - Gestion des scores
 - Gestion de la fin de partie
- **Version 1.0** :
 - Ajout des écrans de titre, des règles et de fin
 - Toujours aucune animation
 - Toujours aucun son
- **Version 1.0** :
 - Ajout de l'animation pour l'écran de titre
 - Ajout des sons

EXIGENCES CONCERNANT L'ÉVALUATION



Votre code final devra être commenté. Nous le récupérerons lors de la soutenance orale.

Pour les commentaires, vous pouvez prendre exemple sur le tutoriel, notamment pour saisir ce qu'il faut commenter et la quantité de commentaires nécessaire pour assurer une bonne compréhension de votre code.

De plus, votre code est personnel. Vous serez interrogé sur son contenu lors de la soutenance orale. Si l'on constate que vous ne l'avez pas compris (soit que vous l'avez simplement recopié sur quelqu'un d'autre), vous serez pénalisé !

AMÉLIORATIONS POSSIBLES :

Cette version de Pong est jouable mais basique. **A ce stade, vous pouvez décider s'il vous reste du temps d'entreprendre un nouveau jeu ou de pousser un peu plus loin celui-ci en l'améliorant.**

PREMIÈRE AMÉLIORATION (v2.0) : DÉMARRAGE D'UN NOUVEAU POINT

Une première amélioration consiste à :

- Permettre uniquement au joueur qui vient de gagner le point, de lancer la balle par l'appui sur un bouton et décider ainsi du moment où le jeu redémarre pour un nouveau point.
- Faire en sorte qu'au lancement de la balle au démarrage, elle se dirige vers le joueur qui vient de perdre le point.
- D'ajouter un peu d'aléatoire au début du nouveau point en dirigeant la balle vers le haut ou vers le bas de l'écran

CONSEILS :

- Vous pouvez utiliser une variable qui sauvegarde le joueur qui vient de marquer le dernier point.
- Pour la direction aléatoire, vous pouvez utiliser la méthode 'random.choice()'

DEUXIÈME AMÉLIORATION (v3.0) : ACCÉLÉRATION DU JEU

Si les joueurs se sont un peu entraînés, un échange de point peut durer beaucoup de temps car il n'y a aucune modification de la mécanique du jeu pour le moment.

Vous allez faire en sorte dans cette version que la balle s'accélère au fur et à mesure que l'échange dure dans le temps.

Si vous vous êtes inspirés du tutoriel, vous avez certainement réglé la vitesse du jeu en utilisant un '**time.sleep(valeur)**' dans votre boucle de jeu.

Ainsi pour augmenter la vitesse du jeu au cours d'un échange, la première idée consiste à diminuer au fur et à mesure la valeur fournie au '**time.sleep()**'.

Effectivement, la balle va s'accélérer mais le mouvement des raquettes également ce qui rendra leur positionnement plus délicat.

Dans les jeux, si l'on veut obtenir des vitesses de déplacements différentes, l'utilisation de '**time.sleep()**' n'est pas adaptée. On va plutôt utiliser la méthode '**time.monotonic()**' que vous connaissez déjà il me semble :

```
while True :  
    Deplacements_raquettes()  
    Deplacement_balle()  
    time.sleep(valeur)
```



```
debut = time.monotonic()  
debut_raquettes = debut  
debut_balle = debut  
  
while True :  
    actuel = time.monotonic()  
    if (actuel - debut_raquettes) > 0.1 :  
        Deplacements_raquettes()  
        debut_raquettes = actuel  
    if (actuel - debut_balle) > 0.05 :  
        Deplacement_balle()  
        debut_balle = actuel
```


Le code est un peu plus long, mais il permet dans cet exemple de faire bouger la balle deux fois plus vite que les raquettes ce qui rend forcément le jeu plus difficile.

CONSEILS :

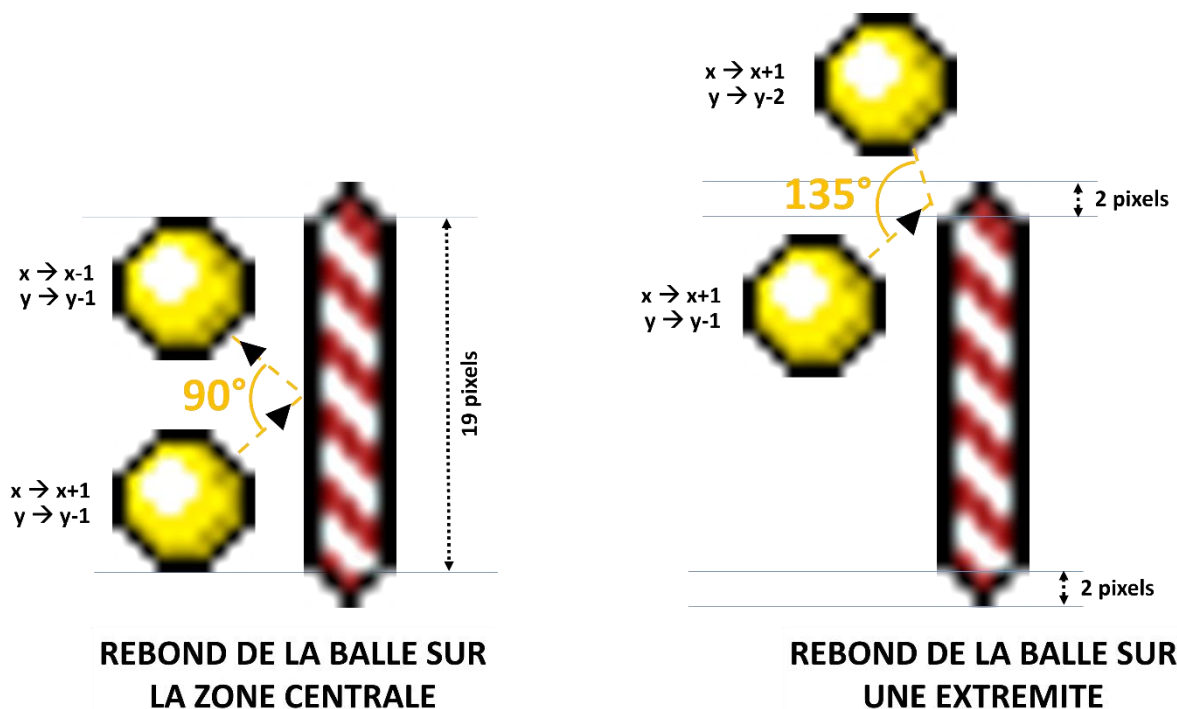
- Une solution consiste à créer une variable qui stocke le nombre de rebonds sur les raquettes. Plus le nombre de rebonds augmente, plus la valeur associée au test 'actuel – debut_balle' soit être faible.
- Vous pouvez utiliser une liste pour stocker des valeurs de test ou trouver une formule qui permette d'augmenter la vitesse en fonction du nombre de rebonds.

TROISIÈME AMÉLIORATION (v4.0) : DIRECTIONS DE REBOND DIFFÉRENTES

Dans la version initiale du Pong de 1972, Allan Alcorn (l'ingénieur-programmeur du jeu) avait divisé chaque raquette en 8 segments différents pour obtenir 4 angles de rebond différents pour la balle selon le segment de la raquette qu'elle touche.

Toutefois, même si l'écran était analogique, il avait une définition approximative de 256x224 pixels. Il était donc environ d'une définition 75% plus élevée que le nôtre. Cela permettait plus de souplesse dans le choix des angles des directions.

Dans notre cas, on va se contenter de gérer deux angles de direction lors des rebonds sur les raquettes :



CONSEILS :

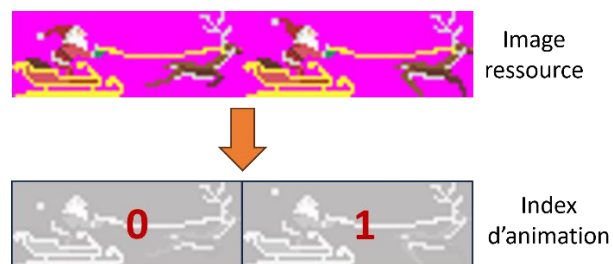
- Alors que vous pouviez faire un seul test auparavant pour gérer le rebond de la balle sur la raquette, il va falloir maintenant en faire 2 ou 3.
- Vous pouvez utiliser une variable 'multiplicateur' pour augmenter le déplacement en y. Celle-ci prendra la valeur 1 ou 2 selon la zone de rebond sur la raquette.

- Vous pouvez également agrandir si vous le jugez utile pour le jeu la zone de rebond aux extrémités et passer de 2 à 3 ou 4 pixels par exemple.

QUATRIÈME AMÉLIORATION (v5.0) : ANIMATIONS DE FIN

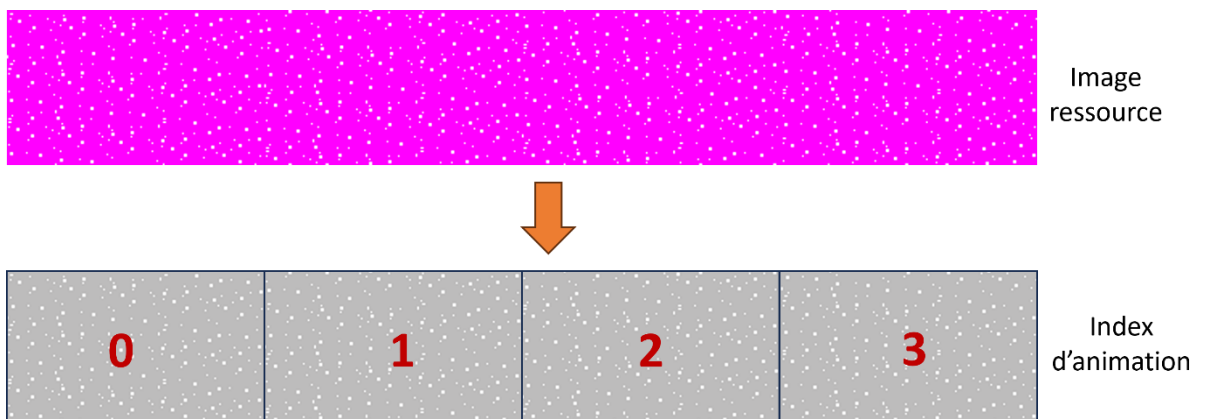


- 'Pong.traineau' :



La feuille d'animation fait 128x24 pixels mais l'image clé ne fait que 64x24 pixels.

- 'Pong.neige' : Il s'agit encore d'une feuille d'animation de 640x96 pixels dont chaque image-clé fait 160x96. Elle occupe presque la totalité de l'écran et représente de la neige qui tombe :



L'idée de cette animation est de faire tomber la neige continuellement jusqu'à l'appui par le joueur sur le bouton 'Reset'.

Concernant le traineau, l'animation consiste à le faire traverser l'écran de gauche à droite. Au fur et à mesure de son avancée, le score du joueur de gauche apparaît, puis le slash et enfin le score du joueur de droite.

DERNIÈRE AMÉLIORATION (v6.0) : MISE EN PLACE DE BONUS

Cette dernière amélioration va ajouter des bonus au cours du jeu. Cette amélioration constitue un petit challenge en termes de programmation mais permet de pimenter le jeu de manière intéressante :



Un père Noël marche sur le muret en faisant des aller-retours (de $x=30$ à $x=110$)



Si la balle touche le père Noël, un cadeau bonus apparaît aléatoirement dans l'espace de jeu



Le joueur qui touche le cadeau bonus avec la balle fait exploser la raquette de l'autre joueur et marque donc le point

Voici les ressources disponibles :

- **'Pong. Santa_claus'** : Feuille d'animation de 256x96 pixels comprenant 3 séquences ('Marche', 'repos' et 'touché'). Chaque image-clé fait 32x32 pixels.



Image
ressource



'MARCHÉ'	0	1	2	3	4	5	6	7
'REPOS'	8	9	10	11	12	13		
'TOUCHÉ'	16	17	18					

Index
d'animation

- **'Pong.cadeau'** : Il s'agit d'une image fixe de 16x18 pixels.



- **'Pong.explosion'** : Feuille d'animation de 352x32 pixels. Chaque image-clé fait 32x32 pixels.



Image
ressource



0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

Index
d'animation

- **'Pong.bruitage_explosion'** : Son à jouer au moment où la raquette d'un joueur explose
- **'Pong.bruitage_ouch'** : Son à jouer lorsque le père Noël est touché par une balle

Je ne vous donne pas plus d'informations, vous allez essayer de vous débrouiller par vous-même, de la façon dont vous le souhaitez. Avec un peu d'organisation et de rigueur dans votre code, vous devriez vous en sortir !