

# Projet POO

BIZEL Edgar, DAHOUN Neil, SERAFINO Jean-Adelin

20 juin 2021

## 1 Utilisation

Lancez le fichier game.py. Pygame 2.x est requis, ainsi que Python 3.9

Contrôles :

- Déplacement : ZQSD
- Items : Touches numériques (Shift + & sur clavier azerty, etc)
- Passer un tour : espace
- Dormir : L
- Détruire les objets équipés : T

## 2 Fonctionnement du jeu

Le point d'entrée se trouve dans main.py. Il initialise les composants de pygame et du jeu, et appelle la fonction responsable du menu principal.

Le menu principal se trouve dans start\_menu.py. Il permet d'accéder aux options et de lancer le jeu. Il s'adapte à tout écran

Le jeu est lancé avec la classe Game, dans le fichier game.py. Sa méthode principale est run. Cette fonction initialise le jeu. Puis elle gère les événements, l'affichage et les updates.

Les updates consistent principalement à déplacer les personnages (move dans Map) et à gérer leurs interactions (fichier meet.py)

## 3 Listes des fonctionnalités et complexité estimée

### Complexité totale : 25

Note : Nous avons dû évaluer par nous-même la complexité des fonctionnalités, étant donné que beaucoup n'étaient pas dans la liste.

Nous ne sommes pas sûrs de la notation. Mais les fonctionnalités paraissent largement suffisantes pour avoir 8/8.

### 3.1 Refonte complète du code **Comp. 4**

Une refonte complète du code a été effectuée. Elle a permis de séparer le code dans plusieurs fichiers, de supprimer le singleton, d'éviter les variables globales...

L'utilisation du typing, avec une extension comme mypy ou Pylance, permet également de détecter les erreurs de type avant le lancement, directement dans l'éditeur.

### 3.2 G.I. Interface graphique **Comp. 4**

Fichiers principaux : hud, images, start\_menu, ui\_elements

### 3.3 Interface graphique : responsive **Comp. 4**

L'interface graphique peut gérer un écran de n'importe quelle taille.  
Elle permet également de régler le nombre de cases, et adaptera la taille des textures.  
Le jeu est dorénavant un rectangle, qui s'adapte lui aussi à l'écran.  
Les spritesheets sont également supportés, grâce à la classe SpriteSheet

### 3.4 G.II. Étages **Comp. 1**

Classe Stairs. Fonctions `_meet_stairs_element` et `Game._update`

### 3.5 G.III. XP et niveau **Comp. 1**

Le héros peut monter en niveau. Fonction `Hero.add_xp` dans `creature.py`

### 3.6 G.IV. Inventaire limité **Comp. 1**

Le héros a un inventaire limité. Les objets identiques se "stack", leur durabilité est combinée. Il peut détruire les objets équipés avec T  
Fonction `Inventory.add`

### 3.7 G.V. Pathfinding très basique **Comp. 1**

Le pathfinding est loin d'être suffisant, mais il permet aux monstres de se retrouver bloqués moins souvent  
Fonction `move_all_monsters` dans `map`

### 3.8 Sons **Comp. 2**

Des sons et musiques ont été ajoutées au jeu.  
Fichier principal : `sounds`

### 3.9 A.I. Jet **Comp. 2**

L'arc peut tirer des projectiles. Ceux-ci sont détruits lorsqu'ils rencontrent un obstacle. Ils peuvent rester au sol en tant que décoration, mais ne pourront pas être ramassés.  
Fonction `_ranged_weapon_usage` dans `element_list`

### 3.10 A.II. Repos **Comp. 1**

Fonction `Game.sleep_hero`

### 3.11 O.II Armes **Comp. 1**

Fichiers principaux : `inventory`, `equipment`

### 3.12 O.IV Armures **Comp. 1**

Fichiers principaux : `inventory`, `equipment`

### 3.13 O.VI. Solidité **Comp. 1**

Fichier principal : `equipment`

### **3.14 M. IV. Invisible Comp. 1**

Ajoute le fantôme, un monstre invisible jusqu'à qu'il attaque ou soit attaqué.  
Il dispose de sa propre classe, Ghost dans creature.py

### **3.15 Nouveau contenu Comp. 0**

Une vingtaine d'éléments ont été ajoutés : monstres, arcs, épées...