



## Rapport de groupe

---

### GROUPE: CD-1

---

- Nihal BELMAKHFI
- Amine BENCHABOUN
- Augustin BOULARD
- Théo KOTSONIS
- Jos LANDURE
- Édouard LEFIZELIER
- Thomas MORIEUX

# 1 Rappel du sujet

Notre équipe souhaite réaliser un RPG en 2D, avec des combats en tour par tour.

## 2 Principales fonctionnalités

### 2.1 Déplacement

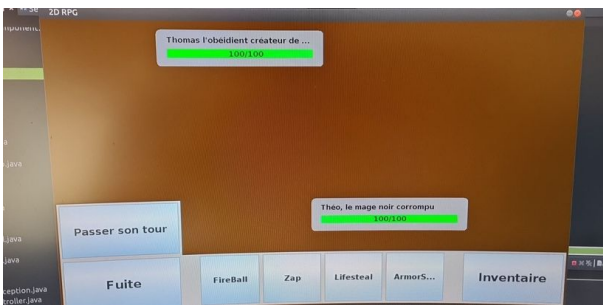
L'une des principales fonctionnalités d'un jeu RPG est de pouvoir se déplacer librement sur carte. Nous avons donc mis en place lors de la deuxième itération un fond sur lequel un personnage peut se déplacer avec des mouvements simples (vers le haut, le bas, la droite et la gauche). De plus, nous avons animés les personnages pour donner l'impression qu'il se déplace en marchant.



De plus, nous avons ajouté des personnages non joueur (PNJ) sur la carte avec lesquelles il est possible d'interagir (parler à un marchand, lancer un combat contre un loup). Enfin, nous avons ajouté un fond musical lorsque le jeu est lancé grâce à différentes classes (AudioClip.java, AudioPlayer.java, MusicClip.java. . .)

### 2.2 Combat

Une autre fonctionnalité importante de notre RPG est le système de combat au tour par tour. Pour ce faire, nous avons créé une première interface pour les différents combats qui auront lieu durant l'aventure :



L'interface de combat se compose de deux zones indiquant la vie restante des combattants ainsi que le nom et d'un ensemble de boutons interactifs permettant au joueur d'indiquer les actions qu'il souhaite effectuer.

On peut voir les différentes options disponibles durant les combats : les 4 capacités différentes, l'inventaire pour utiliser des objets, la fuite pour éviter un combat et la possibilité de passer son tour.

Les points de vie de chaque combattant diminuent en fonction des sorts utilisés et sont affichés à l'aide des barres de vie vertes.

Pour que ces combats puissent avoir lieu, nous avons dû créer d'autres classes:

1. Une classe pour chaque sort (nous n'avons pour l'instant que Fireball.java, Lifesteal.java, Armorshred.java et Zap.java). Chacun d'eux ont des effets différents, certain peuvent soigner, infliger des dégats, augmenter l'armure, etc ... Les effets des compétences sont également influencés par les statistiques du lanceur et de la cible. Enfin, certain sorts possèdent un nombre de tour de rechargement pour limiter leur puissance.
2. Une classe Buff qui sert à mettre en place des améliorations temporaires durant les combats (augmentation de vie, d'armure...) ou des malus (de vie, d'armure, etc ...). L'effet des buffs affectant un combattant s'active au début du tour de ce dernier. Ces effets ne sont pas infinis, ils ne durent que pendant un nombre de tour limité.
3. Une classe Item pour les différents objets utilisable en combat (Potion de soin...).
4. Une classe Enemy qui modélise les ennemis. Cette classe répertorie leur statistique(vie, compétences, nom...), leur compétence et leur IA de combat (une autre classe qui va gérer les compétences que va lancer un ennemi durant un combat).
5. Une classe Statistics.java pour avoir accès aux statistiques du joueur (vie, armure, compétences...).
6. Une classe BattlePanel s'occupe de l'interface graphique d'un combat mais aussi de récupérer les action que va vouloir effectuer un joueur lors du dit combat.

Nous avons aussi créé la classe abstraite EnemyAI qui va gérer le comportement d'un ennemi en combat. Un exemple d'utilisation est la sous-classe EnemyAIdumb qui va renvoyer une compétence aléatoire de l'ennemi qui n'est pas en rechargement ou passer son tour si aucune compétence n'est disponible.

## 2.3 Quêtes

Pour que le joueur puisse avancer dans l'histoire et avoir des indications sur le chemin à suivre, nous avons mis en place un système de quêtes, des objectifs à réaliser afin de gagner divers objets. Il y a trois types de quêtes:

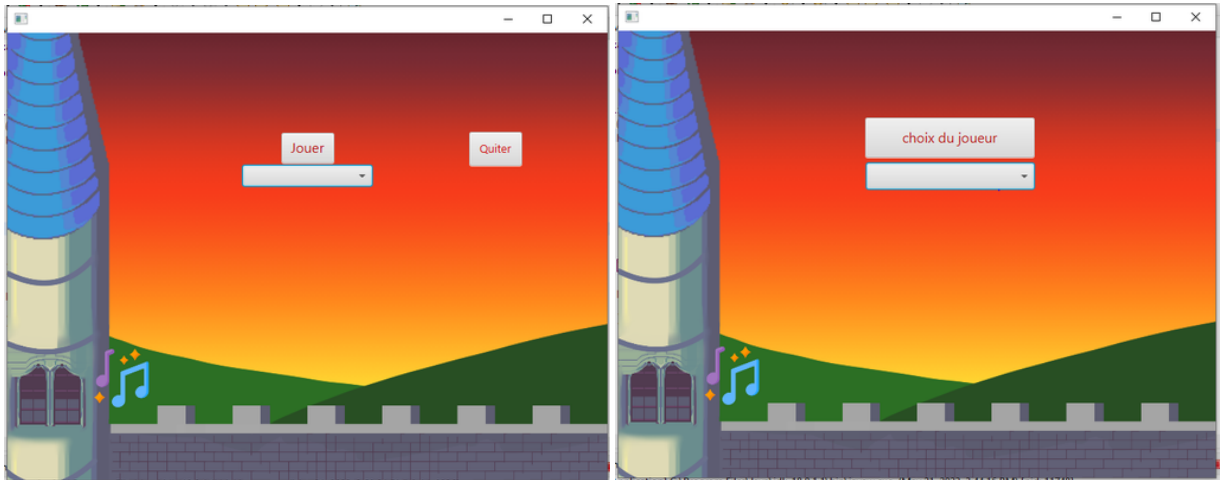
1. Les quêtes de type combat (avec la classe Killing.java) où le joueur doit affronter plusieurs ennemis.
2. Les quêtes de type récolte (avec la classe Gathering.java) durant lesquelles le joueur doit récolter plusieurs matériaux.
3. Les quêtes principales (avec la classe Principale.java) permettant de faire avancer l'histoire du jeu.

## 2.4 Inventaire

Afin d'obtenir, utiliser et garder des objets, le joueur doit avoir un inventaire. Nous avons donc créé une classe Inventory et une classe Item permettant de décrire chaque objet (nom, image, équipement ou consommable...) ainsi que de savoir combien d'objets possède le joueur.

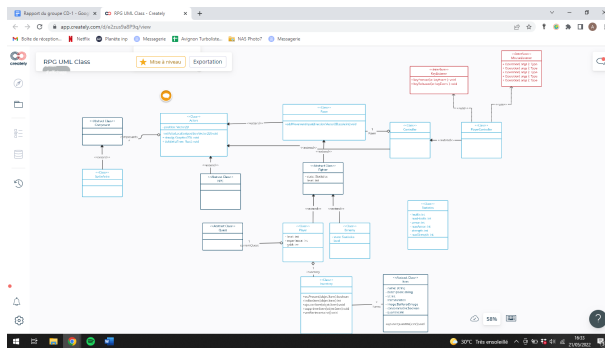
## 2.5 Menu principal

Enfin, nous avons développé un menu principal pour notre jeu, à partir duquel le joueur pourra choisir de débiter une aventure ou de continuer une partie précédente.



### 3 Diagramme de classe

Nous avons complété au fur et à mesure notre diagramme UML en ajoutant les classes principales permettant de comprendre l'architecture de notre application :



Voici un lien pour voir plus clairement le diagramme : <https://app.creately.com/d/e2zus9a8P9q/view>

### 4 Principaux choix de conception

Nous avons décidé de créer en priorité les aspects les plus importants d'un jeu, sans les développer trop en détail afin d'avoir une base solide et de pouvoir montrer au client toutes les fonctionnalités auxquelles il aura accès. Par exemple, nous n'avons créé que quelques compétences sans rentrer trop dans le détail pour avoir une idée générale de leurs effets. De plus, nous avons laissé l'histoire du jeu et ses détails de côté pour nous concentrer principalement sur ce qui rend le jeu jouable.

### 5 Mise en oeuvre des méthodes agiles

#### 5.1 Mise en contexte

Au cours de ce projet, nous avons été encouragés à nous appuyer sur les méthodes agiles pour collaborer au sein de notre équipe. Nous avons donc décidé de mettre en place les différents outils à notre disposition pour nous aider dans le développement de notre jeu RPG.

## 5.2 Utilisateurs

Afin de définir précisément notre application et son utilité, nous avons réalisé un Pitch Elevator :

**POUR:** Les utilisateurs d'ordinateurs

**QUI SOUHAITENT:** Se divertir

**NOTRE PRODUIT EST:** Un jeu RPG (Role Play Game)

**PERMET DE:** Jouer plusieurs personnage dans un monde ouvert, faire des combats, des quêtes etc...

## 5.3 Itérations

**Première itération - Du 28/03 au 09/04 :** Lors de la première itération, nous avons d'abord réfléchi à toutes les fonctionnalités que nous voulions mettre en place (combat, inventaires, quêtes, menu...). Nous avons ensuite commencé l'implémentation des classes principales permettant au jeu de fonctionner ainsi que certaines fonctionnalités (l'inventaire, les quêtes...).

**Deuxième itération - Du 11/04 au 23/04 :** Lors de la deuxième itération, nous avons réussi à déplacer un personnage sur un fond (une forêt), créer une interface de combat (pas encore totalement utilisable) et mettre en place plusieurs classes pour des sorts, des quêtes et des objets. Enfin nous avons développé un menu principal pour notre jeu.

**Troisième itération - Du 09/05 au 21/05 :** Lors de l'itération finale, nous avons réussi à finir l'interface de combat pour la rendre utilisable (pouvoir faire un vrai combat en utilisant des sorts différents avec les points de vie diminuant...). De plus, nous avons ajouté des personnages non joueur sur la carte ainsi que la possibilité d'interagir avec eux (parler avec un marchand, lancer un combat contre un loup). Nous avons aussi créé une interface pour l'inventaire permettant d'afficher les objets que notre personnage possède (potion de soin...). Enfin, nous avons ajouté un fond musical sur la carte.

## 5.4 Répartition du travail

Afin de se répartir le travail et garder une trace de ce que chacun a implémenté, nous avons créé et mis à jour un trello.

<https://trello.com/invite/b/Kti7frjz/89569b0c3577166a18a20072d5dd8033/projet-tob-rpg>

Nous nous sommes aussi régulièrement réunis à l'ENSEEIHNT ou chez l'un des membres du groupe afin de discuter des prochaines étapes du projet, des idées de chacun mais aussi des potentiels problèmes que chacun rencontrait.

## 6 Conclusion

Lors des trois itérations, nous sommes parvenu à remplir une bonne partie des objectifs de ce projet. Nous avons réussi à mettre en place un jeu RPG jouable et qui fonctionne correctement. Cependant, certaines fonctionnalités doivent encore être améliorées, telles que le système de quêtes.