PROJET TBA

Le CLUEDO

groupe Lauriane Riad

PROJET TBA

Guide de l’utilisateur
L'Univers (Plateau et Carte de Réponse) :

Le jeu est une version revisitée du Cluedo, adapté pour correspondre aux règles et à l'univers du jeu TBA. L'action se déroule dans une maison, dont les différentes pièces sont les suivantes :

Le déplacement sur le plateau s'effectue de manière classique : on peut se déplacer vers le nord, sud, est, ou ouest. De plus, des passages secrets sont présents dans chaque coin du plateau, permettant de voyager instantanément au coin opposé de la maison. Les joueurs peuvent également choisir de rester dans la même pièce.

Un meurtre a eu lieu : le Docteur Lenoir a été tué. Le but du jeu est de résoudre cet assassinat en déterminant l'arme du crime, le meurtrier et la pièce où le crime a été commis. Toutes les possibilités sont listées sur la fiche de réponses, qui sert de référence tout au long du jeu.

Pièce	Vrai ou faux
cuisine	
hall	
salon	
bureau	
grand_salon	
bibliothèque	
salle_à_manger	
salle_de_billard	
véranda	

Pièce	Vrai ou faux
Professeur_Violet	
Colonel_Moutarde	
Mademoiselle_Rose	
Madame_Leblanc	
Docteur_Olive	
Madame_Pervenche	

Arme	Vrai ou faux
Poignard	
Revolver	
Chandelier	
Poison	
Corde	
Clé_anglaise	

groupe Lauriane Riad

PROJET TBA

Guide de l’utilisateur

Le But du Jeu :

L'objectif est de formuler des hypothèses et d'affiner les réponses jusqu'à déduire correctement les éléments du crime. Lorsque vous formulez une hypothèse, vous choisissez un arme, un personnage, et la pièce où vous vous trouvez au moment de la proposition.

Si votre hypothèse est incorrecte, une seule des propositions erronées sera révélée aléatoirement au joueur. Il est crucial de noter les mauvaises réponses sur la fiche de jeu, car elles aideront à éliminer des possibilités et à orienter vos futures hypothèses.

Comment Jouer : Trouver l’Hypothèse Finale

Lorsque vous lancez une partie, une courte introduction vous présente le jeu et précise la pièce dans laquelle vous commencez. Elle affiche également les pièces accessibles pour vos déplacements en utilisant la commande *go*. De plus, un aperçu du type d'hypothèse que vous pouvez formuler est fourni par Mr Taupe a chaque tour.

À chaque action effectuée, comme formuler une hypothèse, ou vous déplacer, l'historique des pièces visitées est automatiquement mis à jour. Les déplacements possibles depuis votre position actuelle sont également affichés, vous permettant de planifier vos prochaines actions.

Pour explorer davantage, vous pouvez utiliser la commande look, qui vous permet d'examiner la pièce et d'y trouver des objets. Ces objets peuvent être ramassés avec la commande *take nom_de_l'objet*, et ils seront ajoutés à votre inventaire. Vous pouvez consulter votre inventaire à tout moment avec la commande *check*. Si vous souhaitez déposer un objet dans une autre pièce (ou dans la même pièce après un déplacement), utilisez la commande *drop nom_de_l'objet*.

Pour rendre le jeu plus fluide, un maître du jeu est prévu Mr Taupe. Celui-ci garde en mémoire toutes les commandes incorrectes saisies dans le terminal. Cette fonctionnalité sera accessible via la commande *talk* (actuellement non fonctionnelle).

groupe Lauriane Riad

PROJET TBA

Guide de l’utilisateur
Comment Jouer : Commandes Disponibles

Les commandes disponibles sont les suivantes :

go : Permet de se déplacer sur la carte. Les directions possibles sont : n (nord), s (sud), e (est), o (ouest), ou passage secret.

hypothese : Sert à proposer une hypothèse, une étape cruciale pour progresser dans le jeu.

history : Affiche toutes les pièces que le joueur a déjà visitées, classées dans l'ordre chronologique.

rester : Permet de rester dans la pièce actuelle sans se déplacer.

quit : Quitte le jeu à tout moment.

help : Affiche la liste des commandes disponibles pour aider le joueur.

look : Permet de regarder les objets présents dans la pièce (Optionnel et non essentiel pour le mode de jeu.)

take : Sert à ramasser un objet dans la pièce où l’on se trouve (Optionnel et non essentiel pour le mode de jeu.)

check : permet de voir ce qu’il y a dans l’inventaire du joueur.

drop : Permet de reposer un objet dans la pièce. (Optionnel et non essentiel pour le mode de jeu.)

talk: permet d’avoir le renvoie de l’historique des requêtes fausse gardée par Mr Taupe

groupe Lauriane Riad

PROJET TBA

Guide Du Développeur (le diagramme de classes)

CLASS PLAYER

name: str (Le nom du joueur)
inventory: list[Item] (L'inventaire du joueur, contenant les objets qu'il possède)
current_room: Room (La pièce actuelle dans laquelle se trouve le joueur)
history: list[Room] (Historique des pièces visitées par le joueur)

__init__(self, name): Initialise un joueur avec un nom, un inventaire vide, une pièce actuelle None, un historique vide et des cartes vides.
stay(self): Affiche la description de la pièce actuelle.
get_history(self): Ajoute la pièce actuelle à l'historique si ce n'est pas la dernière visitée.
check_inventory(self): Affiche l'inventaire du joueur.
look(self): Affiche la description de la pièce actuelle et les objets présents dans celle-ci.
take(self, item_name): Permet au joueur de prendre un objet de la pièce actuelle et de l'ajouter à son inventaire.
drop(self, item_name): Permet au joueur de déposer un objet de son inventaire dans la pièce actuelle.
take_item(self, item): Ajoute un objet à l'inventaire du joueur.
drop_item(self, item): Supprime un objet de l'inventaire du joueur.
__str__(self): Retourne une représentation en chaîne de caractères du joueur, incluant son nom et les objets dans son inventaire.
move(self, direction): Permet au joueur de se déplacer dans une direction donnée (nord, sud, est, ouest, passage secret) dans la salle actuelle.

CLASS COMMAND

command_word: str (Le mot-clé de la commande)
help_string: str (La chaîne d'aide décrivant la commande)
action: function (La fonction qui sera exécutée lorsque la commande est appelée)
number_of_parameters: int (Le nombre de paramètres attendus par la commande)

__init__(self, command_word, help_string, action, number_of_parameters): Le constructeur de la classe Command. Initialise les attributs avec les valeurs fournies et effectue les validations nécessaires.
__str__(self): Retourne une représentation sous forme de chaîne de la commande. La chaîne inclut le mot-clé de la commande et sa description.

CLASS GAMEMASTER

finished: bool (Indique si le jeu est terminé)
rooms: list[Room] (Liste des pièces du jeu)
commands: dict[str, Command] (Liste des commandes disponibles)
player: Player (L'objet joueur)
secret_solution: dict (Solution secrète du meurtre)
game_master: GameMaster (Maître du jeu)

__init__(): Initialise l'objet du jeu, configure les attributs et prépare le jeu.
setup(): Configure les éléments du jeu, comme les commandes, les pièces, les objets, le joueur, et le maître du jeu.
play(): Lance la boucle de jeu, appelle la méthode setup() et gère les actions du joueur.
make_hypothesis(): Valide l'hypothèse formulée par le joueur concernant l'arme, le personnage et la pièce.
process_command(): Gère le traitement des commandes saisies par le joueur.
print_welcome(): Affiche un message de bienvenue et décrit la pièce actuelle.
show_turn_summary(): Affiche un résumé du tour du joueur, incluant la pièce actuelle et l'historique des pièces visitées.

groupe Lauriane Riad

PROJET TBA

Guide Du Développeur (le diagramme de classes)

CLASS GAME

finished: bool (Indique si le jeu est terminé)
rooms: list[Room] (Liste des pièces du jeu)
commands: dict[str, Command] (Dictionnaire des commandes disponibles)
player: Player (L'objet joueur)

__init__(): Initialise l'objet du jeu, configure les attributs et prépare le jeu.
setup(): Configure les éléments du jeu, comme les commandes, les pièces, les objets, le joueur, et les sorties des pièces.
play(): Lance la boucle de jeu, appelle la méthode setup() et gère les actions du joueur.
process_command(): Gère le traitement des commandes saisies par le joueur.
print_welcome(): Affiche un message de bienvenue et la description de la pièce actuelle.

CLASS ACTIONS

Aucun attribut spécifique dans cette classe, car elle est composée uniquement de méthodes statiques.

history(game, list_of_words, number_of_parameters): Affiche l'historique des pièces visitées par le joueur.
rester(game, list_of_words, number_of_parameters): Permet au joueur de rester dans la même pièce pour effectuer une autre hypothèse.
go(game, list_of_words, number_of_parameters): Déplace le joueur dans la direction spécifiée.
look(game, list_of_words, number_of_parameters): Affiche les informations sur la pièce actuelle et les objets présents.
take(game, list_of_words, number_of_parameters): Permet au joueur de prendre un objet dans la pièce actuelle.
drop(game, list_of_words, number_of_parameters): Permet au joueur de déposer un objet de son inventaire dans la pièce actuelle.
check(game, list_of_words, number_of_parameters): Affiche l'inventaire du joueur.
quit(game, list_of_words, number_of_parameters): Quitte le jeu.
help(game, list_of_words, number_of_parameters): Affiche la liste des commandes disponibles.

CLASS ROOM

name: str (Nom de la pièce)
description: str (Description de la pièce)
items: list[Item] (Liste des objets dans la pièce)
exits: dict (Dictionnaire des sorties, clé : direction, valeur : pièce correspondante)

__init__(self, name, description, items=None): Initialise la pièce avec un nom, une description, et une liste d'objets (par défaut vide).
get_exit(self, direction) : Retourne la pièce dans la direction donnée si elle existe, sinon None.
get_exit_string(self): Retourne une chaîne décrivant les sorties possibles de la pièce.
get_long_description(self): Retourne une description complète de la pièce, y compris les sorties et les objets.
add_item(self, item): Ajoute un objet à la pièce.
remove_item(self, item): Supprime un objet de la pièce.
__str__(self): Retourne une chaîne représentant la pièce avec son nom et sa description.
list_items(self): Liste les objets dans la pièce.

CLASS ITEM

name: str (Nom de l'objet)
description: str (Description de l'objet)

__init__(self, name, description): Initialise un objet avec un nom et une description.
__str__(self): Retourne une représentation sous forme de chaîne de l'objet, formatée avec son nom et sa description.

PROJET TBA

Les Améliorations Futures

À l'avenir, nous souhaitons rendre la commande talk fonctionnelle. Elle permettrait d'aider le joueur à retrouver le fil de l'intrigue s'il se perd dans ses réponses, facilitant ainsi la progression jusqu'à la fin du jeu.
Nous envisageons également d'introduire un mode de jeu différent, où les objets doivent être collectés et ajoutés à l'inventaire pour pouvoir formuler une hypothèse. Dans ce mode, un poids maximal d'inventaire serait imposé, empêchant les joueurs de ramasser tous les objets en un seul tour de plateau.
Par ailleurs, il pourrait être nécessaire de trouver les personnages et d'interagir avec eux au moins une fois avant de pouvoir les incriminer. Pour augmenter le dynamisme, les personnages changeraient de pièce de manière aléatoire à chaque tour, rendant leur localisation imprévisible.

Amusez-vous bien, et surtout, menez l'enquête avec soin pour découvrir la vérité !

groupe Lauriane Riad