

Rapport Projet TBA :
Gestion d'une armée pour gagner une guerre



IGI-3008 :
SEBBAR Mohamed Reda / Wandaogo Abibatou

Sommaire :

1. Introduction

2. Guide utilisateur

3. Guide développeur

4. Perspectives de développement

5. Conclusion



Introduction :

Bienvenue dans Win The WAR un jeu de stratégie textuel !

Vous incarnez un commandant d'armée avec pour mission d'éliminer toutes les forces ennemies présentes dans différents lieux stratégiques. Votre rôle est de naviguer intelligemment entre les pièces, déployer vos unités de manière optimale, et de remporter des batailles clés. Chaque décision compte : une mauvaise stratégie peut affaiblir votre armée et vous mener à la défaite. Soyez précis et élaborer un plan pour garantir la victoire !

1. Guide utilisateur

- Conditions de Victoire : Vous remportez la partie lorsque toutes les forces ennemies dans tous les lieux sont éliminées.
- Conditions de Défaite : Vous perdez si toutes vos unités sont anéanties par les ennemis
- Commandes disponibles :

go <direction> : Se déplacer dans une direction cardinale (N, S, E, O).

look : Observer la pièce actuelle pour voir sa description et les forces ennemies présentes.

attack : Attaquer les forces ennemies dans la pièce actuelle.

help : Afficher la liste des commandes disponibles.

quit : Quitter le jeu.

back : Revenir en arrière vers la pièce précédente.

check : Vérifier les items dans votre inventaire.

take <item> : Prendre un item dans la pièce actuelle.

drop <item> : Déposer un item dans la pièce actuelle.

talk <pnj> : Parler à un personnage non joueur dans la pièce actuelle. Voici les commandes que vous pouvez utiliser dans le jeu :

- Les différents lieux du jeu :
 - Forteresse
 - Forêt
 - Désert

- Ruine
- Port
- Plaine
- Gorge
- Citadelle
- Tour
- Cryptes

- Les différentes directions :

N : nord, S : sud, E : est, O : ouest, U : haut, D : bas

- Les items :

Swords : Épée Légendaire : Une épée ancienne qui inflige de lourds dégâts.

Bow : Arc Long : Un arc qui permet de tirer sur des ennemis à distance.

Axe : Hache de Guerre : Une arme destructrice contre les boucliers ennemis.

- Les pnj :

Grimgor : Un forgeron robuste et expérimenté.

Lyria : Une éclaireuse agile prête à partager ses connaissances.

Capitaine : Un soldat imposant qui veille à la sécurité de la forteresse.

- Conseils stratégiques :

Choisissez votre chemin intelligemment : Certaines pièces contiennent des armées plus fortes. Évitez de les affronter trop tôt.

Combattez avec prudence : Vos unités sont limitées. Chaque bataille doit être stratégiquement utile.

2. Guide développeur :

Le projet est organisé en plusieurs fichiers Python, chacun responsable d'une partie du jeu :

- ☐ **Game** : La classe principale qui coordonne le fonctionnement du jeu. Elle relie les commandes, les pièces et le joueur.
- ☐ **Room** : Représente chaque lieu du jeu avec ses connexions (exits), ses objets (inventory), et ses personnages (character).
- ☐ **Player** : Gère l'état du joueur, y compris sa position actuelle, son inventaire et son historique de déplacements.
- ☐ **Character** : Représente les PNJ avec des messages interactifs et la capacité de se déplacer entre les pièces.
- ☐ **Item** : Définit les objets interactifs que le joueur peut ramasser et utiliser.

- ❑ **Command** : Définit la structure des commandes disponibles et leur fonctionnement.
- ❑ **Actions** : Contient les implémentations des différentes actions que le joueur peut effectuer.

Classes	Attributs	Méthodes
Game	rooms: list : Liste des pièces du jeu.	setup() : Initialise les pièces et le joueur
	- player: Player : Instance représentant le joueur.	play() : Lance la boucle principale du jeu.
	- finished: bool : Indique si le jeu est terminé.	process_command() : Traite une commande utilisateur.
	- commands: dict : Dictionnaire des commandes disponibles.	- check_victory() : Vérifie si la condition de victoire est atteinte.
		- check_defeat() : Vérifie si la condition de défaite est atteinte.

Room	- name: str : Nom de la pièce.	__init__() : Initialise une pièce.
	description: str : Description de la pièce.	get_exit() : Retourne une pièce connectée dans une direction donnée.

	exits: dict : Sorties vers d'autres pièces.	get_long_description() : Retourne une description complète de la pièce.
	inventory: dict : Objets présents dans la pièce.	get_inventory() : Liste les objets disponibles dans la pièce.
	character: dict : PNJ présents dans la pièce.	

Player	name: str : Nom du joueur.	__init__() : Initialise le joueur.
	current_room: Room : Pièce actuelle du joueur.	get_inventory() : Retourne une liste des objets dans l'inventaire.
	inventory: dict : Objets possédés par le joueur.	move() : Déplace le joueur dans une direction donnée.
	history: list : Historique des déplacements.	get_history() : Affiche l'historique des déplacements.

Character	name: str : Nom du personnage non joueur (PNJ).	__init__() : Initialise un PNJ.
-----------	---	---------------------------------

	description: str : Description du PNJ.	__str__() : Retourne une description textuelle du PNJ.
	current_room: Room : Pièce actuelle du PNJ.	get_msg() : Retourne un message du PNJ.
	Msgs : list : Liste des messages associés au PNJ.	move() : Déplace le PNJ dans une pièce adjacente.

Item

Item	move() : Déplace le PNJ dans une pièce adjacente.	__init__() : Initialise un objet.
	description: str : Description de l'objet.	__str__() : Retourne une description textuelle de l'objet.
	weight: int : Poids de l'objet.	to_dict() : Convertit l'objet en un dictionnaire pour un stockage ou une manipulation facile.

Command

Command	command_word: str : Mot-clé de la commande.	__init__() : Initialise une commande.
	help_string: str : Description de la commande.	__str__() : Retourne une description textuelle de la commande.

	action: function : Fonction associée à la commande.	
	number_of_parameters: int : Nombre de paramètres requis par la commande.	

Action		go() : Déplace le joueur dans une direction donnée.
		- look() : Affiche la description de la pièce actuelle.
		take() : Permet au joueur de ramasser un objet.
		drop() : Permet au joueur de déposer un objet.
		back() : Permet au joueur de revenir en arrière.
		check() : Affiche l'inventaire du joueur.
		help() : Affiche l'aide du jeu.
		quit() : Quitte le jeu.

3. Perspective de développement :

Il y'a beaucoup d'aspect où on peut améliorer notre jeu.

Interface Graphique

- Remplacer l'interface en ligne de commande par une interface graphique interactive avec des cartes des lieux.

Intelligence Artificielle pour l'Ennemi

- Faire bouger les armées ennemies entre les pièces pour ajouter un niveau de difficulté supplémentaire.

Multijoueur

- Intégrer un mode multijoueur où deux joueurs contrôlent chacun une armée et se battent pour le contrôle des lieux.

Histoire et Narration

- Ajouter une histoire immersive avec des dialogues, des choix moraux et des conséquences sur le jeu.

Conclusion

Ce projet de jeu de stratégie textuel nous a permis de concevoir un système immersif combinant exploration, gestion

d'inventaire et combats stratégiques. Avec une structure modulaire et des fonctionnalités bien définies, le jeu offre une base solide pour des améliorations futures. Par exemple, l'ajout d'une interface graphique, d'un mode multijoueur ou d'une intelligence artificielle plus complexe pourrait grandement enrichir l'expérience. Ces idées ouvrent des perspectives intéressantes pour continuer à développer ce projet et offrir une expérience encore plus engageante.