



**MON
VIRUS**

FELIX

CAHIER DES CHARGES

01

PROGRAMMATION ORIENTÉE OBJET / FONCTIONNELLE

02

PLATEAU DE JEU COMPOSÉ DE VIRUS

03

LES RÈGLES

03A

CRÉER UN DEFI

03B

DÉPLACEMENTS DES MOLÉCULES

03C

SORTIR LA PIÈCE VIRUS

04

INTERACTION DU JOUEUR

COMMENT J'AI COMMENCÉ ?



1. Le terrain.
2. Les mouvements.
3. Le menu.



4. Le design.
5. Les détails.
6. Le peaufinage.

```
iss Game:  
    def mettreajour_screen(self):  
  
        # Si pas de niveau sélectionne afficher menu  
        if self.stop == 0 or self.stop == 1 or self.stop == 3:  
            self.stop, self.terrain.terrain = menu.afficher_menu(self.stop)  
            if self.terrain.terrain is not None:  
                self.selected = self.element_select()  
  
        # Si niveau sélectionne afficher terrain  
        if self.stop == 2 and self.terrain.terrain is not None:  
  
            if self.startTemps == 0:  
                self.startTemps = time.time()  
  
            self.mouvement(keys, self.selected, self.tab)  
            self.terrain.draw(self.screen, self.selected, self.tab)  
  
            # Si le virus est dans le coin en haut à gauche afficher "you win"  
            if self.terrain.terrain[0][0] == "car" and self.terrain.terrain[1][0] == "car":  
  
                self.endTemps = time.time()  
  
                # image de fond  
                self.screen.blit(pygame.transform.scale(pygame.image.load("./PygameAssets/Tal  
  
                # écriture ta gagné  
                self.font = pygame.font.Font('freesansbold.ttf', 32)  
                self.textWin = self.font.render('GG', True, (173,216,230))  
                self.text_rect = self.textWin.get_rect(center=(1920 // 2, 1080 // 2 - 200))  
                self.screen.blit(self.textWin, self.text_rect)  
  
                # écriture nbr de coup  
                self.font = pygame.font.Font('freesansbold.ttf', 32)  
                self.textCoup = self.font.render(f'Nombre de move : {self.nbrCoup}', True, (173,216,230))  
                self.text_rect_Coup = self.textCoup.get_rect(center=(1920 // 2, 1080 // 2 + 200))  
                self.screen.blit(self.textCoup, self.text_rect_Coup)  
  
                # écriture temps  
                if self.stopTemps == 0:  
                    self.tempstotal = self.endTemps - self.startTemps  
                    self.stopTemps += 1  
                if self.tempstotal > 60:  
                    self.ecriTempstotal = f'{self.tempstotal//60:.0f} min {self.tempstotal%60:.0f} s'
```

DIAGRAMME DE CLASS

Terrain

Attribute

terrain : None / list de string

Methode

load_assets()
load_globules()
draw()
draw_element()
buttonResetAndMenu()

Game

Attribute

AllGoodRecursion : boolean
reseauTerrain : None/liste de string
select : dict
tab : int
stopTemps, startTemps : int
nbrCoup : int

Methode

run()
mettreajour_screen()
element_select()
mouvement()
moveMolecule()
allvalid()
isValid()
poussageRecursive()

Menu

Attribute

terrains : list de list de string
stop : int

Methode

backgroundPrint()
buttonscharge()
afficher_menu()

PyGame

Attribute

screen : width, height
clock : float
keys : None ou touche
randBloop: liste de .wav
backgroundSong : .wav

Methode

display()
image()
transform()
event()
key()
time()
mixer()
font()

LE CODE

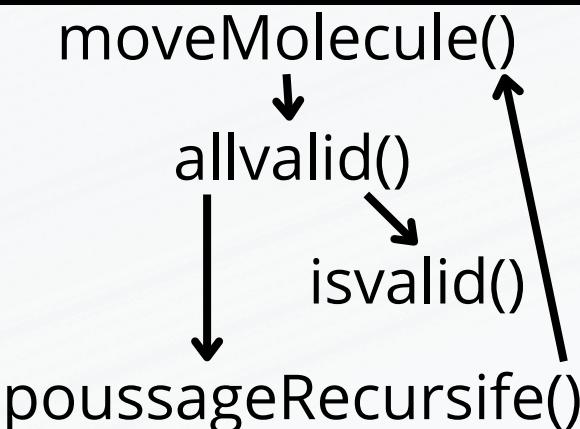
Forme d'un terrain

```
[[ "y", "w", "w", "x"],  
 ["x", "x", "x"],  
 ["y", "w", "u", "g"],  
 ["u", "u", "g"],  
 ["x", "x", "d", "d"],  
 ["x", "car", "d"],  
 ["x", "x", "car", "x]],
```

- Manipulation simple
- Terrain explicite
- Utile pour les tests
- Faciliter l'ajout de terrain
- Gestion des terrains et interactions

- moveMolecule() prend 2 paramètres : la pièce et le mouvement. Elle bouge la pièce si allvalid() retourne true.
- allvalid() vérifie, pour toutes les molécules, la case d'après avec isvalid().
- isvalid() vérifie que la case est dans le terrain et que ce n'est pas une case déjà occupée.
- poussageRecurcive() lance moveMolecule() s'il y a une molécule dans la direction voulue.

Déplacements



LES RECHERCHES

- Les collisions des boutons en utilisant rect

```
self.rectResetGame = pygame.image.load("./PygameAssets/Buttons/RESTART.png").get_rect(topleft=(1920, 1040))
self.rectMenuGame = pygame.image.load("./PygameAssets/Buttons/MENU.png").get_rect(topleft=(1920//2, 1040//2))

# Boutons en Game
if self.rectMenuGame.collidepoint(mouse_pos):
    self.stop, self.stopTemps, self.startTemps, self.nbrCoup = 0, 0, 0, 0 # initialisation
if self.rectResetGame.collidepoint(mouse_pos):
    self.terrain.terrain = [row[:] for row in self.reseTerrain] # définir pa
    self.stopTemps, self.startTemps, self.nbrCoup = 0, 0, 0 # reset le temps
```

- Désynchroniser des valeurs associés

```
self.reseTerrain = [row[:] for row in self.terrain.terrain]
```

```
self.terrain.terrain = [row[:] for row in self.reseTerrain]
```

- Mettre du son avec pygame

```
# INITIATION DU MODULE DE MIXAGE (SON)
pygame.mixer.init()

# Chargement des fichier audio (.wav)
self.randBloop = [pygame.mixer.Sound('sounds/bloopl.wav'), pygame.mi
backgroundSong = pygame.mixer.Sound('sounds/EdgeOfSanity.wav')
backgroundSong.play(loops=50)
```

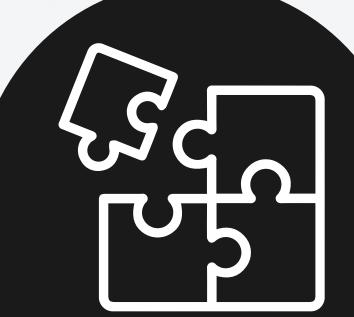
- La condition Si aucune touche n'est pressée

```
if not any(keys):
    self.Good = True

if keys[pygame.K_a] and self.Good == True:
    self.directionListe = 0
    self.AllGoodRecursion = True
    self.otherPiece, self.alreadyTest = [], []
    self.moveMolecule(selected[tab], [-1, -1, -1, 0], 1)
    self.Good = False
```

PRÉSENTATION DU JEU

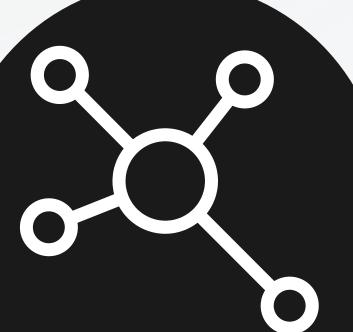
CONCLUSION



Interaction entre class

Utilisation de pygame dans
une class

**GESTION DE
CLASS**



Utiliser plusieurs fichiers/
dossier pour un projet

Interaction entre fichiers

**STRUCTURER UN
PROJET COMPLEXE**



Mettre du son

Mauvaise ordre de passage
dans le terrain

**PROBLEMES
RENCONTRÉS**

**MERCI POUR VOTRE
ÉCOUTE !**

