

OPENCLASSROOMS  
DÉVELOPPEUR D'APPLICATION PYTHON :  
PROJET 3 : AIDEZ MACGIVER À S'ÉCHAPPER!

---

# ÉVADEZ-VOUS DU LABYRINTHE!

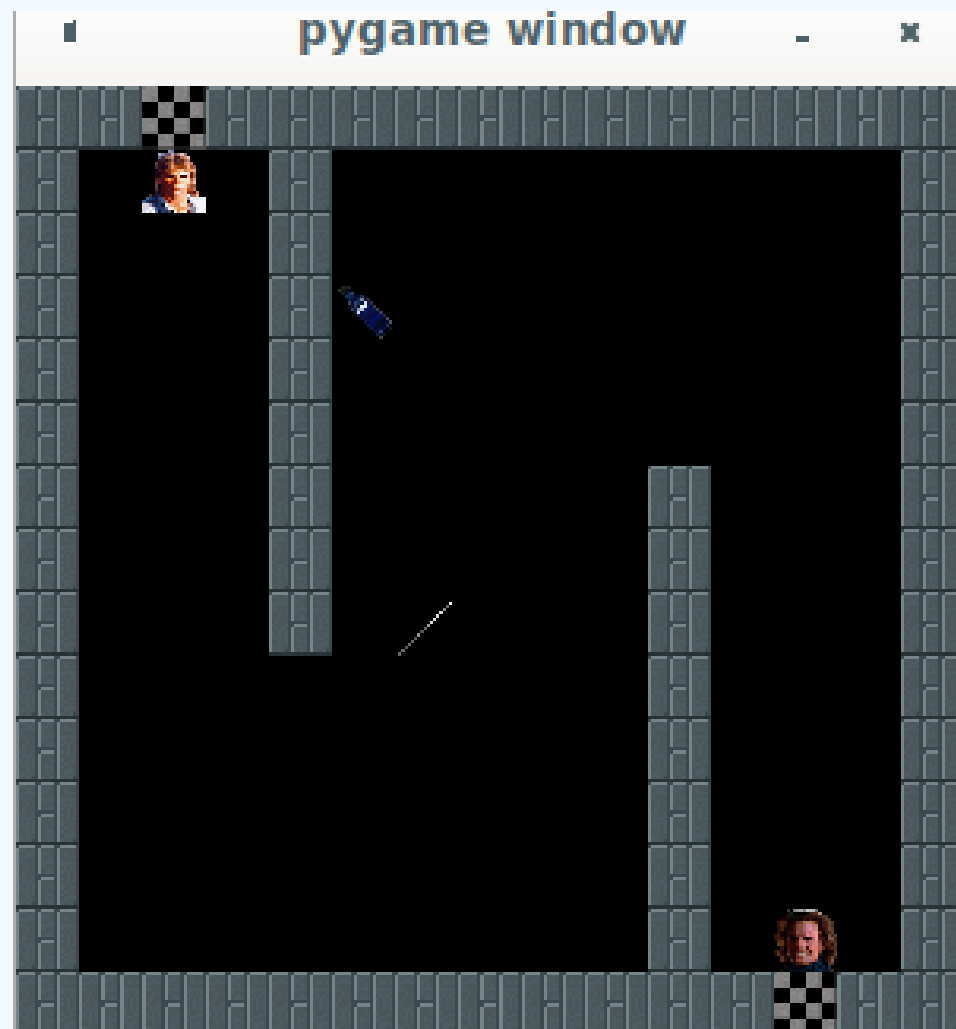
---

Présenté par Céline PELLETIER

# SOMMAIRE :

- Présentation du projet
- Outils utilisés
- L'algorithme:
  - Initialisation
  - Contrôle
  - Affichage du monde
- Les difficultés et solutions

# PRÉSENTATION DU PROJET:



Projet de création d'un petit jeu vidéo en PYTHON.

Labyrinthe avec 2 objets.

Le héro doit récupérer les 2 objets pour endormir le gardien et s'enfuir.

Si il n'a pas les 2 objets en arrivant devant le gardien :  
il perd.

# PYCHARM

- IDE : langage PYTHON.
- permet le versionnage du code.
- permet de commit et push sur Github.
- permet de respecter la PEP 8 de manière intuitive.



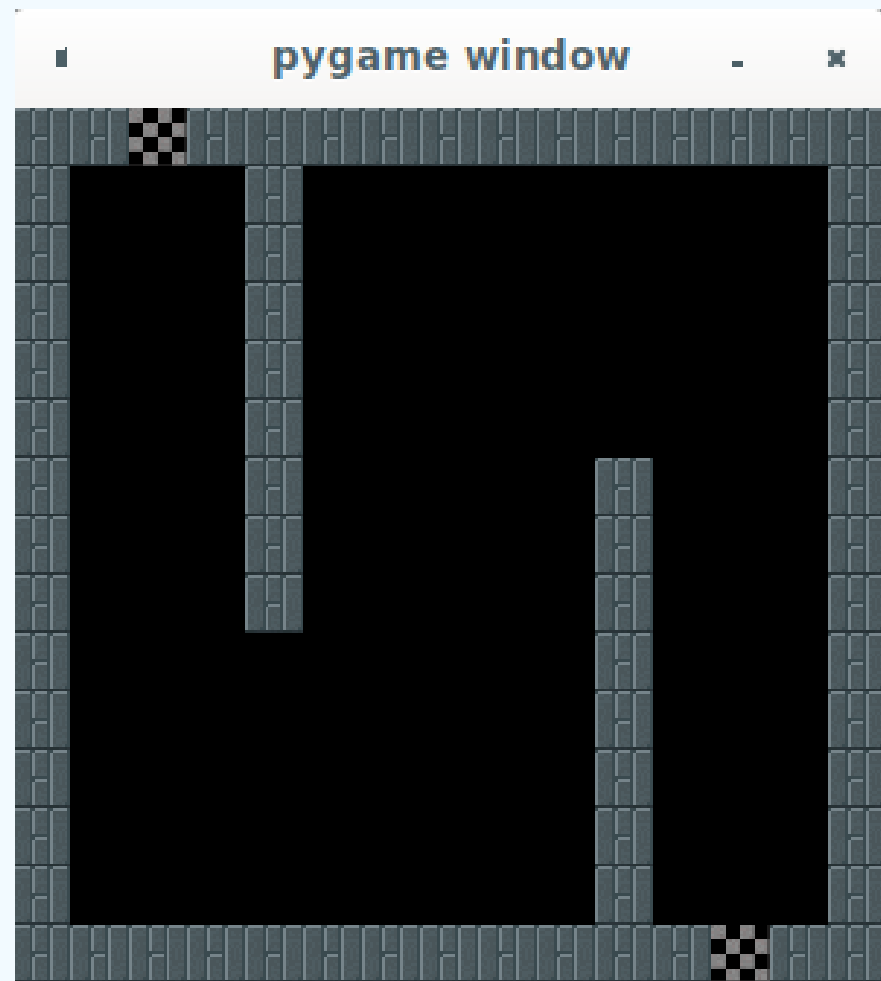
# PYGAME

- Interface Graphique pour PYTHON.
- Permet l'affichage vidéo en 2D
- Permet la gestion de l'audio
- Permet de gérer les évènements. (clavier, souris, collision etc...)



# ALGORITHME

## INITIALISATION :

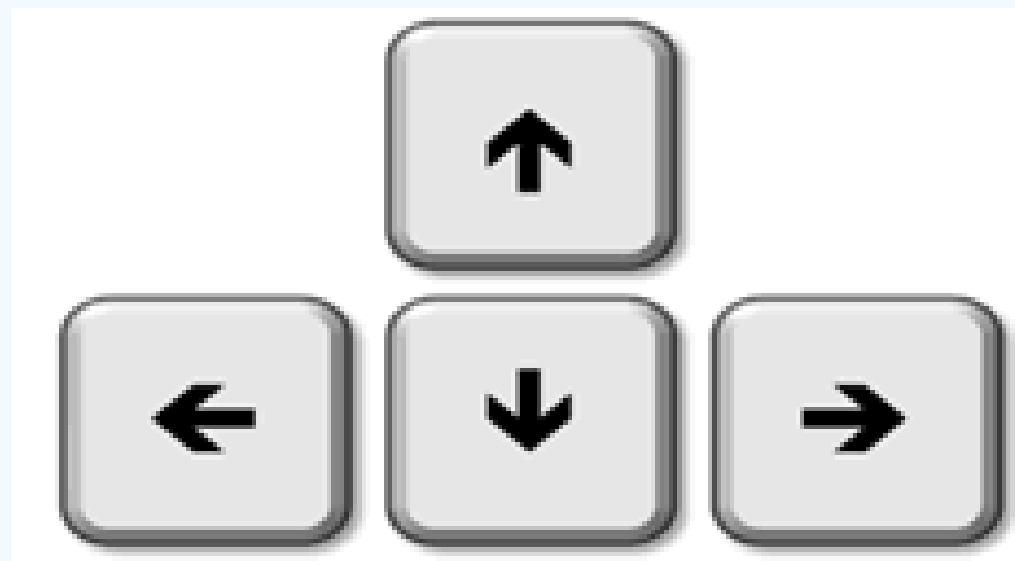


- Création de l'écran.
- Création du labyrinthe.
- Génération de la configuration.
- Création des personnages et des objets.
- Suppression de la position de l'ether de la liste des positions vides.
- Remplacement du "P" en "x" dans la configuration.



# ALGORITHME

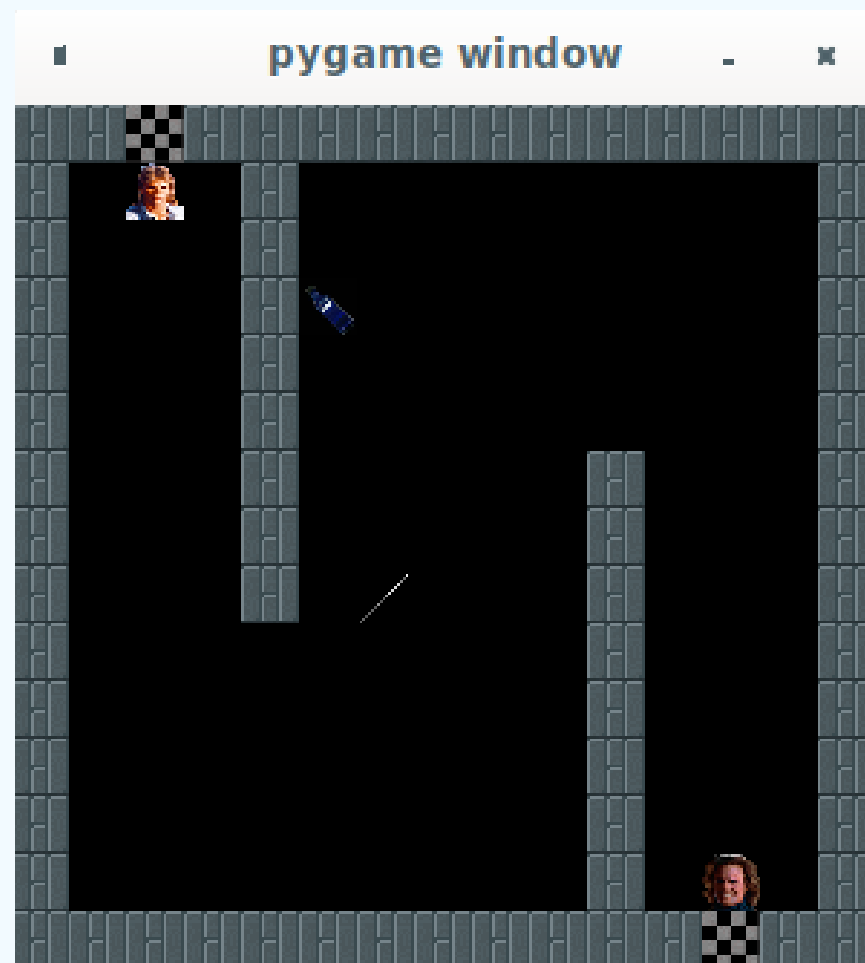
## CONTRÔLE :



- Gestion des contrôles du héros.
- Gestion des collisions.
- Gestion de la récupération des objets du labyrinthe.

# ALGORITHME

## AFFICHAGE DU MONDE :



- Affichage du labyrinthe.
- Affichage des objets.
- Affichage du gardien.
- Affichage du personnage.

# DIFFICULTÉS ET SOLUTIONS

## LE LABYRINTHE

- Afficher des images.
- Afficher des sprites de la map de sols.
- Afficher plusieurs Sprites les uns à côté des autres.
- Créer un labyrinthe en créant des murs.



## SOLUTION

- Créer un labyrinthe manuellement (lettre = Sprite) dans un fichier.
- Créer une classe "Lab".
- Fonction "generate\_lab" : génère la configuration et 2 listes. (murs et positions vides)
- Fonction "diplay\_lab" : affiche le labyrinthe.



# DIFFICULTÉS ET SOLUTIONS

## LES OBJETS

- Afficher aléatoirement sans être sur les murs.
- Effacer quand le héros passe dessus.
- Image au bon format.



## SOLUTIONS

- Création d'une classe "Labobject".
- Image redimensionnée à la création de l'objet.
- Fonction "draw\_me": affiche.

# DIFFICULTÉS ET SOLUTIONS

## LE HÉRO

- Mouvement du héros dans toutes les directions
- Affichage du héros.



## SOLUTIONS

- Création d'une classe "Player".
- Image redimensionnée à sa création.
- Fonction "move\_right", "move\_left", "move\_up" et "move\_down" : déplace le héros.
- Fonction "draw\_me" : affiche.
- Attribut obj1 et 2 == False.

# DIFFICULTÉS ET SOLUTIONS

## IMAGE DU HÉRO AU DÉPART

- Image du héros qui reste au départ.



## SOLUTIONS

- Fonction "create\_character" : crée le héros à sa position initiale.
- Fonction : "erase\_pos\_character" (évite d'avoir l'image du héros tout le temps au départ): remplace "P" par "x" dans la configuration.

# DIFFICULTÉS ET SOLUTIONS

## LE GARDIEN

- Affichage du gardien.



## SOLUTION

- Création d'une classe "Guardian".
- Fonction "draw\_me" : affiche
- Fonction "my\_rect" : retourne le rectangle du gardien.

# DIFFICULTÉS ET SOLUTIONS

## COLLISIONS

- Collisions avec les murs.
- Collisions avec les objets.
- Collisions avec le gardien.



## SOLUTIONS

- Collision murs :  
liste des murs => si collision avec mur de la liste => retour à la position précédente.
- Collision objets :  
si collision avec ether, obj1 == True  
si collision avec needle, obj2 == True
- Collision gardien :  
obj1, obj2 == True => Vous gagnez.  
sinon => Vous perdez.

MERCI DE VOTRE  
ATTENTION.

Céline PELLETIER

