

# Projet TBA

Matthieu ZOZOR

Alice LIN

Dimanche 19 Janvier 2025

<b>1. Guide Utilisateur.....</b>	<b>2</b>
1.1 Introduction.....	2
1.2 Commandes du Jeu.....	2
1.3 Règles du jeu.....	3
<b>2. Guide Développeur.....</b>	<b>3</b>
<b>3. Perspectives de Développement.....</b>	<b>5</b>
3.1 Améliorations possibles.....	5
3.2 Optimisations techniques.....	5

# 1. Guide Utilisateur

## 1.1 Introduction

### Contexte du Jeu :

"L'Ascension du Boss" est un jeu d'aventure où le joueur incarne un personnage déterminé à renverser Diavolo, le tyran à la tête de la mafia *Passione*. Plutôt que de rechercher le pouvoir, le joueur cherche à protéger les innocents en devenant le Boss, le chef légendaire de la mafia qui œuvre pour le bien.

Tout au long de l'aventure, le joueur devra explorer des zones contrôlées par *Passione*, ramasser des objets puissants et les utiliser pour affronter les ennemis. Ces objets cachés permettront de combattre les membres de la mafia et de vaincre Diavolo. Le jeu met l'accent sur la stratégie et la gestion des ressources pour surmonter les obstacles.

Le joueur devra également faire face à des alliances et des trahisons tout en suivant son idéal de justice. La quête pour renverser Diavolo est semée d'embûches, et chaque choix influencera le destin du monde.

Jeu inspiré de l'anime Jojo's Bizarre Adventure.

### But du jeu :

Dans "**L'Ascension du Boss**", le joueur doit renverser Diavolo, chef tyrannique de *Passione*, et devenir le **Boss** pour protéger les innocents. En explorant et en collectant des objets puissants, le joueur affronte des ennemis pour restaurer la justice et mettre fin à la tyrannie.

## 1.2 Commandes du Jeu

Les commandes du jeu se font directement depuis le terminal.

Pour lancer le jeu il faut taper la commande : **python game.py**

Vous pouvez utiliser les commandes suivantes pour interagir avec l'environnement, gérer votre inventaire, et avancer dans l'histoire :

- **help** : Affiche une aide détaillée listant toutes les commandes disponibles.
- **quit** : Quitte le jeu immédiatement.
- **go <direction>** : Permet de se déplacer dans une direction cardinale. Les directions valides sont :
  - N : Nord
  - E : Est
  - S : Sud
  - O : Ouest
  - U : Up
  - D : Down

*Exemple* : **go N** pour se déplacer vers le Nord.

- **history** : Affiche l'historique des pièces visitées.
- **back** : Retourne à la pièce précédente.
- **check** : Affiche l'inventaire du joueur.
- **look** : Affiche les objets présents dans la pièce actuelle.
- **take <objet>** : Permet de prendre un objet et de l'ajouter à l'inventaire.  
*Exemple : take arc pour prendre l'arc.*
- **drop <objet>** : Permet de poser un objet de l'inventaire dans la pièce actuelle.  
*Exemple : drop arc pour poser l'arc.*
- **talk <personnage>** : Engage une conversation avec un personnage non-joueur.  
*Exemple : talk Polnareff pour parler à Polnareff.*
- **fight <ennemi>** :  
Engage un combat avec un ennemi ou un PNJ spécifié.  
*Exemple : fight Doppio pour combattre Doppio.*

### 1.3 Règles du jeu

#### Conditions de victoire :

Pour gagner le jeu, il faut trouver et battre le boss du jeu.

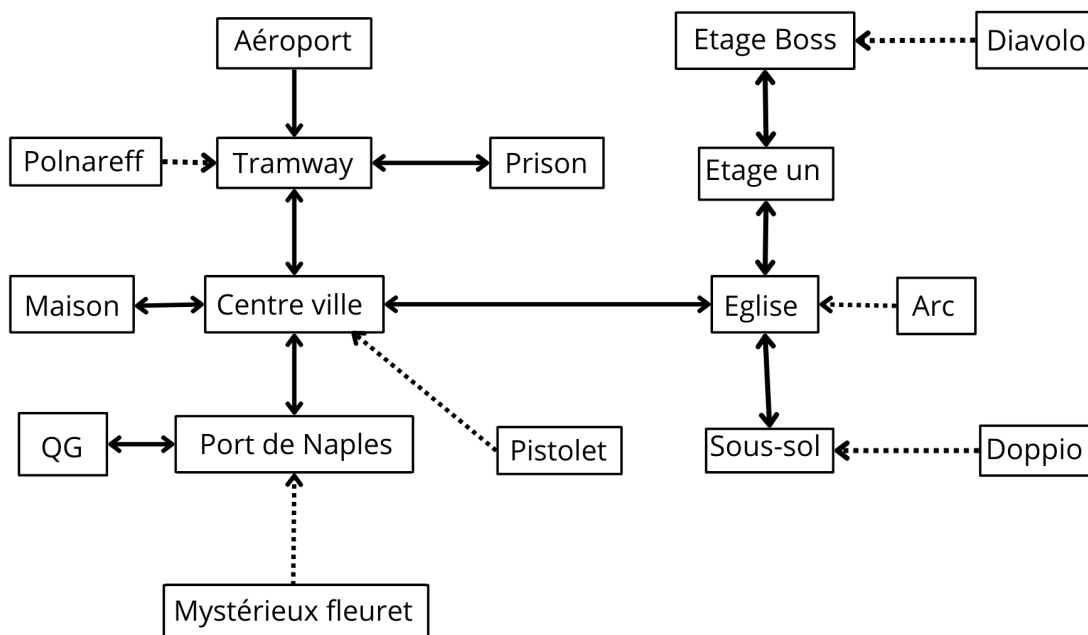
#### Conditions de défaite :

On perd le jeu dès qu'on perd contre un ennemi dont on a décidé d'engager le combat.

#### Objets :

Pour battre certains ennemis il faut trouver des objets spécifiques pour les vaincre.

Voici la carte du jeu :

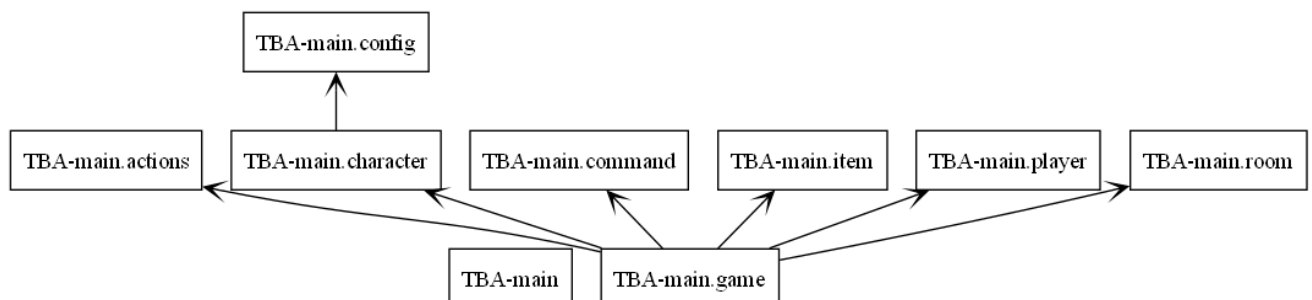
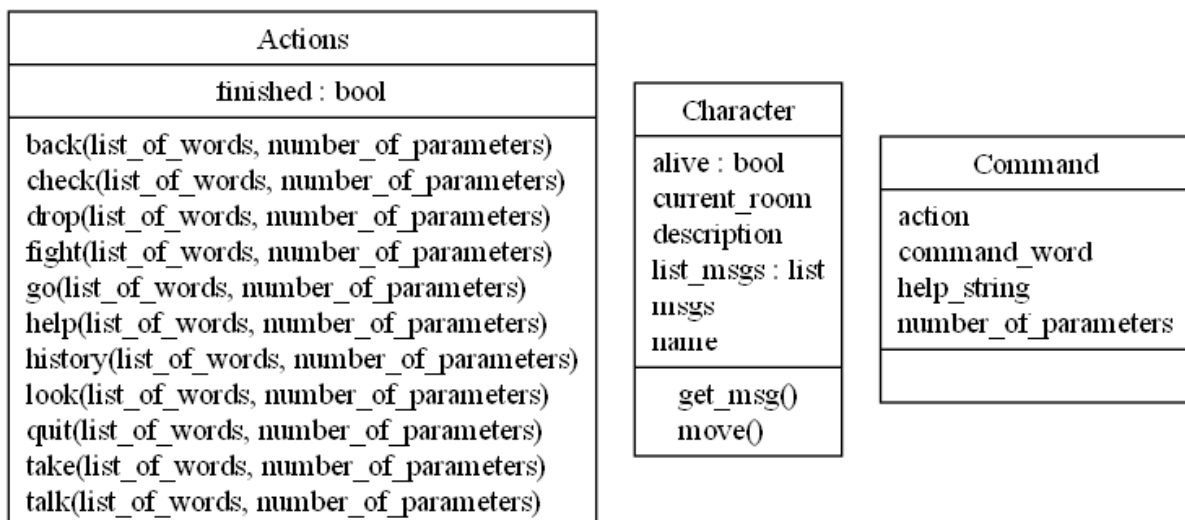
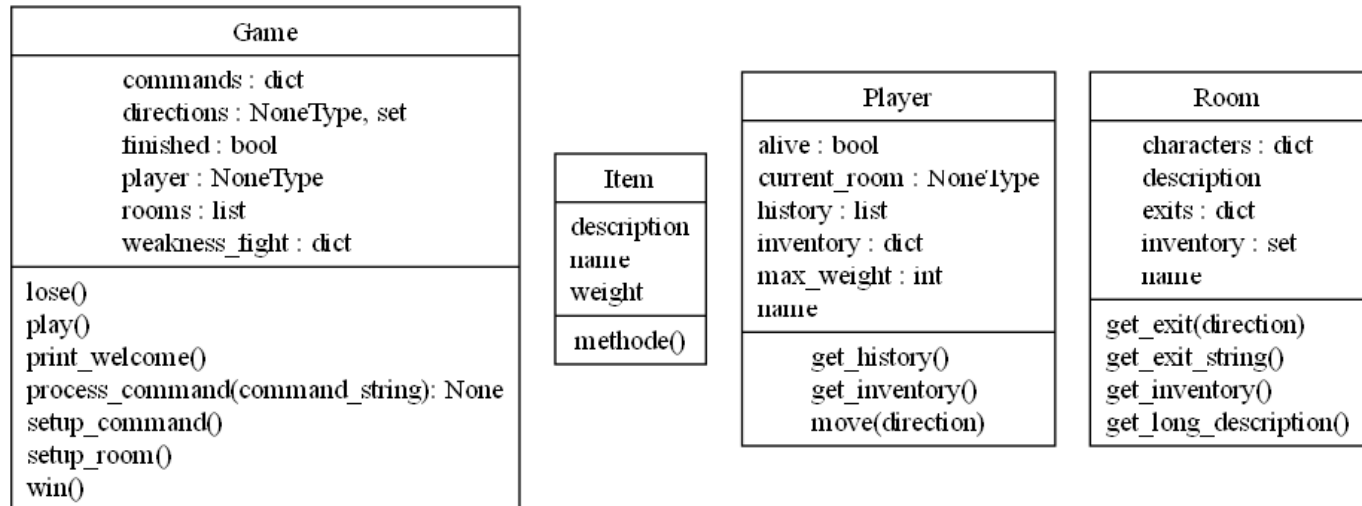


## 2. Guide Développeur

### Diagramme de Classes

config.py est utilisé dans room.py, game.py, player.py, character.py, actions.py, item.py

game utilise tous les fichiers, config est utilisé dans tous les fichiers en tant qu'outil pour importer DEBUG.



### **3. Perspectives de Développement**

#### **3.1 Améliorations possibles**

On pourrait faire un système de point de vie pour combattre les ennemis, pour que ce soit moins facile à gagner le combat.

On pourrait aussi ajouter un système de niveau, par exemple on peut battre certains ennemi à un certain niveau puis ainsi vaincu on gagne des niveaux pour pouvoir avoir le niveau pour vaincre le boss. Il faudrait aussi rajouter des ennemis pour mettre en difficulté le joueur. On pourrait aussi faire une partie graphique pour mieux visualiser le jeu.

#### **3.2 Optimisations techniques**

Si nous avons eu les capacités nécessaires, nous aurions pu essayer d'optimiser un maximum le code de ce jeu. On pourrait utiliser le fichier `conf.py` pour stocker toutes les instances au lieu de les avoir dans le fichier `game.py`. En utilisant la propriété d'hérédité des classes avec les classes "player" et "character" nous pourrions éviter la répétition des attributs comme "nom", "alive", "current\_room" et potentiellement une description.