

# **Tropical Mystery**

Rapport TBA

Amine Muskud E3S

Florian Kenzoua E3FD

## **Le guide utilisateur :**

Le jeu est une escape game à monde ouvert prenant place sur une île paradisiaque et où vous devez découvrir un code caché. Pour découvrir le code, vous devez vous armer de courage et d'ingéniosité pour explorer la carte et interagir avec les différents habitants de l'île. Chaque habitant possède une bague avec un l'un des numéros du code secret gravé dessus. Si vous parvenez à découvrir le bon code alors vous remportez la partie. Au contraire si vous ne parvenez pas à découvrir le code, vous êtes condamné à rester bloqué sur l'île indéfiniment, vous perdez la partie. Mais attention, soyez sur de vous, vous n'avez qu'une seule tentative pour déverrouiller le coffre !

Pour se déplacer, interagir avec son environnement, vous pouvez utiliser de multiples commandes qui rendent le jeu interactif et dynamique. Voici la liste des commandes qui sont à votre disposition :

- help : cette commande permet de lister les commandes qui vous sont disponibles au cours de la partie, c'est très utile si vous cherchez en quelque sort votre chemin...
- go : dans un jeu d'exploration, c'est toujours important de pouvoir se déplacer, la commande go est faite pour vous !
- back : si vous avez le sentiment d'avoir oublié quelque chose sur votre chemin, pas de panique ! Vous pourrez revenir au lieu précédent avec cette commande.
- look : "Just take a look at me now nah nah nah..." (Phil Collins), la commande look est parfait pour regarder autour de vous attentivement et mieux comprendre votre environnement.
- history : pas obligé de se souvenir des lieux que vous visitez, on le fait pour vous et cette commande vous retournera votre historique de visite : c'est pas beau la vie ?
- take : un objet vous fais de l'œil ? Et hop dans la poche.
- drop : un micro un peu trop encombrant dans votre inventaire ? Drop the mic.
- check : qui a dit qu'on ne pouvait pas faire de check après 20 ans ? Avec la commande check vous pourrez facilement checker votre incroyable collection d'objets dans votre inventaire. Attention tout de même, à ne pas dépasser la limite de poids autorisée !
- talk : cette fonction fais parler les bavards, c'est elle qui vous aidera à animer les précieuses discussions avec les habitants de l'île.
- quit : c'est commande vous permet de quitter le jeu à tout moment, attention votre progression sera effacée : vous voulez nous quitter ? Pas de souci, mais il faut en payer le prix !

Après cette brève explication des règles du jeu, vous êtes prêt à partir explorer l'île. Bonne aventure à vous et bonne chance !

## **Le guide développeur (diagramme des classes) :**

Voici le guide de développeur du Tropical Mystery. Il liste les différentes classes du jeu sous la forme d'un diagramme des classes sous la forme :

Classe :

Attribut(s)

Méthode(s)

### **La classe Game :**

Attributs :

- finished : bool
- rooms : List[Room]
- commands : Dict[str, Command]
- player : Player

Méthodes :

- \_\_init\_\_()
- setup()
- play()
- process\_command(command\_string : str) -> None
- print\_welcome()

### **La classe Player :**

Attributs :

- name : str
- current\_room : Room
- inventory : Dict[str, Item]
- max\_weight : int

- history : List[Room]

#### Méthodes :

- \_\_init\_\_(self, name: str)
- get\_history(self, history: List[Room]) -> None
- move(self, direction: str) -> bool
- get\_inventory(self) -> None

#### **La classe Room :**

##### Attributs :

- name : str
- description : str
- exits : Dict[str, Room]
- inventory : Set[Item]
- characters : Dict[str, Character]

##### Méthodes :

- \_\_init\_\_(self, name: str, description: str)
- \_\_str\_\_(self) -> str
- get\_exit(self, direction: str) -> Room
- get\_exit\_string(self) -> str
- get\_long\_description(self) -> str
- get\_inventory(self) -> None

#### **La classe Item :**

##### Attributs :

- name : str
- description : str
- weight : float

##### Méthodes :

- \_\_init\_\_(self, name: str, description: str, weight: float)
- \_\_str\_\_(self) -> str

#### **La classe Command :**

#### Attributs :

- command\_word : str
- help\_string : str
- action : function
- number\_of\_parameters : int

#### Méthodes :

- \_\_init\_\_(self, command\_word: str, help\_string: str, action: function, number\_of\_parameters: int)
- \_\_str\_\_(self) -> str

#### **La classe Actions :**

##### Attributs :

- MSG0 : str
- MSG1 : str

##### Méthodes :

- go(game, list\_of\_words, number\_of\_parameters) -> bool
- quit(game, list\_of\_words, number\_of\_parameters) -> bool
- help(game, list\_of\_words, number\_of\_parameters) -> bool
- back(game, list\_of\_words, number\_of\_parameters) -> bool
- look(game, list\_of\_words, number\_of\_parameters) -> bool
- take(game, list\_of\_words, number\_of\_parameters) -> bool
- check(game, list\_of\_words, number\_of\_parameters) -> bool
- drop(game, list\_of\_words, number\_of\_parameters) -> bool
- talk(game, list\_of\_words, number\_of\_parameters) -> bool
- unlock(game, list\_of\_words, number\_of\_parameters) -> str

#### **La classe Character :**

##### Attributs :

- name : str
- description : str
- current\_room : Room
- msgs : List[str]

### Méthode :

- `__init__(self, name,description,current_room, msgs)`
- `__str__(self) -> str`
- `get_msg(self, character_name) -> str`

### **Les perspectives de développement :**

Dans la suite, la première chose que nous voudrions ajouter à notre jeu est la mise à disposition d'une interface graphique qui permettrait de dialoguer avec les habitants PNJ de l'île.

Aussi nous voudrions donner plus d'importance aux interactions entre le joueur et les PNJ avec des choix multiples notamment pour les dialogues. Cela viendrait terminer de compléter notre jeu en plus d'offrir des conséquences directes sur le jeu en fonction des choix entrepris par le joueur.

Enfin une autre perspective que nous voudrions ajouter par la suite est la gestion de paramètres de santé du joueur comme la faim, la déshydratation etc... Cela permettrait de complexifier l'expérience de jeu et apporter de nouvelles quêtes secondaires autour de la satisfaction des besoins du joueur.