

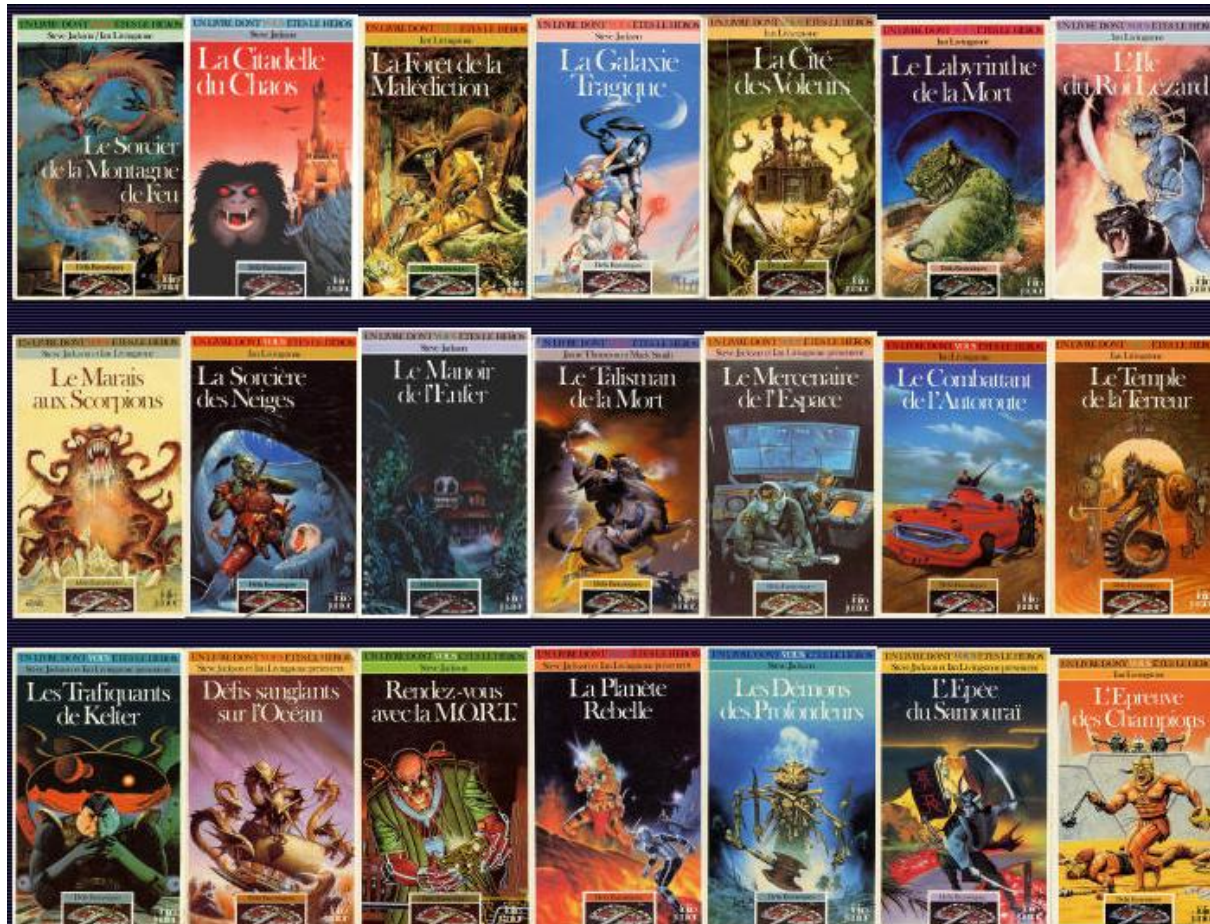
Cahier des Charges Mini projet

Jeu Java "Livre dont vous êtes le héros"

1	INTRODUCTION	2
2	OBJECTIFS	2
3	EXEMPLE ARCHITECTURE DU PROJET	3
3.1	PACKAGE MODEL (MODELE)	3
3.2	PACKAGE CONTROLLER (CONTROLEUR).....	3
3.3	PACKAGE VIEW (VUE).....	3
4	FONCTIONNALITES PRINCIPALES	3
5	ÉVOLUTIVITE	3
6	ORGANISATION DU PROJET.....	4
7	PLANNING (EXEMPLE)	4
8	LIVRABLES	4

1 Introduction

Ce projet vise à développer une application en Java permettant de jouer à un "Livre dont vous êtes le héros". (<https://bibl.remz.ca/>) L'application devra être évolutive et modulaire afin de permettre l'ajout de nouveaux scénarios et fonctionnalités.



2 Objectifs

- Développer une architecture logicielle modulaire et extensible.
- Permettre la création et l'exécution d'un scénario interactif.
- Offrir une gestion fluide des choix et des conséquences des actions du joueur.
- Fournir une interface graphique en Swing (prioritaire) et éventuellement une version en ligne de commande.

3 Exemple Architecture du Projet

Voici un exemple d'architecture de votre projet

3.1 Package model (Modèle)

- Scenario : Contient l'ensemble des chapitres et leurs connexions.
- Chapitre : Représente une section du livre avec un texte et des choix.
- Choix : Associe un texte de choix et la référence vers le chapitre suivant.
- Personnage : (optionnel) Peut contenir des informations sur le joueur (inventaire, statistiques, etc.).

3.2 Package controller (Contrôleur)

- GameController : Gère la progression de l'histoire et l'interaction utilisateur.
- ScenarioLoader : Charge les scénarios depuis un fichier JSON, XML ou base de données.

3.3 Package view (Vue)

- SwingUI : Interface graphique principale en Swing.
- TextUI : Interface en ligne de commande en option pour tester le jeu plus rapidement.

4 Fonctionnalités Principales

- Affichage d'un chapitre et de ses choix.
- Saisie du choix par le joueur et navigation entre les chapitres.
- Chargement dynamique d'un scénario depuis un fichier externe.
- Possibilité de sauvegarder et charger une partie.

5 Évolutivité

- Le projet doit être conçu de manière à pouvoir être complété après la première version.
- Ajout d'un système d'inventaire et de gestion des statistiques du joueur.
- Création d'un éditeur de scénarios.
- Possibilité d'intégrer d'autres bibliothèques graphiques si besoin.

6 Organisation du Projet

- **Nombre de participants** : 2 à 3 personnes.
- **Outils recommandés** : GitHub/GitLab pour la gestion des versions et le travail collaboratif , Trello pour la gestion de projet.
- **Langage** : Java (JDK 17 ou supérieur recommandé).
- **Bibliothèques possibles** : Swing (prioritaire), Son/Jackson (JSON).

7 Planning (Exemple)

Le projet se déroule du **6 mars au 21 mai**.

- **Semaine 10-15 (06/03 - 8/04) : Analyse & Conception**
 - Définition des classes et packages.
 - Élaboration d'un premier scénario de test.
 - Mise en place du dépôt Git et répartition des tâches.
- **Semaine 15-19 (8/04 - 10/05) : Développement de base**
 - Implémentation des classes du modèle (Scenario, Chapitre, etc.).
 - Développement du contrôleur (GameController, ScenarioLoader).
 - Création de l'interface graphique en Swing.
- **Semaine 19-20 (10/05 - 15/05) : Tests et améliorations**
 - Tests unitaires et correction de bugs.
 - Ajustement de l'interface graphique.
 - Ajout d'éventuelles améliorations (ex. gestion de sauvegarde).
- **Semaine 20-21 (15/05 - 20/05) : Finalisation et rendu**
 - Préparation de la version fonctionnelle du jeu.
 - Rédaction de la documentation technique et utilisateur.
 - Présentation du projet et démonstration.

8 Livrables

- Un document détaillant l'analyse et la conception du projet
- Code source documenté.
- Une partie jouable du projet, même incomplète, avec un minimum de fonctionnalités fonctionnelles.
- Documentation utilisateur.