

OPENCLASSROOMS
DÉVELOPPEUR D'APPLICATION PYTHON :
PROJET 3 :

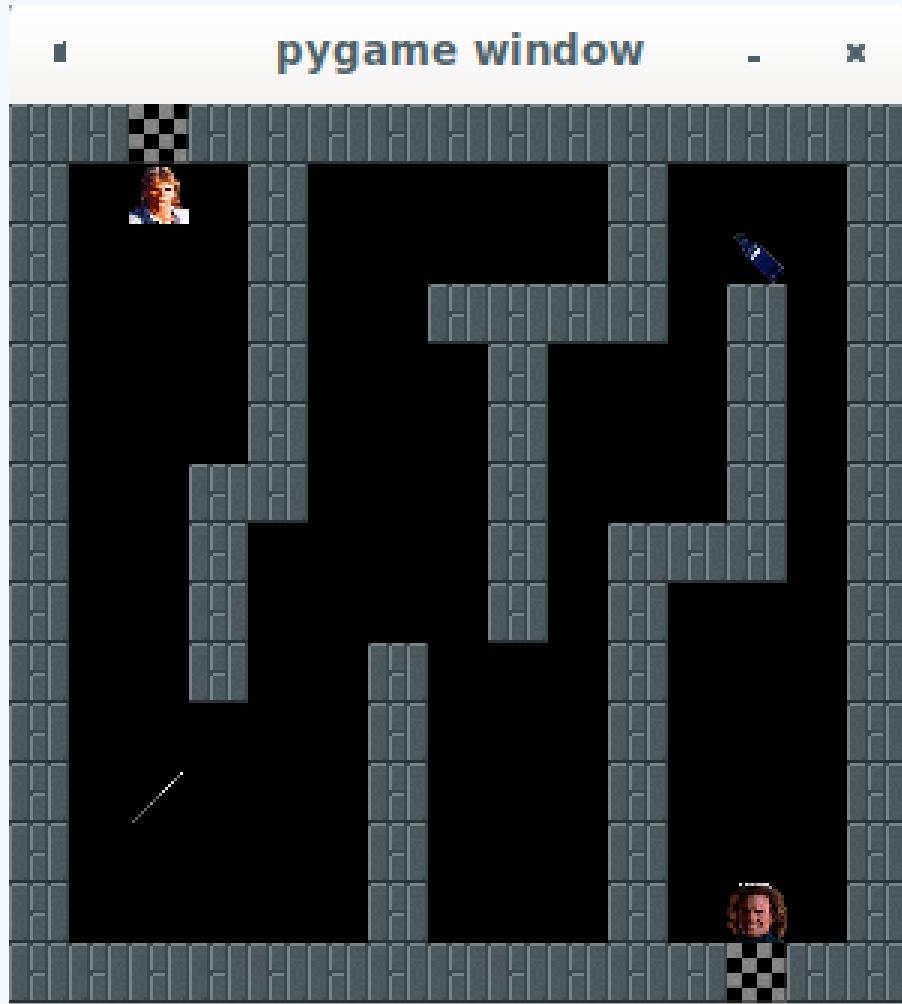
AIDEZ MAC GIVER
À S'ÉCHAPPER!

Présenté par Céline PELLETIER

SOMMAIRE :

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

PRÉSENTATION DU PROJET:



Projet de création d'un petit jeu vidéo en Python3.

Labyrinthe contenant 2 objets.

Le héros 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.

OUTILS :

IDE : PYCHARM

- Partie développement.
- Permet le versionnage du code.
- Permet de commit et push sur Github.
- Permet de respecter la PEP 8 de manière intuitive.



LIBRAIRIE : PYGAME

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

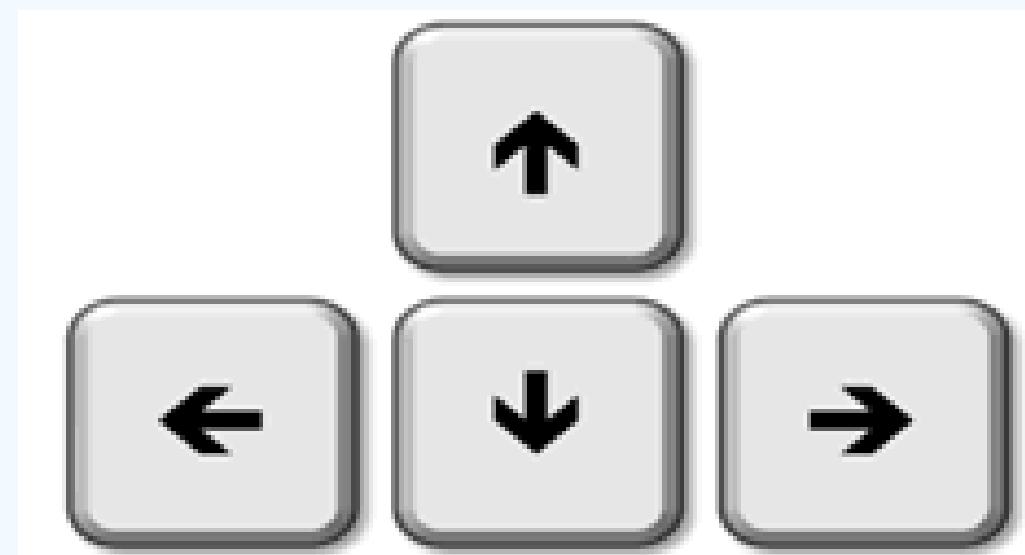
ALGORITHME

INITIALISATION :

- Création de l'écran.
- Création du labyrinthe.
- Génération de la configuration.
- Création des personnages et des objets.

ALGORITHME

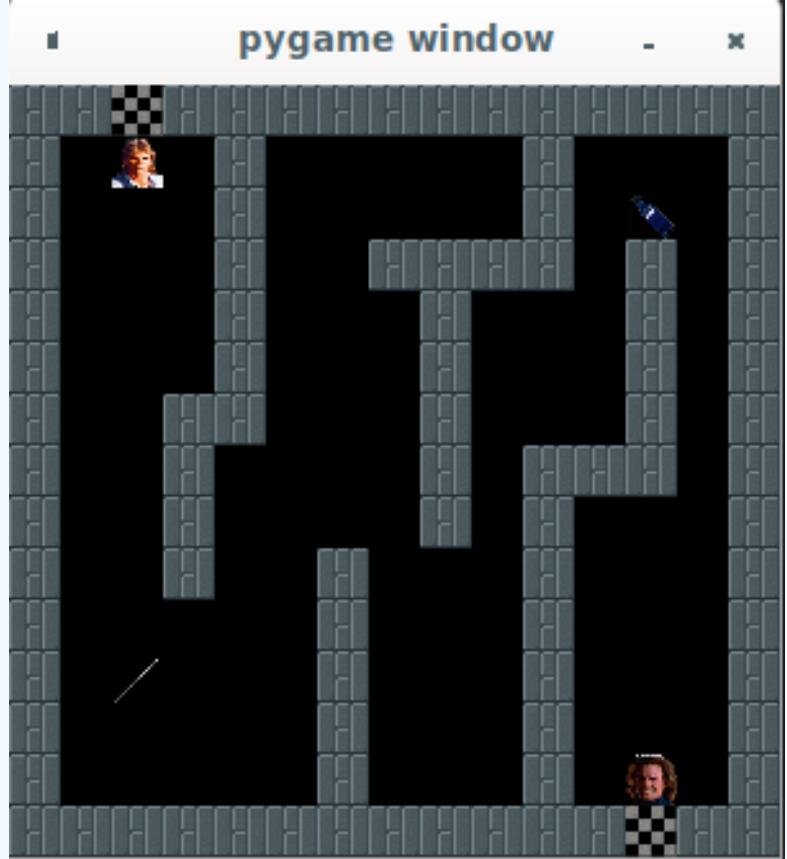
CONTRÔLE :



- Gestion des contrôles du héro.
- Gestion des collisions.
- Gestion de la récupération des objets du labyrinth.

ALGORITHME

AFFICHAGE DU MONDE :



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

DIFFICULTÉS ET SOLUTIONS

CONFIGURATION DU LABYRINTHE

- Création simple d'une configuration.
- Où la contenir?



FICHIER "labyrinth"

- Configuration :

```
1 mmmmmmmmmmmmmmmmm  
2 mxxxxxxxxxxxxxxm  
3 DPxxxxxxxxxxxxx  
4 mxxxxxmmmmxxxxx  
5 mmmmmmmxxxxxxm  
6 mxxxxxxxxxxxxxxm  
7 mxxxxxxxxmmmmmm  
8 mxxmxxxxxxxxxxm  
9 mxxmmmmmmxxxxxx  
10 mxxmxxxxxxxxxxm  
11 mmmmmxxxmmmmmmmm  
12 mxxxxxxxxxxxxxxm  
13 mxxmmmmmmxxxxxGA  
14 mxxxxxxxxxxxxxxm  
15 mmmmmmmmmmmmmmm
```

DIFFICULTÉS ET SOLUTIONS

CRÉATION DU LABYRINTHE

- Génération de la configuration.
- Affichage.



CLASSE "Lab"

- `__Init__():`
[...]
`self.l_wall = []`
`self.l_none = []`
`self.config = []`
- `def generate_lab(self):`
- `def diplay_lab(self):`

DIFFICULTÉS ET SOLUTIONS

REDIMENSION DES IMAGES

Image des objets, du gardien et du héros pas au bon format de sprite.



"pygame.transform.scale"

`pygame.transform.scale(image, (20, 20))`

DIFFICULTÉS ET SOLUTIONS

CRÉATION DES OBJETS

Afficher aléatoirement sans
être sur les murs.



CLASSE "Labobject"

- Position :
`random.choice(labyrinth.l_none)`
- `def draw_me(self):`

DIFFICULTÉS ET SOLUTIONS

CRÉATION DU GARDIEN

- Affichage du gardien.
- Rectangle de son image.



CLASSE "Guardian"

- def draw_me(self):
- def my_rect(self):

DIFFICULTÉS ET SOLUTIONS

CRÉATION DU HÉRO

- Mouvement du héros dans toutes les directions
- Affichage du héros.
- Récupération des objets



CLASSE "Player"

- `__init__(...):`
 - `self.obj1 = False`
 - `self.obj2 = False`
- `def move_right(self):`
- `def move_left(self):`
- `def move_up(self):`
- `def move_down(self):`
- `def draw_me(self):`

DIFFICULTÉS ET SOLUTIONS

COLLISIONS

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



".colliderect() .collidelist()

- Collisions avec les murs :
`player.collidelist(labyrinth.l_wall) != -1`
- Collisions avec les objets :
`player.colliderect(ether.pos):`
`player.colliderect(needle.pos):`
- Collisions avec le gardien :
- `player.colliderect(guardian.my_rect()):`
- vérification des conditions de victoire du jeu.

DIFFICULTÉS ET SOLUTIONS

POSITIONNEMENT DU HÉRO ET DU GARDIEN

Utilisation de leurs lettres et de leurs classes pour les créer dans la configuration du labyrinthe.



"create_character"

```
def create_character(labconfig, lettre, classe, screen1):
    """Draw a character on the labyrinth."""
    i = 0
    for row in labconfig:
        y = 0
        for column in row:
            if column == lettre:
                o = classe((i*20), (y*20), screen1)
                return o
            y += 1
        i += 1
```

DIFFICULTÉS ET SOLUTIONS

IMAGE DU HÉRO AU DÉPART

Image du héro qui reste au départ malgré les déplacements.



"erase_pos_character"

```
def erase_pos_character(labconfig, lettre):
    """Change the letter of the character to a "x" in the configuration of the labyrinth."""
    i = 0
    for row in labconfig:
        y = 0
        for column in row:
            if column == lettre:
                row[y] = 'x'
            return labconfig
            y += 1
        i += 1
```

CONCLUSION

- DIFFICULTÉS DANS LE CHOIX
DE L'ALGORITHME
- PERFECTIONNEMENT DU
PROJET
- RETOUR PERSONNEL

MERCI DE VOTRE ATTENTION.

Céline PELLETIER

A. Phosphorylation of glucose

Phosphorylated sugar molecules do not readily penetrate cell membranes, because there are no specific transmembrane carriers for them. Glucose 6-phosphate is therefore effectively trapped in the cytosolic compartment, thus committing it to further metabolism in the several isozymes of the enzyme hexokinase that catalyze the conversion of glucose to glucose 6-phosphate.

In most tissues, the phosphorylation of glucose is catalyzed by hexokinase, one of three regulatory enzymes of glycolysis (glucokinase, β -phosphofructokinase and pyruvate kinase). Hexokinase has broad substrate specificity and is able to phosphorylate many hexoses in addition to glucose. Hexokinase is the rate-limiting enzyme in glycolysis, producing the first product, glucose 6-phosphate, which is required for further metabolism of this hexose phosphate. Hexokinase has a low K_m (and, therefore, a high affinity) for glucose and, therefore, cannot tolerate glucose in the form of phosphorylated glucose more than the cell can use.

Hexokinase is found in almost all cells, except for parenchymal cells and β cells of the pancreas (also called hexokinase II or hexokinase B). It is the primary enzyme in glycolysis.