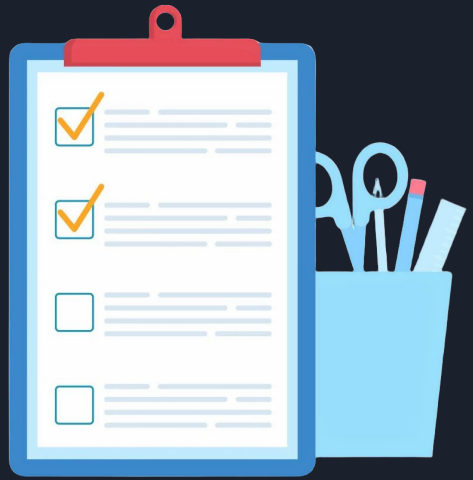
Mini jeu Anti Virus

Projet de NSI Hippolyte Jombart

Cahier des charges : Refaire le jeu d’Anti-Virus



- Utilisé la programmation orientée objet (et fonctionnelle)
- Implémenter un système de déplacement des pièces
- Utiliser uniquement les 8 molécules fournis
- Utiliser la bibliothèque pygame
- Un premier rendu le 7/11/2024
- Respecter les règles du jeu de base

Étapes de développement

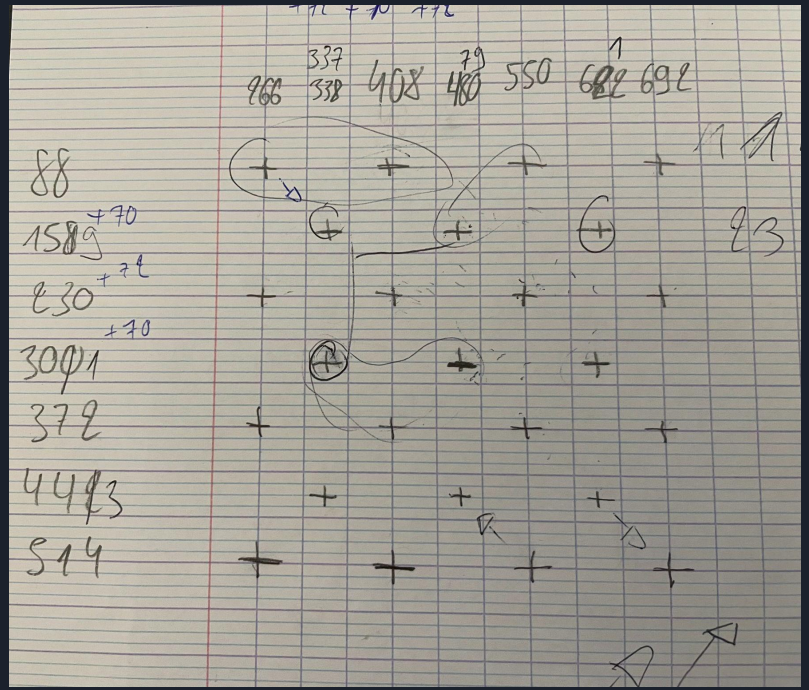
1. Menu
2. Partie graphique des six niveaux
3. Implémentation du déplacement des pièces
4. Implémentation des “collisions”
5. Partie graphique des six niveaux

Le diagramme de classe

Zoom sur le travail personnel

```
while InMenu or CurrentLevel==7 and running:
    screen.blit(bg_menu,(0,0))
    screen.blit(Button.get_img(Starter),Button.get_coo(Starter))
    screen.blit(Button.get_img(Junior),Button.get_coo(Junior))
    screen.blit(Button.get_img(Expert),Button.get_coo(Expert))
    screen.blit(Button.get_img(Master),Button.get_coo(Master))
    screen.blit(Button.get_img(End),Button.get_coo(End))
    screen.blit(logo,(50,50))
    screen.blit(Indicator,(309,140))

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        if event.type == pygame.MOUSEBUTTONDOWN:
            for but in ListBut:
                if Button.get_page(but)=="Menu" or Button.get_page(but)=="All" :
                    if pygame.mouse.get_pos()[0] > Button.get_coo(but)[0] and pygame.mouse.get_pos()[1] > Button.get_coo(but)[1]
                        if Button.get_path(but)[0]==False:
                            running=False
                        elif Button.get_path(but)[1]==False:
                            if Button.get_path(but)[2]!=0:
                                bo=False
                                screen.blit(WaitingBG,(0,0))
                                pygame.display.flip()
                                time.sleep(0.5)
                                CurrentLevel=Button.get_path(but)[2]
                                InMenu = False
    pygame.display.flip()
```



```
if bo == False:
    Blok.set_coo(Virus,(358,180))
    Blok.set_coo(BlocV,(287,109))
    Blok.set_coo(Bloc1,(500,322))
    Blok.set_gridpos(Virus,[[(2,1),(3,1)],[(3,1)],[(2,1),(3,1)],[(2,1)],[(2,1),(3,1)]])
    Blok.set_gridpos(BlocV,[[(1,0),(1,1),(3,0)],[(1,0),(1,1),(3,0)],[(1,0),(1,1),(3,0)],[(1,0),(1,1),(3,0)],[(1,0),(1,1),(3,0)]])
    Blok.set_gridpos(Bloc1,[(4,4)])
    grid=[[True,True,True,True,None],[False,False,True,None],[True,False,True,True,None],[True,False,True,None],[True,True,False,True,None],[True,True,True,None],[True,True,T
bo = True
```

```
if key_pressed == True:
    if keys[pygame.K_RIGHT] and keys[pygame.K_DOWN]:
        for i in range (len(Blok.get_gridpos(movingB)[1])):
            if Blok.get_gridpos(movingB)[1][i][0]%2!=0:
                if grid[Blok.get_gridpos(movingB)[1][i][0]+1][Blok.get_gridpos(movingB)[1][i][1]+1]==True and moving:
                    moving=True
                else:
                    moving=False
            else:
                if grid[Blok.get_gridpos(movingB)[1][i][0]+1][Blok.get_gridpos(movingB)[1][i][1]]==True and moving:
                    moving=True
                else:
                    moving=False
        if moving:
            for i in range(len(Blok.get_gridpos(movingB))):
                for j in range(len(Blok.get_gridpos(movingB)[i])):
                    grid[Blok.get_gridpos(movingB)[i][j][0]][Blok.get_gridpos(movingB)[i][j][1]]=True
            for n in range(len(Blok.get_gridpos(movingB))):
                for h in range(len(Blok.get_gridpos(movingB)[n])):
                    if Blok.get_gridpos(movingB)[n][h][0]%2!=0:
                        grid[Blok.get_gridpos(movingB)[n][h][0]+1][Blok.get_gridpos(movingB)[n][h][1]+1]=False
                        Blok.set_gridposSpe(movingB,n,h,(Blok.get_gridpos(movingB)[n][h][0]+1,Blok.get_gridpos(movingB)[n][h][1]+1))
                    else:
                        grid[Blok.get_gridpos(movingB)[n][h][0]+1][Blok.get_gridpos(movingB)[n][h][1]]=False
                        Blok.set_gridposSpe(movingB,n,h,(Blok.get_gridpos(movingB)[n][h][0]+1,Blok.get_gridpos(movingB)[n][h][1]))
            Blok.set_coo(movingB,(Blok.get_coo(movingB)[0]+71,Blok.get_coo(movingB)[1]+71))
            key_pressed = False
```

Le code recyclé

```python
import pygame

pygame.init()

# Dimensions de la fenêtre
screen = pygame.display.set_mode((800, 600))

# Couleur du fond (bleu)
blue = (0, 255, 0)

# Boucle principale
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    # Remplir l'écran avec la couleur bleue
    screen.fill(blue)

    pygame.display.flip()

pygame.quit()
```

L'IA pas si indispensable que ça