

Rapport : Jeu "Légendes de Jumanji"

1. Introduction :

Le jeu "Légendes de Jumanji" offre une aventure immersive dans la jungle mystique de Jumanji. Le héros intrépide se retrouve pris au piège dans la jungle mystérieuse de Jumanji, avec pour objectif ultime de découvrir le trésor caché. Cependant, des obstacles périlleux bloquent son chemin, et il doit surmonter des épreuves, résoudre des énigmes, et relever des défis, tout en faisant face à des mystérieux sabotages..

Accès au dépôt GIT : <https://github.com/maymiled/ProjetJumanji2>

2. Univers et Personnages :

- Le joueur incarne un héros choisi parmi différentes classes (un aventurier ou un héros "intelligent"). En fonction de l'héros qu'il choisit, le joueur sera face à des épreuves différentes . Par exemple, l'aventurier devrait plutôt choisir des épreuves de force et de combat, tandis que l'intelligent devrait plutôt faire des choix sages. En général l'aventurier perdra davantage de points de vie en faisant de sages décisions plutôt qu'en décidant de prendre des risques.
- Il rencontre des alliés et défie des ennemis dans la jungle mystique de Jumanji.
- Les joueurs sont amenés à explorer des environnements uniques tels que la Clairière des Lucioles Éthérées, les Cascades d'Émeraude Brillante, et plus encore.

3. Mécaniques de Jeu :

- Au début du jeu, l'aventurier ne connaît pas le chemin exact vers la sortie de la jungle. À chaque obstacle sur son chemin, s'il réussit l'épreuve, il sera amené à faire des choix pour s'échapper.
- En cas d'échec, l'utilisateur perd des points de vie et se retrouve dans une destination aléatoire.

4.Épreuves et Obstacles:

- Chaque obstacle est associé à une épreuve que l'aventurier doit surmonter pour avancer sur le bon chemin dans le jeu.
- Les épreuves peuvent varier en complexité, allant d'énigmes à des défis physiques. Des phénomènes aléatoires peuvent apparaître sur leur chemin faisant varier la vie du joueur.

5.Santé de l'Aventurier :

- L'aventure commence avec une certaine quantité de vie (=100)
- Un échec lors d'une épreuve fait perdre de la vie.
- Si l'aventurier perd toute sa vie, le jeu se termine.

6. Interface Utilisateur :

- Nous avons implémenté une interface textuelle. Celle-ci présente un menu à l'utilisateur lui permettant notamment de reprendre une ancienne sauvegarde au début du jeu, de sauvegarder sa partie ou de la quitter au milieu ou encore de voir sa jauge de vie tout au long du jeu.

7. Système d'Événements et Narration :

- Événements dynamiques basés sur les choix des joueurs et les résultats aléatoires.
- Narration immersive avec des dialogues et des descriptions pour renforcer l'histoire.
- A chaque nœud, sa description s'affiche ainsi que les titres des nœuds voisins pour faire les choix suivants.

8. Système de Progression :

- Évoluer en perdant le moins de points de vie possible pour réussir à s'échapper.
- L'idéal pour l'utilisateur est de trouver le chemin le plus court vers la sortie.
- Tous les choix qu'il a fait précédemment influent d'une manière ou d'une autre ses choix qui suivent.

Structure du projet et hiérarchie des classes :

1. Package 'representation' :

ChanceNode.java : Représente un nœud dans lequel le joueur peut rencontrer des événements aléatoires ou des situations imprévues.

DecisionNode.java : Modélise un nœud dans lequel le joueur doit prendre une décision qui affectera le déroulement du jeu.

Event.java : Classe générique représentant des événements dans le jeu.

FabriqueVideotheque.java : Classe responsable de la création des graphes liés à la vidéothèque du jeu. C'est comme si on pouvait tirer différentes cassettes en fonction de la partie que nous souhaitons jouer.

Game.java : Représente le moteur principal ou le point d'entrée du jeu.

Graph.java : Présentation de la classe qui travaille avec des structures de graphe pour modéliser le déroulement du jeu.

ImageNode.java : Une classe représentant un nœud dans lequel des images peuvent être affichées, peut-être pour des scènes spécifiques.

InnerNode.java : Une classe représentant un nœud interne, classe de base pour d'autres types de nœuds.

Node.java : Classe de base pour tous les types de nœuds dans le jeu.

NodeDecorator.java : Une classe qui peut être utilisée pour ajouter des fonctionnalités supplémentaires aux nœuds existants (pattern décorateur).

SoundNode.java : Représente un nœud dans lequel des fichiers son peuvent être joués.

TerminalNode.java : Représente un nœud final ou terminal dans lequel le jeu peut se conclure.

2. Package 'univers':

L'organisation du package `univers` semble également suivre une structure logique. Voici une brève explication de chaque classe basée sur leur nom :

BanqueEnigme.java : Classe qui stocke et gère un ensemble d'énigmes pour le jeu.

BouleMagique.java : Représente une boule mystérieuse qui peut faire perdre ou gagner de la vie, une épreuve spécifique dans le jeu.

ChoixSimple.java : Classe qui modélise des choix simples que les joueurs peuvent faire dans le jeu sans passer par un défi spécifique

CourseMontre.java: Représente une course contre la montre, une épreuve spécifique dans le jeu.

Enigme.java : Une classe représentant une énigme dans le jeu.

Epreuve.java : Représente une épreuve ou un défi que les joueurs doivent surmonter.

IPersonnage.java :interface représentant un personnage dans le jeu.

ObjetMagique.java : Une classe représentant un objet magique dans le jeu.

Personnage.java : Représente un personnage dans le jeu avec les caractéristiques qui lui sont propres

TypeObjetMagique.java : Énumération ou classe représentant les différents types d'objets magiques.

3. Package 'data':

SaveData.java : Classe qui permet d'effectuer les sauvegardes.

4. Package 'test':

Package spécifique aux tests unitaires

5. Package 'main'

Classe principale du jeu

6. Package 'vue'

FabriqueUI.java : Classe fabrique qui crée des objets UI (interface utilisateur). Cela peut être utile pour instancier différents types d'interfaces utilisateur en fonction des besoins.

UI.java : Cette classe représente une interface utilisateur générique. Elle peut fournir des méthodes et des fonctionnalités de base que toutes les interfaces utilisateur doivent avoir.

UITexte.java : Représente une interface utilisateur basée sur du texte. Elle peut afficher des informations textuelles aux utilisateurs.

UITexteCouleur.java : Une extension de la classe UITexte qui prend en charge la couleur du texte. Cela pourrait permettre d'afficher des informations de manière plus visuelle en utilisant différentes couleurs. Cette interface n'a pas été utilisée au dernier moment

L'organisation suggère une hiérarchie où `UI` est une classe de base et `UITexte` ainsi que `UITexteCouleur` sont des classes dérivées qui ajoutent des fonctionnalités spécifiques. La présence d'une fabrique (`FabriqueUI.java`) suggère une approche de conception de fabrique pour créer des instances d'interfaces utilisateur, ce qui peut être bénéfique pour la flexibilité et l'extensibilité du code.

Challenges et difficultés du Projet "Légendes de Jumanji" :

Nos points forts du Projet :

1. Implémentation d'Épreuves Interactives :

- L'un des points forts du projet réside dans la réussite de l'implémentation d'épreuves interactives telles que des énigmes et l'épreuve de la course à la montre ou la boule magique. Ces mini-jeux ajoutent une dimension stimulante à l'aventure, offrant aux joueurs des défis variés.

2. Banque d'Énigmes Dynamique :

- La création d'une banque d'énigmes permettant aux joueurs de recevoir plusieurs énigmes différentes est pour nous une innovation notable. Cela offre une expérience de jeu riche et renouvelée, stimulant la réflexion des joueurs tout en évitant la répétition.

3. FabriqueVidéotheque pour la Diversité des Histoires :

- La conception de la FabriqueVidéotheque est un point fort majeur. Cette structure permet la création de plusieurs graphes de jeu, offrant ainsi la possibilité de développer une multitude d'histoires avec des options différentes. Cela accroît considérablement la rejouabilité du jeu.

Nos défis majeurs et difficultés rencontrés:

Mécanique de la FabriqueVidéotheque :

- La mise en œuvre réussie de la FabriqueVidéotheque, bien que créative, a nécessité une approche complexe pour garantir la cohérence et la variété dans les histoires générées. Trouver l'équilibre entre la flexibilité et la cohérence a été une tâche délicate.

Mini-Jeux Interactifs :

- La conception et l'implémentation de mini-jeux interactifs comme la course à la montre ont exigé une attention particulière pour maintenir l'équilibre entre le défi et le plaisir de jeu. Mais nous avons réussi à créer des mécaniques de jeu engageantes.

Mode Coloré (UiTexteColore) :

- Malheureusement, l'implémentation d'un mode de jeu en couleur, appelé "UiTexteColore", s'est révélée plus complexe que prévu. Malgré nos efforts, nous n'avons pas réussi à intégrer avec succès la coloration du texte dans le jeu à la dernière minute à cause de l'importation de java.org.