



# Rapport de projet.

# Sommaire.

**01 Introduction.**

**02 Univers du jeu**

**03 Objectifs du jeu**

**04 Guide utilisateur**

**05 Guide développeur**

**06 Compétences & perspectives**

# 01 Introduction.

Ce rapport présente le développement d'un jeu d'aventure textuel où le joueur incarne un loup dans un univers de fantasy. Le jeu se déroule à travers une série de pièces interconnectées, chaque pièce offrant des objets à récupérer, des personnages avec qui interagir et des énigmes à résoudre. L'objectif principal est de collecter des objets spécifiques et de les rapporter à la pièce centrale, la Great\_cave, afin que le personnage incarné par le joueur obtienne la confiance de la meute.

Le projet met l'accent sur l'utilisation de la programmation orientée objet (POO), une gestion efficace des objets et de l'état du jeu, et une interactivité fluide entre le joueur et l'environnement.



# 02 Univers du jeu

Le jeu plonge le joueur dans un monde où il doit naviguer à travers divers types de pièces : des forêts, des montagnes, des rivières et des cavernes. Chaque pièce contient des éléments interactifs, comme des objets que le joueur peut prendre ou utiliser, ainsi que des personnages non-joueurs (PNJ) avec qui il peut dialoguer. Le monde est également peuplé de dangers et de défis, ce qui rend chaque déplacement stratégique.

- Les Pièces : Les pièces sont des environnements avec des descriptions textuelles qui changent en fonction de la direction empruntée par le joueur. Chaque pièce possède des sorties, qui permettent au joueur de se déplacer vers d'autres pièces, et des objets qui peuvent être pris ou utilisés.
- Les Objets : Les objets sont disséminés dans les différentes pièces et sont essentiels pour progresser dans le jeu. Certains objets sont nécessaires pour franchir des obstacles, comme un badge pour traverser une rivière.
- Les Personnages Non-Joueurs (PNJ) : Les personnages interactifs offrent des indices, des quêtes ou des objets. Le joueur peut parler à ces PNJ pour obtenir des informations ou résoudre des énigmes.

L'objectif du jeu : Le joueur doit explorer ces pièces, interagir avec leur contenu, collecter des objets clés, puis les rapporter dans la Great\_cave pour terminer le jeu.

# 03 Objectif du jeu

L'objectif du jeu est simple :

collecter tous les objets nécessaires et les ramener à la "Great\_cave" pour gagner. Ces objets sont répartis dans différentes pièces et nécessitent souvent des interactions spécifiques pour être obtenus. Certaines pièces comportent des dangers, comme des rivières ou des ennemis, qui imposent au joueur de prendre des décisions stratégiques.

Le jeu comporte plusieurs conditions de défaite :

- Le joueur peut perdre s'il tente de traverser une rivière sans l'objet nécessaire.
- Si le joueur se retrouve dans une pièce avec des ennemis ou des pièges, il peut également être capturé ou tué.

# 04 Guide utilisateur

Le jeu repose entièrement sur des commandes textuelles, où le joueur doit entrer des instructions pour interagir avec l'environnement. Voici les principales commandes que le joueur peut utiliser :

- `go <direction>` : Permet au joueur de se déplacer dans la direction spécifiée. Exemples : `go north`, `go east`.
- `quit` : Quitte le jeu immédiatement.
- `help` : Affiche une liste de toutes les commandes disponibles.
- `history` : Affiche l'historique des pièces visitées.
- `back` : Retourne à la dernière pièce visitée.
- `take <item>` : Permet au joueur de prendre un objet dans la pièce actuelle.
- `drop <item>` : Dépose un objet dans la pièce actuelle.
- `look` : Affiche la description détaillée de la pièce actuelle.
- `check` : Affiche le contenu de l'inventaire du joueur.
- `talk <character>` : Permet de parler à un personnage dans la pièce.

# 05 Guide Développeur

## 5.1. Architecture et Techniques Utilisées

Le jeu est construit en utilisant la programmation orientée objet (POO), ce qui permet une gestion claire et extensible des différentes entités du jeu :

le joueur, les pièces, les objets, les personnages, et les commandes. Chaque aspect du jeu est modélisé par une classe dédiée. Voici un aperçu technique des classes principales :

- **Game** : La classe centrale qui gère le flux du jeu. Elle crée les pièces, configure les commandes disponibles et gère l'interaction avec le joueur. Elle est responsable de la boucle principale qui attend les entrées du joueur et exécute les actions correspondantes.
- **Room** : Chaque pièce est une instance de cette classe. Elle contient une description, une liste d'objets et de personnages, ainsi qu'un dictionnaire des sorties vers d'autres pièces. Elle gère aussi l'ajout et la suppression d'objets et de personnages.
- **Player** : Cette classe représente le joueur et contient son inventaire, son poids maximum (la quantité d'objets qu'il peut transporter), et sa position actuelle. Elle gère aussi les interactions liées à l'inventaire et aux déplacements dans les pièces.

# 05 Guide Développeur

## 5.1. Architecture et Techniques Utilisées

- Item : La classe qui représente les objets que le joueur peut prendre. Elle contient un nom, une description, et un poids. L'objet peut être utilisé ou déposé par le joueur selon les besoins du jeu.
- Inventory : L'inventaire du joueur est une instance de cette classe, qui gère la liste des objets du joueur. Elle vérifie également si un objet peut être ajouté ou retiré de l'inventaire en fonction du poids total.

### Améliorations possibles pour l'architecture :

- Système de sauvegarde et de chargement de la partie : Implémenter une fonctionnalité permettant de sauvegarder l'état du jeu et de le recharger. Cela permettrait au joueur de reprendre sa partie à tout moment.
- Gestion des interactions complexes entre objets : Les objets pourraient interagir entre eux, comme des combinaisons d'objets ou des objets à utiliser dans des contextes spécifiques. Par exemple, une clé pourrait ouvrir une porte ou un objet magique pourrait être utilisé pour accomplir une tâche spéciale.



## 06 Compétences & perspectives

Bien que le jeu soit déjà fonctionnel, plusieurs améliorations sont envisageables :

- Ajout de Quêtes et de Missions : Introduire un système de quêtes où le joueur doit accomplir des tâches avant de pouvoir progresser.
- Interface Graphique : Le jeu pourrait être enrichi d'une interface graphique utilisant Tkinter ou Pygame pour rendre l'expérience visuellement plus engageante et agréable.
- Nouveaux Types d'Objets et d'Interactions : Ajouter des objets interactifs qui pourraient être combinés entre eux ou utilisés dans des situations spécifiques, par exemple, des potions ou des outils.
- Système de Combat et de Compétences : Ajouter un système de combat contre des ennemis ou des animaux sauvages, avec des compétences à débloquent au fur et à mesure de l'aventure.