

Cours 14 : SOLUTIONS Jeu des allumettes

Objectifs du projet :

- Récapitulation de toutes les notions de programmation vues au cours ;
- Élaboration du logigramme correspondant à un énoncé ;
- Mise en application dans Python :
 - **Entrée / sortie ;**
 - **Variables ;**
 - **Séquence ;**
 - **Répétition : bornée et non bornée ;**
 - **Exécution conditionnelle.**
- Réalisation d'un projet complet par étapes :
 - « **Jeu des allumettes** ».

Les étapes de conception du jeu des allumettes :

0. Jouer au jeu avec votre collègue pour comprendre les règles et élaborer l'algorithme ;
1. Dessiner le logigramme du jeu ;
2. Tirer un nombre entier pseudo-aléatoire entre deux bornes et l'afficher ;
3. Deux joueurs :
 - Tant qu'il reste au moins une allumette :
 - Les joueurs prennent alternativement 1, 2 ou 3 allumettes.
 - Le joueur qui prend la dernière allumette a **gagné**. (Il existe d'autres variantes.)
4. Les joueurs sont nommés :
 - Les joueurs donnent leur nom en début de partie ;
 - A chaque message le nom du joueur concerné est affiché dans le message.
5. Le joueur joue avec l'ordinateur
 - C'est l'ordinateur qui commence ;
 - L'ordinateur joue en tirant un nombre pseudo-aléatoire entre 1 et 3 (bornes comprises).
6. Le joueur joue avec l'ordinateur : l'ordinateur gagne à toutes les parties
 - C'est le joueur qui commence ;
 - L'ordinateur utilise la stratégie gagnante.
7. Validité des saisies
 - Vérifier que les saisies du joueur sont valides (nombre entier entre min et max) ;
 - Si la saisie n'est pas valide, informer le joueur et lui redemander un autre nombre.
8. Rejouer
 - A la fin d'une partie, le jeu demande au joueur s'il souhaite rejouer.

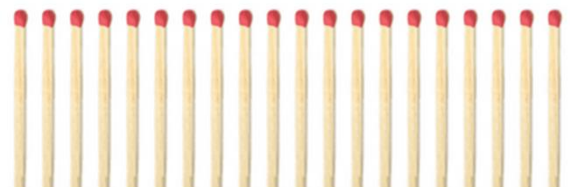
Règles du jeu des allumettes

Le jeu des allumettes fait partie des jeux de Nim qui sont des jeux de stratégie pure.

Voir par exemple : https://fr.wikipedia.org/wiki/Jeux_de_Nim

Règle du jeu :

- Le jeu commence avec 20 allumettes.
- A tour de rôle les joueurs retirent 1, 2 ou 3 allumettes.
- Le joueur qui prend la dernière allumette à GAGNE



Remarque : il existe plusieurs autres variantes de ce jeu.

Cours 14 : SOLUTIONS Jeu des allumettes

Etape n° 2 (Nom fichier : C14_E02)

Tirer un nombre entier pseudo-aléatoire entre deux bornes et l'afficher.

Exemple :

Tirer un nombre entre 1 et 3 : **écrire directement ce code dans l'éditeur pour tester.**

```
#Projet 01 : JEU DES ALLUMETTES
#Tirage d'un nombre entier pseudo-aléatoire
import random

borne_inf = 1
borne_sup = 3

#Tirage d'un nombre pseudo-aléatoire
n = random.randint(borne_inf, borne_sup)
print(n, ": nombre pseudo-aléatoire entre", borne_inf, "et", borne_sup, "inclus")
```

Etape n° 3 (Nom fichier : C14_E03)

Deux joueurs :

- Tant qu'il reste au moins une allumette :
 - Les joueurs prennent alternativement 1, 2 ou 3 allumettes.
- Le joueur qui prend les dernières allumettes a gagné.

(Voir également le code complet après l'analyse.)

Déclaration des Constantes et Variables Initiales :

```
python
ALLUMETTES = 20
allumettes_restantes = ALLUMETTES
joueur_actuel = 1
```

- **ALLUMETTES** est une constante qui représente le nombre total d'allumettes avec lesquelles on commence le jeu. En Python, les constantes sont généralement écrites en majuscules pour les distinguer des variables.
- **allumettes_restantes** est une variable qui garde une trace du nombre d'allumettes restantes. Elle est initialement définie sur la même valeur que ALLUMETTES.
- **joueur_actuel** est une variable qui indique quel joueur doit jouer. Elle alterne entre 1 et 2 pour représenter les deux joueurs.

Boucle du Jeu (while) :

```
while allumettes_restantes > 0:
```

- Cette ligne commence une boucle while qui continue tant qu'il reste des allumettes (allumettes_restantes > 0).

Affichage du Nombre d'Allumettes et du Tour du Joueur :

```
print(f"\nIl reste {allumettes_restantes} allumettes.")
print(f"Joueur {joueur_actuel}, c'est votre tour.\n")
```

- Ces deux lignes affichent le nombre d'allumettes restantes et indiquent quel joueur doit jouer. Les f avant les guillemets indiquent qu'il s'agit de f-strings, une façon de formater les chaînes de caractères en Python qui permet d'inclure des variables directement dans la chaîne.

Cours 14 : SOLUTIONS Jeu des allumettes

Demande de l'Action du Joueur :

```
prise = int(input("Combien d'allumettes prenez-vous (1, 2, 3) ? "))
```

- Cette ligne demande au joueur actuel de choisir combien d'allumettes prendre. La fonction input est utilisée pour saisir une entrée utilisateur, et int convertit cette entrée en un nombre entier.

Vérification de la Validité de la Prise :

```
if prise >= 1 and prise <= 3 and prise <= allumettes_restantes:
    allumettes_restantes = allumettes_restantes - prise
    if joueur_actuel == 1:
        joueur_actuel = 2
    else:
        joueur_actuel = 1
else:
    print("\nPrise invalide. Veuillez prendre 1, 2 ou 3 allumettes.")
```

- Cette partie vérifie si le nombre d'allumettes pris est valide (entre 1 et 3 et ne dépasse pas le nombre restant). Si la prise est valide, le nombre d'allumettes restantes est réduit et le tour passe au prochain joueur. Sinon, un message d'erreur s'affiche.

Vérification de la Fin du Jeu :

```
if allumettes_restantes == 0:
    if joueur_actuel == 2:
        print(f"\nFélicitations Joueur 1 ! Vous avez gagné !\n")
    else:
        print(f"\nFélicitations Joueur 2 ! Vous avez gagné !\n")
```

- Lorsque toutes les allumettes ont été prises (allumettes_restantes == 0), le jeu se termine. Le joueur qui n'a pas pris la dernière allumette est déclaré gagnant.

CODE PYTHON

```
# Nombre initial d'allumettes en tant que constante
ALLUMETTES = 20

# Nombre d'allumettes restantes
allumettes_restantes = ALLUMETTES

# Numéro du joueur actuel (1 ou 2)
joueur_actuel = 1

# Boucle du jeu
while allumettes_restantes > 0:
    # Afficher le nombre d'allumettes restantes avec un saut de ligne
    print(f"\nIl reste {allumettes_restantes} allumettes.")
    print(f"Joueur {joueur_actuel}, c'est votre tour.\n")

    # Demander au joueur combien d'allumettes il souhaite prendre
    prise = int(input("Combien d'allumettes prenez-vous (1, 2, 3) ? "))

    # Vérifier si la prise est valide
    if prise >= 1 and prise <= 3 and prise <= allumettes_restantes:
        allumettes_restantes = allumettes_restantes - prise
        # Changer de joueur
        if joueur_actuel == 1:
            joueur_actuel = 2
        else:
            joueur_actuel = 1
    else:
        print("\nPrise invalide. Veuillez prendre 1, 2 ou 3 allumettes.")

    # Vérifier si le jeu est terminé
    if allumettes_restantes == 0:
        if joueur_actuel == 2: # Inverse le joueur pour annoncer le gagnant
            print(f"\nFélicitations Joueur 1 ! Vous avez gagné !\n")
        else:
            print(f"\nFélicitations Joueur 2 ! Vous avez gagné !\n")
```

Cours 14 : SOLUTIONS Jeu des allumettes**Etape n° 4** (Nom fichier : C14_E04)**Les joueurs sont nommés :**

- Les joueurs donnent leur nom en début de partie ;
- A chaque message le nom du joueur concerné est affiché dans le message.

CODE PYTHON

```
# Nombre initial d'allumettes en tant que constante
ALLUMETTES = 20

# Demander aux joueurs d'entrer leurs prénoms
prenom_joueur1 = input("Joueur 1, veuillez entrer votre prénom : ")
prenom_joueur2 = input("Joueur 2, veuillez entrer votre prénom : ")

# Numéro du joueur actuel (1 ou 2)
joueur_actuel = 1

# Boucle du jeu
allumettes_restantes = ALLUMETTES
while allumettes_restantes > 0:
    # Afficher le nombre d'allumettes restantes avec un saut de ligne
    print(f"\nIl reste {allumettes_restantes} allumettes.")

    # Afficher le prénom du joueur actuel
    if joueur_actuel == 1:
        print(f"{prenom_joueur1}, c'est votre tour.\n")
    else:
        print(f"{prenom_joueur2}, c'est votre tour.\n")

    # Demander au joueur combien d'allumettes il souhaite prendre
    prise = int(input("Combien d'allumettes prenez-vous (1, 2, 3) ? "))

    # Vérifier si la prise est valide
    if prise >= 1 and prise <= 3 and prise <= allumettes_restantes:
        allumettes_restantes = allumettes_restantes - prise
        # Changer de joueur
        if joueur_actuel == 1:
            joueur_actuel = 2
        else:
            joueur_actuel = 1
    else:
        print("\nPrise invalide. Veuillez prendre 1, 2 ou 3 allumettes.\n")

    # Vérifier si le jeu est terminé
    if allumettes_restantes == 0:
        if joueur_actuel == 2: # Inverse le joueur pour annoncer le gagnant
            print(f"\nFélicitations {prenom_joueur1} ! Vous avez gagné !\n")
        else:
            print(f"\nFélicitations {prenom_joueur2} ! Vous avez gagné !\n")
```

Cours 14 : SOLUTIONS Jeu des allumettes**Etape n° 5** (Nom fichier : C14_E05)**Le joueur joue avec l'ordinateur**

- C'est l'ordinateur qui commence ;
- L'ordinateur joue en tirant un nombre pseudo-aléatoire entre 1 et 3 (bornes comprises);

CODE PYTHON

```
import random # Importer le module random pour générer des nombres aléatoires

# Nombre initial d'allumettes en tant que constante
ALLUMETTES = 20

# Nom de l'ordinateur
NOM_ORDINATEUR = "HAL";

# Demander au joueur d'entrer son prénom
prenom_joueur = input("Veuillez entrer votre prénom : ")

# Numéro du joueur actuel (1 pour le joueur , 2 pour l'ordinateur)
joueur_actuel = 1

# Boucle du jeu
allumettes_restantes = ALLUMETTES
while allumettes_restantes > 0:
    # Afficher le nombre d'allumettes restantes avec un saut de ligne
    print(f"\nIl reste {allumettes_restantes} allumettes.")

    if joueur_actuel == 1: # Tour du joueur
        print(f"{prenom_joueur}, c'est votre tour.\n")
        prise = int(input("Combien d'allumettes prenez-vous (1, 2, 3) ? "))
        if prise >= 1 and prise <= 3 and prise <= allumettes_restantes:
            allumettes_restantes = allumettes_restantes - prise
            joueur_actuel = 2 # Passer au tour de l'ordinateur
        else:
            print("\nPrise invalide. Veuillez prendre 1, 2 ou 3 allumettes.")

    else: # Tour de l'ordinateur
        prise = random.randint(1, min(3, allumettes_restantes))

        print(f"{NOM_ORDINATEUR} prend {prise} allumettes.")
        allumettes_restantes -= prise
        joueur_actuel = 1 # Revenir au tour du joueur

# Vérifier si le jeu est terminé
if allumettes_restantes == 0:
    if joueur_actuel == 2: # Inverse le joueur pour annoncer le gagnant
        print(f"\nFélicitations {prenom_joueur} ! Vous avez gagné !\n")
    else:
        print(f"\n{NOM_ORDINATEUR} a gagné.\n")
```

Cours 14 : SOLUTIONS Jeu des allumettes**Etape n° 6** (Nom fichier : C24_E06)**Le joueur joue avec l'ordinateur : l'ordinateur gagne à toutes les parties**

- C'est le joueur qui commence ;
- L'ordinateur utilise la stratégie gagnante.

CODE PYTHON**Stratégie gagnante :**

```
if allumettes_restantes % 4 != 0:
    prise = allumettes_restantes % 4
else:
    prise = random.randint(1, min(3, allumettes_restantes))
print(f"HAL prend {prise} allumettes.")
```

- Description de la stratégie gagnante :

Stratégie Gagnante dans le Jeu de Nim

Le jeu de Nim, dans sa forme la plus simple, implique deux joueurs qui prennent alternativement entre 1 et 3 allumettes d'un tas. Le gagnant est celui qui prend la dernière allumette. La stratégie gagnante pour ce jeu repose sur la compréhension des multiples de 4 :

- L'objectif est de forcer l'adversaire à prendre des allumettes de manière que le nombre restant d'allumettes soit toujours un multiple de 4 après votre tour.
- Si vous pouvez prendre les allumettes de manière à laisser 4, 8, 12, 16, etc., allumettes, vous pouvez garantir une victoire, car quel que soit le nombre d'allumettes que l'adversaire prend ensuite (1, 2 ou 3), vous pouvez toujours ramener le total à un autre multiple de 4.

Exemple :

1. Le jeu commence avec 20 allumettes.
2. Vous prenez 2 allumettes, il en reste 18 (multiple de 4).
3. Peu importe combien l'adversaire prend ensuite (1, 2 ou 3), vous pouvez toujours ajuster votre prise pour revenir à un multiple de 4.

Position de Force pour le Second Joueur : Si le premier joueur commence et que le nombre total d'allumettes est un multiple de 4 (comme 20), le second joueur peut toujours ajuster sa prise pour revenir à un autre multiple de 4 après son tour.

Stratégie de Réponse : Peu importe combien d'allumettes le premier joueur prend (1, 2, ou 3), le second joueur peut toujours ajuster sa prise pour que le total d'allumettes restantes après son tour soit de nouveau un multiple de 4.

Cours 14 : SOLUTIONS Jeu des allumettes

Rappel Mathématique sur le Modulo

Le modulo est une opération mathématique qui donne le reste d'une division entière. En d'autres termes, si vous divisez un nombre a par un nombre b , le modulo est le reste de cette division. En Python, le modulo est représenté par l'opérateur `%`.

Exemples d'Utilisation du Modulo :

1. $5 \% 2$ donne 1 car lorsqu'on divise 5 par 2, le reste est 1.
2. $9 \% 3$ donne 0 car 9 est divisible par 3 sans reste.
3. $10 \% 4$ donne 2 car 10 divisé par 4 donne 2 comme reste.

Utilisation du Modulo dans le Programme

Dans notre jeu de Nim, le modulo est utilisé pour déterminer la stratégie de l'ordinateur (HAL). L'idée est que si HAL peut laisser un nombre d'allumettes qui est un multiple de 4 après son tour, cela augmente ses chances de gagner.

Voici comment cela fonctionne :

- Si le nombre d'allumettes restantes (`allumettes_restantes`) modulo 4 ($\% 4$) donne un résultat différent de 0, cela signifie que le nombre d'allumettes restantes n'est pas un multiple de 4. Dans ce cas, HAL essaie de prendre suffisamment d'allumettes pour laisser un multiple de 4.
- Si `allumettes_restantes % 4` donne 0, HAL choisit un nombre aléatoire d'allumettes à prendre.

Explication Détaillée du Code

Importation du Module Random et Initialisation des Variables :

- Le module `random` est importé pour générer des nombres aléatoires.
- `ALLUMETTES` est une constante représentant le nombre initial d'allumettes.
- `prenom_joueur` stocke le nom du joueur.
- `joueur_actuel` indique qui doit jouer (1 pour le joueur, 2 pour HAL).
- `allumettes_restantes` suit le nombre d'allumettes restantes dans le jeu.

Boucle de Jeu (While) :

- La boucle continue tant qu'il reste des allumettes.

Tour du Joueur :

- Le joueur est invité à prendre 1, 2 ou 3 allumettes. Si le choix est valide, les allumettes sont prises et c'est au tour de HAL.

Tour de HAL (l'Ordinateur) :

- HAL utilise la stratégie basée sur le modulo pour déterminer combien d'allumettes prendre. Si `allumettes_restantes % 4` n'est pas 0, HAL prend un nombre d'allumettes pour atteindre le prochain multiple de 4. Sinon, il prend un nombre aléatoire.

Fin du Jeu et Annonce du Gagnant :

- Lorsqu'il ne reste plus d'allumettes, le joueur qui n'a pas pris la dernière allumette est déclaré gagnant.

Cours 14 : SOLUTIONS Jeu des allumettes**CODE PYTHON**

```
import random # Importer le module random pour générer des nombres aléatoires

# Nombre initial d'allumettes en tant que constante
ALLUMETTES = 20

# Nom de l'ordinateur
NOM_ORDINATEUR = "HAL";

# Demander au joueur d'entrer son prénom
prenom_joueur = input("Veuillez entrer votre prénom : ")

# Numéro du joueur actuel (1 pour le joueur , 2 pour l'ordinateur)
joueur_actuel = 1

# Boucle du jeu
allumettes_restantes = ALLUMETTES
while allumettes_restantes > 0:
    # Afficher le nombre d'allumettes restantes avec un saut de ligne
    print(f"\nIl reste {allumettes_restantes} allumettes.")

    if joueur_actuel == 1: # Tour du joueur
        print(f"{prenom_joueur}, c'est votre tour.\n")
        prise = int(input("Combien d'allumettes prenez-vous (1, 2, 3) ? "))
        if prise >= 1 and prise <= 3 and prise <= allumettes_restantes:
            allumettes_restantes = allumettes_restantes - prise
            joueur_actuel = 2 # Passer au tour de l'ordinateur
        else:
            print("\nPrise invalide. Veuillez prendre 1, 2 ou 3 allumettes.")
    else: # Tour de l'ordinateur

        # Stratégie gagnante
        if allumettes_restantes % 4 != 0:
            prise = allumettes_restantes % 4
        else:
            prise = random.randint(1, min(3, allumettes_restantes))

        print(f"{NOM_ORDINATEUR} prend {prise} allumettes.")
        allumettes_restantes -= prise
        joueur_actuel = 1 # Revenir au tour du joueur

# Vérifier si le jeu est terminé
if allumettes_restantes == 0:
    if joueur_actuel == 2: # Inverse le joueur pour annoncer le gagnant
        print(f"\nFélicitations {prenom_joueur} ! Vous avez gagné !\n")
    else:
        print(f"\n{NOM_ORDINATEUR} a gagné.\n")
```