

# **RAPPORT PROJET PYTHON :**

## **Les Secrets du Manoir : À la Recherche de la Fortune Perdue**

## **Table des matières**

Description et but du jeu.....	
Guide utilisateur.....	
Guide développeur.....	
Perspective.....	

## **Description et but du jeu**

### **Description:**

Plongez dans l'obscurité mystérieuse du manoir ancestral, un héritage d'une famille extraordinairement riche. Les rumeurs murmurent que des fantômes tourmentés hantent chaque recoin de cette demeure majestueuse. Ce RPG textuel vous invite à incarner un aventurier courageux, attiré par la promesse d'une fortune légendaire, dissimulée dans les profondeurs du manoir hanté.

Vos compétences seront mises à l'épreuve à travers des explorations de sombres corridors, des résolutions d'énigmes et des interactions avec des personnages surnaturels. Pour progresser, il faudra récupérer des objets précieux, découvrir des indices cruciaux.

### **But du jeu:**

Le joueur se déplace à l'aide du terminal, de salle en salle, pour résoudre les énigmes et doit équiper dans son inventaire, des objets disponibles dans chaque salle, utiles ou non pour ouvrir le coffre-fort. Dans chaque pièce, se trouve une énigme à résoudre pour accéder à la salle suivante. Le but; résoudre les énigmes de chaque salle, peaufiner son inventaire afin d'ouvrir le coffre-fort et éviter les pièges.

### **Comment jouer au jeu?**

Vous entrez dans le Manoir Cabot, et vous êtes piégé. Aucune issue possible. Vous devrez résoudre des énigmes, interagir avec des personnages

fantomatiques, collecter des objets, et découvrir les secrets enfouis dans chaque pièce pour ouvrir le coffre-fort et vous échapper.

### Objectif du jeu :

Votre objectif ultime est d'ouvrir le coffre-fort du manoir pour percer ses mystères. Pour cela, vous devrez :

- Explorer les différentes salles du manoir.
- Collecter des objets utiles pour résoudre des énigmes.
- Écrire les **trois mots magiques** dans le cahier de notes.
- Obtenir la **clé magique** et utiliser le code pour déverrouiller le coffre.
- Tapez le bon code de la porte. (Attention, vous avez 3 tentatives)

Commandes	Description
<b>help</b>	Affiche toutes les commandes disponibles.
<b>go &lt;direction&gt;</b>	Se déplacer dans une direction donnée (exemple : <b>go NE</b> ou <b>go SW</b> ).
<b>look</b>	Inspecte la pièce actuelle pour trouver des objets ou des personnages.
<b>take &lt;objet&gt;</b>	Ramasse un objet dans la pièce et l'ajoute à votre inventaire.
<b>drop &lt;objet&gt;</b>	Déposez un objet de votre inventaire dans la pièce actuelle.

<b>check</b>	Affichez le contenu de votre inventaire.
<b>talk &lt;nom&gt;</b>	Parle à un personnage dans la pièce.
<b>write &lt;objet&gt; &lt;texte&gt;</b>	Écrire du texte dans un objet spécifique (comme le cahier de notes).
<b>charge_beamer</b>	Charge le beamer avec la pièce actuelle (si vous le possédez).
<b>use_beamer</b>	Téléporte le joueur à la pièce mémorisée par le beamer (si chargé).
<b>back</b>	Retourne à la pièce précédente visitée.
<b>history</b>	Affiche l'historique des salles visitées.
<b>open_coffre</b>	Ouvre le coffre-fort (si vous avez la clé magique).
<b>enter_code &lt;code&gt;</b>	Permet d'entrer le code pour déverrouiller la porte électrique dans le Hall d'Entrée.

**Les étapes principales :**

### **1. Explorer les salles :**

★ Déplacez-vous avec la commande **go <direction>**.

- ★ Inspectez la pièce avec la commande **look** pour découvrir des objets ou parler aux personnages.

## **2. Interagir avec les personnages :**

- ★ Utilisez la commande **talk <nom>** pour parler aux personnages présents dans la pièce.
- ★ Certains vous donneront des indices sur les mots magiques ou des conseils pour progresser.

## **3. Trouver et utiliser les objets :**

- ★ Ramassez des objets avec la commande **take <objet>** et ajoutez-les à votre inventaire.
- ★ Certains objets sont nécessaires pour progresser, comme le cahier de notes.

## **4. Trouver les trois mots magiques :**

- ★ Écrivez les mots magiques dans le cahier de notes en utilisant la commande **write cahier\_de\_notes <mot>**.
- ★ Les mots magiques sont liés à des indices donnés par les personnages et des objets spécifiques.

## **5. Obtenir la clé magique :**

- ★ Une fois les trois mots magiques inscrits dans le cahier de notes, la clé magique apparaîtra dans votre inventaire.

## **6. Déverrouiller le coffre-fort :**

- ★ Parlez au tableau mystérieux pour comprendre comment ouvrir le coffre-fort.
- ★ Utilisez la commande `open_coffre` avec la clé magique.

## 7. Sortir du manoir :

- ★ Une fois le coffre ouvert, récupérez le code.
- ★ Rendez-vous dans le Hall d'Entrée et entrez le code avec `enter_code <code>` pour déverrouiller la porte et gagner le jeu.

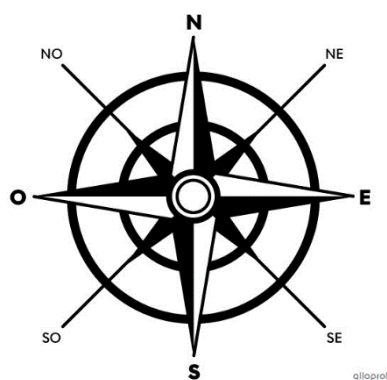
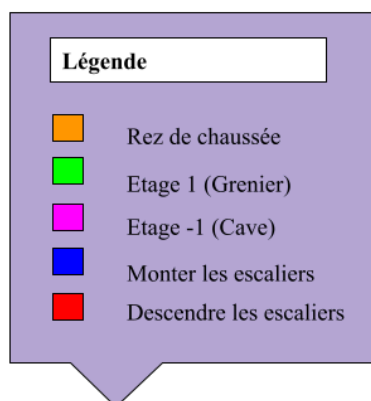
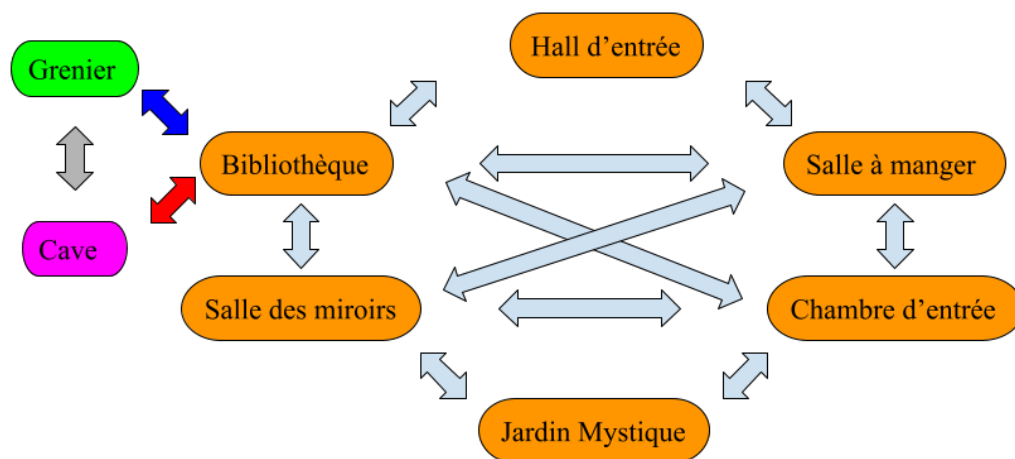
## Conseils utiles :

- ★ **Soyez attentif aux indices** : Parlez aux personnages et inspectez bien les objets, ils contiennent des informations cruciales.
- ★ **Équipez-vous intelligemment** : Certains objets ne sont pas essentiels, mais peuvent vous aider dans votre aventure.
- ★ **Utilisez votre cahier de notes** : C'est un outil clé pour rassembler les mots magiques et progresser dans le jeu.

## Conditions de victoire :

- ★ Avoir inscrit les **trois mots magiques** dans le cahier de notes.
- ★ Utiliser la **clé magique** pour ouvrir le coffre-fort et récupérer le code.
- ★ Entrer le **code correct** dans le Hall d'Entrée pour déverrouiller la porte et échapper au manoir.

## Carte des salles





## Liste des salles constituant la carte :

**Hall d'entrée :** L'entrée majestueuse où chaque voyageur intrépide commence son périple. Des ombres dansent, les tapis fanés racontent des histoires du passé, et les murs semblent murmurer des secrets enfouis.

**Bibliothèque :** Un sanctuaire de connaissances oubliées, la bibliothèque regorge de grimoires poussiéreux et de tomes anciens. Les murmures des érudits du passé semblent résonner, défiant le silence de l'obscurité.

**Salle à Manger :** Un espace jadis animé, où des chaises vides semblent attendre des convives invisibles. Les bruits étouffés d'anciens banquets résonnent, faisant écho à une époque révolue.

**Chambre Hantée :** Un lieu où les ombres prennent vie et des murmures déchirants résonnent. Des esprits tourmentés errent dans les couloirs, laissant derrière eux des visions fugaces du passé.

**Cave :** Un labyrinthe obscur de passages humides, où des gouttes d'eau résonnent comme des murmures surnaturels. Les secrets les plus sombres du manoir semblent être enfouis dans ses profondeurs.

**Jardin mystique :** Un extérieur énigmatique où les arbres murmurent des histoires oubliées. Des énigmes naturelles et des rencontres mystiques attendent ceux qui osent s'aventurer dans ce jardin éthéré.

**Salle des Miroirs :** Un espace où la réalité semble se déformer. Les miroirs reflètent des images déformées et des reflets fantomatiques, mettant à l'épreuve la perception des aventuriers.

**Grenier :** Un royaume poussiéreux rempli d'objets oubliés. Des indices cruciaux peuvent être dissimulés parmi les reliques du passé, attendant d'être découverts par des yeux attentifs.

### **Coffre-Fort :**

Le saint Graal de l'exploration, le coffre-fort renferme la richesse incommensurable de la famille décédée. Seuls les plus ingénieux et courageux pourront percer ses secrets.

## Liste d'objets disposés dans le manoir :

### Bibliothèque:

**Aspirateur:** Un aspirateur mystérieux pouvant tout aspirer. Son utilité reste énigmatique.

### Salle à Manger

**Couteau:** Un simple couteau de cuisine. Peut être utilisé pour résoudre certaines énigmes ou pour interagir avec des objets spécifiques.

### Chambre Hantée

**Cahier de Notes:** Un cahier vide dans lequel le joueur doit écrire les trois mots magiques (**La Russie, Rose Morte, Année 1894**) pour obtenir la **clé magique**.

*Utilisation :* Nécessaire pour progresser vers l'objectif final.

**Calculatrice:** Une calculatrice scientifique, ajoutée comme élément de décor ou pour des énigmes potentielles.

### Cave

**Bouteille de Vin:** Une bouteille de vin datant de l'année **1894**. Cet objet contient un indice pour l'un des mots magiques nécessaires.

## Jardin Mystique

**Pelle:** Une petite pelle permettant de déterrer des objets cachés, notamment la **Rose Morte**, un autre mot magique.

## Salle des Miroirs

**Poupée Russe:** Une poupée russe mystérieuse et lourde. Elle représente **La Russie**, un des mots magiques nécessaires pour progresser.

## Grenier

**Coffre:** Un coffre fermé, contenant un trésor. Il ne peut être ouvert qu'avec la **clé magique** obtenue après avoir découvert les mots magiques.

**Beamer:** Un appareil permettant de se téléporter dans une pièce mémorisée. Très utile pour les déplacements rapides.

## Inventaire de Départ

**Lampe UV :** Une lampe permettant de révéler des indices ou des messages cachés dans certaines pièces sombres.

**Lettre:** *Description :* Une lettre mystérieuse destinée à Louis Cabot, datée de **1824**.

**Carte:** : Une ancienne carte du manoir, révélant des passages secrets pour aider le joueur à planifier ses déplacements.

## Personnages non joueurs :

Nom du Personnage	Rôle	Message Transmis
Roberta	Réceptionniste	"Bienvenue, aventurier ! Vous êtes piégé dans le manoir. Trouvez un moyen de percer ses secrets pour vous échapper."
Alfred	Le Gardien	"Le premier mot est inspiré des mystères d'un vaste pays à l'Est.",  "Le second mot évoque la beauté fanée d'une fleur.", "Le dernier mot est gravé sur une bouteille qui porte une date ancienne."
Maurice	Fantome	"Vous devez écrire les trois mots magiques dans le cahier de notes pour percer les mystères du manoir."
Le tableau qui parle de Barbe Blanche	Tableau	"Pour ouvrir ce coffre, vous devrez connaître les trois mots magiques."

## Diagramme des classes :

### Structure UML des classes principales :

#### Classe Game:

Gestion Globale du jeu

#### Attributs:

#### Rooms:

```
# Création des pièces

    Hall_d_Entrée = Room("Hall_d_Entrée", "dans un grand hall
sombre avec un lustre suspendu. Des portraits effrayants ornent les
murs. Une porte électrique massive se trouve ici, verrouillée par un
code mystérieux.", game=self)

    self.rooms.append(Hall_d_Entrée)

    Bibliothèque = Room("Bibliothèque", "dans une bibliothèque
poussièreuse remplie de vieux livres. Des chandelles vacillantes
éclairent la pièce.", game=self)

    Grenier = Room("Grenier", "dans un grenier sombre rempli de
vieux objets poussiéreux et de malles abandonnées.", game=self)

    self.rooms.append(Grenier)
```

#### Commands

```
help = Command("help", " : afficher cette aide", Actions.help, 0)
self.commands["help"] = help

quit = Command("quit", " : quitter le jeu", Actions.quit, 0)
self.commands["quit"] = quit
```

## Player:

```
# Ajoutez des objets de départ au joueur

self.player.inventory["lettre"] = Item("lettre", "Une lettre destinée
à Louis Cabot datée de 1824", 0.01)

self.player.inventory["carte"] = Item("carte", "Une carte ancienne
révélant des passages secrets dans le manoir", 0.5)

self.player.inventory["lampe_uv"] = Item("lampe_uv", "Une lampe UV
pour révéler des messages cachés", 1)
```

## Root (interface graphique):

```
# Configuration de l'interface graphique

self.root = tk.Tk()

self.root.title("Manoir Cabot")

self.root.geometry("600x400")
```

## Méthodes:

### Start\_gui:

```
def start_gui(self):  
    """Démontre l'interface graphique."""  
    self.setup() # Configure le jeu  
    if hasattr(self, "output_text") and self.output_text is not  
None:  
        self.display_message("""Bienvenue, aventurier courageux !  
        """)  
        Introduction  
        """) # Message d'accueil  
    self.root.mainloop() # Démontre la boucle principale de Tkinter
```

### Process\_command:

```
def process_command(self):  
    command = self.input_entry.get().strip() # Récupère la  
commande entrée  
    self.input_entry.delete(0, "end") # Efface l'entrée  
    if not command:  
        self.display_message("Veuillez entrer une commande.")  
        return  
    self.display_message(f"> {command}") # Affiche la commande  
entrée
```



```
self.execute_command(command) # Exécute la commande

if self.finished:
    self.display_message("Le jeu est terminé ! Merci d'avoir
joué.")
    self.submit_button.config(state="disabled")
    self.input_entry.config(state="disabled")
```

## Classe Room

Gestion des salles dans le manoir, avec leurs caractéristiques et interactions.

Attributs:

**Personnages:**

```
# Characters

Bibliothèque.characters = {

    "Roberta": Character(

        "Roberta la Réceptioniste",

        "la réceptioniste fantomatique du manoir",

        Bibliothèque,

        ["Bienvenue, aventurier ! Vous êtes piégé dans le
manoir. Trouvez un moyen de percer ses secrets pour vous échapper."]

    )

}
```

**Exits:**

```
# Création des sorties des pièces

Hall_d_Entrée.exits = {"SW": Bibliothèque, "SE":
Salle_a_Manger}

Bibliothèque.exits = {"SW": Cave, "SE": Chambre_Hantée, "NW":
Grenier, "NE": Hall_d_Entrée, "E": Salle_a_Manger, "S":
Salle_des_Miroirs}

Salle_a_Manger.exits = {"NW": Hall_d_Entrée, "S":
Chambre_Hantée, "W": Bibliothèque}
```

Méthode:

**get\_long\_description:** Fournit une description complète de la salle, incluant les sorties et objets.

```
def get_long_description(self):  
    """  
    Retourne une description complète de la pièce, y compris ses  
    sorties.  
    """  
    return f"\nVous êtes  
{self.description}\n\n{self.get_exit_string()}\n"
```

**get\_inventory:** Retourne une liste formatée des objets disponibles dans la salle.

```
def get_inventory(self):  
    if self.inventory:  
        return "\n".join([f"- {item.name}: {item.description}" for  
item in self.inventory])  
    return "Il n'y a aucun objet dans cette pièce."
```

## Classe Player:

Gestion du joueur, de ses déplacements et de son inventaire.

### Attributs:

```
self.name = name

self.debug = debug

self.current_room = None

self.inventory = {}

self.history = []

self.beamer_location = None

self.game = game # Référence au jeu
```

### Méthodes:

**Move:** Permet au joueur de se déplacer d'une pièce à une autre.

```
def move(self, direction):

    if not self.current_room or direction not in
self.current_room.exits:

        self.game.display_message(f"Impossible d'aller dans cette
direction : {direction}")

        return False
```

**get\_inventory:** Retourne une liste formatée des objets dans l'inventaire du joueur.

```
def get_inventory(self):

    if not self.inventory:

        self.game.display_message("Votre inventaire est vide.")

        return
```

```

        inventory_list = "\n".join([f"- {item.name}:  

{item.description}" for item in self.inventory.values()])

        self.game.display_message(f"Inventaire :\n{inventory_list}")

```

## Classe Character

Gestion des personnages non-joueurs (PNJ), leurs dialogues et mouvements.

### Attributs:

```

self.name = name

self.description = description

self.current_room = current_room

self.msgs = msgs if msgs else []

```

### Méthodes:

**get\_msg:** Retourne un message ou dialogue aléatoire du personnage.

```

def get_msg(self):
    """
    Retourne un message aléatoire du personnage ou un message par
    défaut s'il n'a rien à dire.
    """
    if self.msgs:
        return "\n".join(self.msgs)

    return f"{self.name} n'a rien à dire pour le moment."

```

**Move:** Déplace le personnage dans une pièce adjacente, avec une probabilité aléatoire.

```
def move(self):  
    if random.choice([True, False]): # 50% de chance de se  
déplacer  
        possible_exits = list(self.current_room.exits.values())  
        next_room = random.choice(possible_exits)  
        if next_room:  
            self.current_room.characters.pop(self.name)  
            self.current_room = next_room  
            self.current_room.characters[self.name] = self  
            return True  
        return False
```

### Classe Item:

Gestion des objets trouvés dans le manoir et leurs propriétés.

#### Attributs:

```
self.name = name  
  
self.description = description  
  
self.weight = weight  
  
self.content = "" # Utilisé pour stocker du texte écrit, si  
applicable
```

## Méthodes:

`__str__`: Retourne une représentation textuelle de l'objet.

```
def __str__(self):  
    """  
    Retourne une représentation textuelle de l'objet.  
    """  
  
    if self.name.lower() == "cahier_de_notes":  
        # Inclut le contenu du cahier s'il contient du texte  
        contenu = f"\nContenu : \n{self.content.strip()}" if  
self.content else "\nContenu : (vide)"  
  
        return f"{self.name} : {self.description} (Poids :  
{self.weight} kg) {contenu}"  
  
        return f"{self.name} : {self.description} (Poids :  
{self.weight} kg) "
```

## Classe Command:

Gestion des commandes disponibles dans le jeu et leur structure.

## Attributs:

```
self.command_word = command_word  
  
self.help_string = help_string  
  
self.action = action  
  
self.number_of_parameters = number_of_parameters
```

### **Méthodes:**

**\_\_str\_\_**: Retourne une description formatée de la commande

```
# The string representation of the command.  
  
def __str__(self):  
    return self.command_word \  
           + self.help_string
```



## Perspective des développeurs

Dans le cadre de la conception du jeu, on cherche d'abord à enrichir l'expérience du joueur. Tout d'abord, on a pensé à ajouter des salles, chacune offrant son propre ensemble de défis et d'énigmes.

Pour ajouter une dimension de danger et de suspense, des pièges auraient pu être dispersés dans certaines salles, obligeant les joueurs à rester vigilants et à faire preuve de prudence et de réflexion dans leur exploration. De plus, une mécanique intéressante consistera à récompenser les joueurs qui résolvent correctement une énigme en leur permettant d'ajouter un objet à leur inventaire. Ces objets ne sont pas seulement des trophées, mais ils ont également une utilité significative dans le déroulement du jeu. Cependant, il y a un défi supplémentaire : si un joueur ne parvient pas à résoudre une énigme après deux tentatives, un indice sera fourni pour l'aider dans sa recherche. Cela maintient le niveau de difficulté tout en offrant une assistance pour éviter toute frustration excessive.

Chaque objet collecté correspondra à une pièce spécifique du puzzle global du jeu. L'objectif ultime est de rassembler toutes les pièces du puzzle, généralement associées aux réponses correctes des énigmes, pour déverrouiller la solution à l'énigme finale. Cela encourage les joueurs à explorer activement chaque recoin du jeu, stimulant ainsi leur curiosité et leur engagement.

Enfin, une règle intrigante est mise en place pour créer une connexion entre les différentes salles du jeu. Il est spécifié que l'entrée de la chambre hantée ne sera autorisée que si le joueur a exploré toutes les autres pièces. Cette restriction incite les joueurs à être complets dans leur exploration, ajoutant ainsi une dimension tactique à leur progression.

En combinant ces éléments, notre jeu promet une expérience immersive, stimulante et cohérente, encourageant les joueurs à résoudre des énigmes, éviter des pièges et assembler les pièces du puzzle pour débloquer l'ultime mystère de la chambre hantée.

Une des améliorations envisagées serait d'introduire des personnages non-joueurs (PNJ) évolutifs, capables d'interagir de manière dynamique avec le joueur. Ces PNJ pourraient ajuster leurs dialogues en fonction des progrès du joueur dans le manoir. Par exemple, au début du jeu, un PNJ pourrait fournir des conseils généraux et une introduction à l'histoire. À mesure que le joueur explore le manoir et découvre des indices, ces dialogues pourraient évoluer pour offrir des informations plus spécifiques, des indices supplémentaires ou des avertissements sur les dangers à venir. Cette

mécanique ajouterait de la profondeur à l'histoire et renforcerait l'impression que les PNJ suivent et participent activement au voyage du joueur.

Pour enrichir davantage le gameplay, l'introduction d'énigmes complexes nécessitant plusieurs étapes serait une amélioration majeure. Au lieu de résoudre une énigme en un seul essai, les joueurs pourraient être confrontés à des défis demandant la collecte d'indices dispersés dans plusieurs salles ou l'utilisation d'objets spécifiques dans un ordre précis. Par exemple, une énigme pourrait demander de combiner les informations d'un tableau mystérieux, d'un livre ancien et d'un objet trouvé dans une autre salle pour déverrouiller une pièce cachée. Ce type d'énigmes ajouterait un niveau de réflexion stratégique et inciterait les joueurs à revisiter des zones déjà explorées, rendant l'expérience encore plus immersive.

Un système de progression basé sur des niveaux de difficulté croissants apporterait une structure claire au jeu et augmenterait l'engagement des joueurs. Chaque zone du manoir pourrait être associée à un niveau de difficulté, les énigmes devenant plus complexes à mesure que le joueur avance. Les premières salles offriraient des énigmes simples pour familiariser le joueur avec les mécaniques du jeu, tandis que les salles suivantes nécessiteraient une réflexion approfondie et l'utilisation combinée de plusieurs objets de l'inventaire. Ce système de progression permettrait aux joueurs de ressentir un sentiment d'accomplissement tout en maintenant un défi constant et stimulant.

*Dans le cas de l'élaboration de certaines de nos fonctions et du choix de certains contexte de l'histoire nous avons choisis d'utiliser Chat gpt.*