



Des items pour les chevaliers... Chevaleresk ! ¹

Objectifs

Ce travail vise principalement à vous faire expérimenter les aspects suivants:

- Appliquer une méthode agile de gestion de projets pour mener à bien le développement d'un projet.
- Concevoir une base de données.
- Concevoir la logique applicative.
- Concevoir un prototype visuel.
- Effectuer les tests unitaires, d'intégration et d'acceptation.
- Produire un document de conception.
- Travailler en équipe.

Mise en situation

Nous sommes dans un environnement du jeu vidéo de type médiéval, et nous souhaitons développer l'application *Chevaleresk* qui permet l'achat en ligne d'un certain nombre d'items concernant le jeu, la gestion des items pour le jeu et l'évaluation de l'efficacité d'un item.

Dans l'environnement du jeu, il y a plusieurs joueurs. Chaque joueur peut acheter plusieurs items et des joueurs différents peuvent acheter des items identiques. La quantité d'un item acheté peut être supérieure à 1. **Pour les achats, nous utiliserons un panier d'achat.**

¹ Source de l'image : https://fr.freepik.com/photos-gratuite/chevalier-medieval-gris_9279970.htm

Items du jeu

1. Un item du jeu a un numéro unique, un nom, une quantité en stock, un type d'item (armes, armures, potions, des éléments etc..), un prix unitaire en écu, et une photo.
2. Les armes ont une efficacité, un genre (une main, deux mains) une description.
Exemples : Flèches, arc, marteau, épée, bâton de pierre, sabre etc...
3. Les armures on matière qui compose l'armure et la taille : exemple capuche flamboyante, masques crâniens, etc....)
4. Les potions ont un effet attendu et la durée pour l'effet. Exemple : Potion de vitesse, potion de protection, potion de soin, potion pour augmenter la puissance d'attaque etc.
Les potions sont de deux types : attaque et défense.
5. Les éléments servent à concocter des potions. Une potion est réalisée à partir de plusieurs éléments. Un élément peut faire partie de la composition de plusieurs potions. Un élément a en plus le type (plante, épice , sang etc..) rareté , la dangerosité. Les éléments sont moins chers que les potions
6. Pour faire une potion, nous avons besoin de connaitre la quantité de chaque élément qui compose la potion et il faut être un alchimiste.

Les joueurs

1. Un joueur a un alias unique, un nom, un prénom et un solde en **écu**. Les joueurs sont identifiés par un numéro séquentiel et ont un niveau. Certains joueurs sont des « alchimistes »
2. Tous les joueurs peuvent acheter n'importe quel item sauf les éléments.
3. Seuls les alchimistes peuvent acheter des éléments.
4. Lorsqu'un joueur alchimiste possède les composants d'une potion avec les quantités suffisantes, alors il peut concocter la potion.
5. Initialement tous les joueurs alchimistes de niveau débutant. Les autres joueurs sont de niveau « aucun »

Les pièces :

1. Initialement tous les joueurs ont un nombre fixe de pièces (écus). C'est le solde (ou le capital) initial. Ce solde est le même pour tous les joueurs.
2. Les joueurs peuvent augmenter leur capital de la façon suivante :
 - Faire une quête. Une quête consiste à résoudre une énigme. Chaque quête a un nombre d'écus. Ce nombre dépend de la difficulté de l'énigme ou du type de

l'énigme. Exemple, une quête concernant les éléments et les potions rapportent plus de points.

- Demander à l'administrateur : L'administrateur augmente le nombre d'écus avec un montant fixe. L'admin peut augmenter le capital du joueur de 200 écus à chaque demande pour un maximum de 600 écus.

Indications concernant l'application d'achat d'items pour *Chevaleresk*

Lorsque la personne n'est pas un joueur inscrit (pas de nom usager ni de mot de passe), alors on peut consulter les items selon les critères suivants: (ce sont des critères de recherches et non un tri)

1. Tous les items : l'affichage doit être ordonné par type.
2. Par type d'item : l'affichage doit se faire par prix. Ici on souhaite consulter la liste des items selon le type. Par exemple, je veux juste afficher les armes.
3. En combinant plusieurs types : l'affichage doit être ordonné par type, puis par prix. Ici on souhaite consulter la liste des items selon des types combinés. Par exemple, je veux juste afficher les armes et les potions
4. Si aucun type n'est sélectionné, alors on doit pouvoir afficher tous les items.
5. S'inscrire au site.

Si la personne est un joueur inscrit alors le joueur peut :

1. S'identifier avant d'acheter.
2. Faire un achat d'un ou plusieurs items et mettre à jour la BD après un achat.
 - Lors d'un achat, on doit pouvoir ajouter, supprimer ou modifier la quantité d'un item dans le « panier ».
 - Lors d'un achat, on ne doit pas pouvoir acheter un item avec une quantité N, si cette quantité est plus grande que la quantité en stock → équivalent à dire : Je ne peux pas acheter 2 potions s'il en existe juste 1
 - Lors d'un achat, si le montant total de l'achat est plus élevé que le capital du joueur, alors il ne peut pas acheter à moins de modifier son panier ou d'augmenter son capital.
 - Si le joueur n'est pas un alchimiste, il ne peut pas acheter des éléments.
3. Modifier son profil.
4. Consulter la liste de ses items dans son inventaire.
5. Évaluer et commenter un item qu'il a acheté. Chaque joueur poste un commentaire sur un item qu'il a déjà acheté et évalue l'item en donnant une note de 1 à 5 pour : Efficacité.
6. Consulter les évaluations des : Nous souhaitons :

- Afficher le nombre de personnes ayant attribué un nombre d'étoiles (exemple 3 personnes ont donné 4 étoiles, une personne a donné 3 étoiles, 2 personnes ont donné 2 étoiles).
- Afficher la moyenne des évaluations. (Tel item a en moyenne 3 étoiles...) et effectuer une recherche selon le nombre d'étoiles (exemple tous les items avec le nombre d'étoiles 3 ou plus).

Indications concernant la partie administration de l'application *Chevaleresk*

Minimalement, l'application doit permettre :

- L'ajout d'un item (nouvelle arme, nouvelle armure, etc.)
- Augmenter le capital du joueur.

PanoraMix , pour concocter les potions

Lorsque le joueur alchimiste a les éléments en quantités suffisantes pour concocter une potion et qu'il souhaite concocter une potion alors l'inventaire du joueur doit être mis à jour :

- On ajoute la potion à l'inventaire du joueur
- On met à jour la quantité des éléments utilisés pour la potion.
- Si la quantité est nulle pour un élément après avoir concocter une potion alors cet élément est supprimé de l'inventaire du joueur.
- Lorsque le joueur a concocté 3 potions de type attaque ou 5 potions de type défense il passe au niveau « intermédiaire » .
- Lorsque le joueur a concocté 5 potions de types attaque ou 10 potions différentes il passe au niveau « Expert »
- Lorsqu'un joueur non-« alchimiste » devient alchimiste, il passe du niveau « aucun » au niveau « débutant »
- Pour devenir alchimiste, il faut faire des quêtes en rapport avec l'alchimie

Enigma, pour les quêtes

Les quêtes pour augmenter le capital du joueur sont réalisées à l'aide de l'application **Enigma**. L'application doit fournir au joueur des énigmes à résoudre. Lorsque le joueur a réussi à résoudre l'énigme le capital du joueur est augmenté.

Ce ne sont pas toutes les énigmes qui donnent le même montant. Avant de résoudre l'énigme, le joueur doit se connecter à l'application.

Les énigmes difficiles donnent 200 écus les énigmes de difficulté moyenne donnent 100 écus et les énigmes faciles donnent 50 écus. L'énigme à résoudre est tirée de manière aléatoire, mais le joueur pourrait décider de choisir la difficulté de l'énigme.

Lorsque le joueur a répondu à au moins 3 questions en rapports avec les éléments ou les potions alors il devient « alchimiste ».

Dans *Enigma*, nous pouvons également garder un certain nombre de statistiques comme : le nombre d'énigmes résolues, le nombre d'énigme échouées etc...

Recommandation pour la conception

Pour **Chevaleresk**, vous avez le choix de conception entre LAMP (MySQL, PHP) ou ASP.NET (SQL Server et C#, ASP.NET). Dans tous les cas l'application doit être hébergée. Afin de prévenir les injections SQL, **Il est recommandé** d'utiliser des procédures stockées pour votre base de données.

Le développement

Le développement est un développement agile SCRUM, donc le backlog est **VIVANT** et **le client peut apporter des changements en tout temps.**

La description de ce qui est attendu dans l'application peut être complétée par les clients concernés en tout temps. Il est de votre responsabilité de clarifier les éléments du backlog en tout temps