

# GodMel Corporation

## Rapport de soutenance n°1

---

---

### Proxima's Quest

---

---

Marco CHAN WITZEL  
Elias BOUARROUDJ  
Godric BOUTELOUP  
Lucile PELOU

Mercredi 08 mars 2023



**PROXIMA'S**  
QUEST



---

# TABLE DES MATIÈRES

<b>INTRODUCTION</b>	<b>2</b>
<b>1 Vue générale du projet</b>	<b>3</b>
1.1 Rappel des tâches . . . . .	3
1.2 Moyens mis en place . . . . .	3
<b>2 Avancement du projet</b>	<b>5</b>
2.1 Scénario . . . . .	5
2.2 Mécaniques de jeu . . . . .	5
2.2.1 Mouvements et actions du joueurs . . . . .	5
2.2.2 Les différents Items . . . . .	7
2.2.3 Système monétaire et magasins . . . . .	8
2.2.4 Inventaires et fonctionnalités . . . . .	9
2.3 Interfaces et fonctionnalités . . . . .	10
2.3.1 Menus . . . . .	10
2.3.2 Multijoueur . . . . .	11
2.3.3 HUD . . . . .	12
2.3.4 Indications pour le joueur . . . . .	12
2.4 Intelligences artificielles . . . . .	13
2.4.1 Ennemis . . . . .	13
2.4.2 PNJ . . . . .	13
2.4.3 Boss . . . . .	14
2.5 Graphismes . . . . .	15
2.5.1 Maps . . . . .	15
2.6 Animations . . . . .	16
<b>3 Bilan sur l'avancement</b>	<b>17</b>
<b>4 Développement futur</b>	<b>19</b>

## TABLE DES MATIÈRES

---

4.1	Fonctionnalités à implémenter dans le futur . . . . .	19
4.2	Améliorations possibles . . . . .	20
	<b>Conclusion</b>	<b>22</b>



---

# INTRODUCTION

Dans le cadre de notre projet informatique de second semestre, nous avons décidé de réaliser un jeu vidéo : **Proxima's Quest**.

**Proxima's Quest** est un jeu de rôle de type action-RPG. Le joueur incarne un humain qui évolue dans un monde médiéval-fantastique en vue troisième personne. Ce monde est constitué de plusieurs royaumes, chacun gouverné par un dirigeant. La population vit sous le joug de ce gouvernement et des forces maléfiques qui imprègnent leurs mondes.

Nous vous proposons ainsi de découvrir la réalisation du projet durant le premier temps de travail qui nous a été accordé.

---

# Vue générale du projet

## 1.1 Rappel des tâches

Tâches	Marco	Elias	Godric	Lucile
Developpement du joueur		R		S
Menus	S	R		
Interfaces	S			R
Scénarios et Dialogues	S			R
Musiques et sons	R	S		
Multijoueur	R	S		
PNJ	S			R
Ennemis basiques	S		R	
Boss		R	S	
Graphismes et Animations			S	R
Site Web		S	R	

Nous avons rajouté deux sections par rapport au cahier des charges pour plus de précision.

## 1.2 Moyens mis en place

Pour la réalisation de notre projet, nous avons mis en place plusieurs moyens de collaboration et de suivi. Tout d'abord, nous avons utilisé GitHub afin de travailler en collaboration sur les mêmes fichiers et de résoudre les conflits éventuels.

Nous avons également utilisé Discord pour communiquer, échanger des idées et travailler. C'est aussi sur cette plateforme que nous mettons tous les avancements, les problèmes rencontrés et les choses à faire.

En ce qui concerne le suivi de projet, nous avons utilisé Notion pour créer un tableau de bord de l'avancement actuel du projet et des deadlines d'un point de vue général.

Enfin, nous avons utilisé Google Docs pour collaborer sur la rédaction de documents tels que le scénario ou le GameDesign.

---

# Avancement du projet

## 2.1 Scénario

L'histoire est une partie importante de notre jeu, même si sa compréhension reste non nécessaire pour permettre au joueur d'avancer. Nous avons déjà écrit le fil directeur de l'histoire, afin de situer les ambiances des différents niveaux que le joueur devra parcourir. Il nous manque encore certains détails concernant l'histoire en règle général. Le joueur incarne un habitant vivant sur la planète Proxima, au tout début du jeu, il se retrouve dans un cachot. Il rencontre alors le premier PNJ du jeu, qui va lui donner les informations nécessaires pour débiter sa première quête. Pour les dialogues, nous avons fait quelques ébauches concernant les premiers niveaux.

## 2.2 Mécaniques de jeu

### 2.2.1 Mouvements et actions du joueurs

#### Mouvements du joueur

Il est essentiel pour un jeu Action-RPG que le joueur puisse se déplacer aisément dans l'environnement. Nous avons donc créé un script pour permettre au joueur de marcher, courir, sauter et de s'accroupir. Celui-ci utilise un système simple au travers des méthodes "transform.translate", "transform.Rotate" et "rigidbody.Addforce" pour le saut. Cependant, il arrivait que le joueur ait encore des problèmes de collisions avec son environnement. Nous avons donc dû baisser la vitesse de celui-ci. En passant sur ces méthodes, nous avons réglé beaucoup de problèmes.

Néanmoins, il arrive que le joueur ait encore des problèmes de collisions, par exemple il est souvent éjecté assez loin si un ennemi ou un PNJ nous pousse.

### **Attaque du joueur**

Le joueur peut également attaquer les ennemis grâce à son arme avec le clic droit. Lorsque son arme touche un ennemi (différencié des PNJs grâce au tag “enemy”). De plus, cette attaque dispose d’un compte à rebours et son arme perd sa capacité à faire des dégâts lorsque celui-ci n’attaque pas.

### **Lock on**

Au commencement, nous voulions que le joueur puisse bouger sa caméra librement autour de lui. Nous avons donc créé un script pour cela. Cependant, nous nous sommes dit que cela simplifierais le jeu si nous donnions la capacité de viser un ennemi au joueur. L’objectif du lock était que le joueur puisse placer sa caméra afin d’avoir une vue d’ensemble lui permettant de se déplacer tout en observant l’ennemi le plus proche pour simplifier le combat contre ce dernier. Toutefois, il n’est pas totalement opérationnel actuellement.

### **Ramassage d’un item**

Notre jeu est composé d’items que le joueur peut manipuler. Ainsi, il peut ramasser les objets qu’il trouve au sol, en sachant que ces différents items se placeront directement dans son inventaire si celui-ci n’est pas plein. Pour cela nous avons créé un script “Pickup” qui vérifie si le joueur n’est pas à proximité d’un item grâce au Physics.Raycast. Dans ce cas, alors le joueur pourra ramasser l’item. De plus, lorsqu’il effectue cette action, il ne peut plus bouger, ni attaquer.

## 2.2.2 Les différents Items

Comme dit précédemment, notre jeu possède différents items. Nous avons donc créé une classe Item héritant de la classe ScriptableObject. Il y a également différents types d'items dans le jeu, c'est pourquoi nous avons fait un script pour chaque type d'items héritant de la classe Item.

La classe Item possède les attributs suivant :

- Un nom
- Une description
- Le visuel de l'objet dans l'inventaire
- Le modèle 3D de l'objet
- Un booléen vérifiant si l'objet est stackable
- Le nombre maximum d'objets de même type stackable
- Une autre version du modèle 3D
- Le prix de l'objet

En effet, certains objets nécessitant une manipulation du joueur, tel que l'épée ou encore le bouclier, possèdent deux modèles 3D afin de faciliter le rendu en jeu. Pour le moment, il existe trois types d'objets en jeu : les équipements, les armes et les potions. Par la suite, nous implémenterons plusieurs autres types comme la nourriture ou encore les artefacts.



FIGURE 2.1 – Potion de force



FIGURE 2.2 – Bouclier

### 2.2.3 Système monétaire et magasins

Le joueur peut donc ramasser des items sur le sol, mais ce n'est pas la seule méthode pour obtenir des objets. En effet, afin de permettre l'expérience de jeu plus agréable, le joueur peut également se procurer les items de base en les achetant aux boutiques du village. Pour cela, nous avons inclus un système monétaire dans le jeu. Cette monnaie, appelée "Écu", pourra être trouvée par le joueur en tuant les différents ennemis et boss. Comme dit plus haut, le joueur peut voir sa monnaie en tout temps dans un emplacement spécifique. Cette statistique se met à jour quand le joueur achète des objets au magasin et prochainement lorsqu'il récupère de l'argent en tuant des monstres. Pour le moment il n'a aucun moyen de récupérer de l'argent. De plus, en passant sa souris sur un des items vendus au magasin, les informations le concernant apparaissent, c'est-à-dire son nom et sa description.

Pour faire le magasin nous avons créer une classe Slots avec les variables suivantes :

- L'item
- Le visuel de l'item
- Le prix de l'item
- Un booléen vérifiant si les informations concernant l'item sont ouvertes

Ce script est attribué à tous les emplacements. Cela nous permet de rendre le magasin dynamique si nous voulons changer les items présents ou si nous voulons créer d'autres boutiques.



FIGURE 2.3 – Boutique de Potions

## 2.2.4 Inventaires et fonctionnalités

### Inventaire principal

Pour gérer tous ces items, le joueur possède un inventaire. Dans celui-ci se trouve 12 emplacements. Certains objets peuvent être empilés, comme les potions par exemple, avec un maximum de 5 par pile. Il peut manipuler chacun des objets en cliquant dessus ou en utilisant les touches associées à chaque action. Les actions sont différentes selon le type de l'item, pour les équipements et les armes, le joueur peut les équiper, tandis que pour les potions, il peut les utiliser. Il peut également jeter les objets au sol ce qui les fera apparaître à côté de lui. Pour le moment, il peut seulement équiper les armes et les boucliers. Pourtant, il ne peut pas encore utiliser le bouclier et les divers consommables mais uniquement les armes.

### Inventaire rapide

Le joueur possède également une barre de trois emplacements ou chaque emplacement est réservé à un type d'objet spécifique. On l'appelle une barre d'inventaire rapide. Les objets qui sont actuellement équipés sur lui seront affichés dans cette barre d'inventaire rapide. Le premier emplacement est réservé pour une arme, le deuxième pour le bouclier et le troisième pour un consommable, que cela soit des potions ou de la nourriture. Cette barre d'inventaire rapide est toujours visible par le joueur. En appuyant sur la flèche du haut, il changera son arme actuelle par la suivante dans son inventaire. De même pour la flèche du bas qui lui permettra de changer son bouclier par le suivant dans son inventaire. Une fois l'épée équipée, il peut l'utiliser pour attaquer. Tout comme les magasins, en passant la souris sur un item d'un des deux inventaires, le joueur peut voir les informations sur l'item en question.

C'est le script "inventory" qui gère la plupart des choses qui sont liés à l'inventaire.

Nous avons créer dans ce script une classe "ItemInInventory" qui contient trois variables :

- L'item (de type Item)
- Le type d'inventaire
- Le nombre d'item

L'inventaire est géré par une liste d'ItemInInventory. Nous avons créé plusieurs méthodes pour faciliter son utilisation :

La méthode AddItem() qui va ajouter un item à l'inventaire, cette méthode est utilisée quand le joueur ramasse un item ou en achète un. La méthode RemoveItem() qui va supprimer un objet de l'inventaire, est utilisée lorsque le joueur jette un item au sol. La méthode RefreshContent() qui va mettre à jour le visuel de l'inventaire. Pour finir, des méthodes sont présentes pour vérifier si l'inventaire est rempli, ouvrir le panel d'action, pour utiliser, équiper ou encore jeter les objets.

Pour les emplacements de l'inventaire, nous avons utilisé la même méthode que pour ceux des magasins.

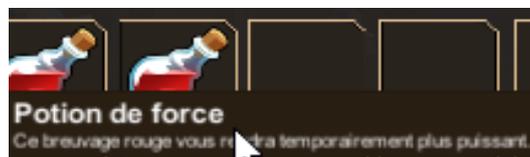


FIGURE 2.4 – Information concernant la potion de force

## 2.3 Interfaces et fonctionnalités

### 2.3.1 Menus

Concernant les menus du jeu, nous avons développé le menu principal, permettant de lancer une nouvelle partie, de charger une sauvegarde précédente, seulement en solo pour le moment, ou de quitter le jeu. Nous avons aussi fait un menu permettant de se connecter à une partie multijoueur. Ensuite, si on lance le jeu, nous disposons d'un menu pause, d'où nous pouvons revenir au menu principal, changer ses options, ici certaines touches et le plein écran du jeu, ou encore sauvegarder. Les données sauvegardées sont mises dans un fichier texte et le jeu se charge de les lire pour bien repositionner le joueur ou remettre son inventaire correctement. Actuellement, celle-ci est accessible dans un fichier texte, permettant de modifier les sauvegardes à n'importe quel moment du jeu. Dans le futur, nous pourrions possiblement envisager le fait de l'encrypter. Enfin, un écran de Game Over est mis en place si le joueur meurt.



FIGURE 2.5 – Menu Principal du Jeu



FIGURE 2.6 – Menu Pause



FIGURE 2.7 – Menu de Sauvegarde



FIGURE 2.8 – Menu des options

### 2.3.2 Multijoueur

En ce qui concerne le multijoueur, nous avons posé les bases en utilisant Photon comme il était prévu. En essayant de connecter deux joueurs, certains problèmes sont survenus car les explications que l'ont suivait portaient sur un jeu en deux dimensions. Par exemple, la caméra du joueur client étaient celle du joueur hôte et inversement pour la caméra du joueur hôte. La solution était de gérer correctement l'appartenance des points de vues dans le script du joueur intervenant dans le multijoueur. Ainsi, grâce aux composants liés au "Photon View", notre multijoueur permet à deux joueurs de se connecter sur la map "Village" impliquant la synchronisation des positions et des animations des joueurs. Toutefois, tout n'est actuellement pas implémenté, par exemple les menus ou encore les IAs.

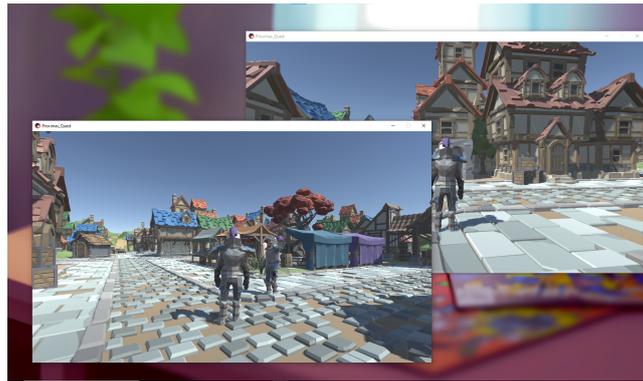


FIGURE 2.9 – Illustration du Multijoueur

### 2.3.3 HUD

A l'écran le joueur peut voir plusieurs informations essentielles. Tout d'abord sa barre de vie, fonctionnant par progression, ses points de vie exacts n'étant pas visibles. Il peut également voir sa monnaie actuelle et son inventaire rapide.



FIGURE 2.10 – Interface HUD

### 2.3.4 Indications pour le joueur

Le joueur étant capable d'interagir avec son environnement, nous avons donc mis en place une boîte d'informations dynamique qui s'affiche si nécessaire. Cela lui donne des informations concernant la touche à appuyer et l'action qu'il veut effectuer. Une deuxième boîte d'informations, elle aussi dynamique, a été mise en place pour l'informer des actions effectuées ou de l'impossibilité de certaines d'entre-elles.



FIGURE 2.11 – Action possible



FIGURE 2.12 – Action réalisée par le joueur

## 2.4 Intelligences artificielles

### 2.4.1 Ennemis

Pour le moment, nous avons créé un type d'ennemi, le soldat. Nous lui avons implémenté un comportement de patrouille, lors de celui-ci, des temps de pauses régulier lui ont été intégrés pour un air plus naturel.

Ainsi, les ennemis se déplacent de manière autonome dans l'environnement en suivant un chemin prédéfini et en utilisant des nœuds placés à des points clés de la carte. Lorsque le joueur est assez proche, les ennemis détectent sa présence et activent leur comportement d'attaque. Ils poursuivent le joueur jusqu'à ce qu'il soit hors de portée ou jusqu'à ce que l'un des deux soit vaincu. Bien que les ennemis basiques soient programmés de manière relativement simple, leur comportement ajoute une couche d'imprévisibilité et de difficulté au jeu, offrant ainsi une expérience plus immersive pour les joueurs.

### 2.4.2 PNJ

En accord avec la recherche sur le fonctionnement des ennemis, nous avons aussi fonctionné avec la notion de "State Machine". Nous avons donc créé un PNJ qui se promène dans le village à l'aide deux comportements dont celui nommé "Idle" et l'autre "Promenade". Les scripts associés à ces états lui permettent de marcher de point en point donnant l'impression qu'il se promène dans le village. De plus, des retours en Idle sont programmés pour un air plus naturel.

En découle le PNJ interactif, le prêtre, dont les scripts ont été modifiés par rapport au simple PNJ pour que celui-ci s'arrête et nous regarde lorsque l'on s'approche de lui à l'aide d'une "Box Collider". Par la suite, à l'aide du Canvas, une proposition pour dialoguer avec lui s'affiche. En appuyant sur "F", la boîte de dialogue accompagnée de son script permettant le défilement du texte s'affiche. Lorsque le dialogue est fini, après la fermeture de la boîte, il est possible de recommencer l'opération. Pour le moment, les dialogues ne sont pas dynamiques car la boîte

de dialogue du prêtre nous a, en premier lieu, permis l'apprentissage du fonctionnement de la boîte de dialogue.

Enfin, deux marchands ont été créés, l'alchimiste et l'armurier. Nous pouvons interagir avec eux en entrant dans leurs "Box Collider". À l'aide de Canvas et de scripts, ces shops deviennent fonctionnels.



FIGURE 2.13 – PNJ : Marchand



FIGURE 2.14 – PNJ



FIGURE 2.15 – PNJ : Prêtre

### 2.4.3 Boss

Nous avons implémenté un premier boss nommé "Noopy", qui serait le boss d'introduction du jeu. Il s'agit d'un chevalier muni d'une masse, expliquant son paternelle d'attaques lourdes et lentes. Cependant, seule l'attaque de base a été implémentée actuellement. Les Boss se trouveront dans une salle, qui, à l'entrée du joueur, affichera la vie et le nom du boss et fermera la salle derrière lui. Pendant le combat, une fois que le boss a atteint la moitié de sa vie, il rentre dans une seconde phase, ses dégâts et sa résistance sont alors doublés et il inflige un malus au joueur, ici une division par 2 de ses dégâts.

En ce qui concerne son IA, elle est similaire à celle des ennemis basiques, puisque le boss sera en "Idle" en attendant que le joueur entre dans la salle, et se mettra à pourchasser le joueur si celui-ci entre. Cependant, cette IA n'est pas la version finale. En effet, ce n'est pas celle indiquée dans le cahier des charges. Ainsi, plus tard, celui-ci aura 3 attaques en plus.



FIGURE 2.16 – Barre d'HP du boss

## 2.5 Graphismes

Au tout départ, nous voulions apprendre à utiliser Blender pour faire les différentes maps et objets composant le jeu, mais nous nous sommes vite rendu compte que ce n'était pas si facile et que nous allions manquer de temps. Nous avons donc décidé d'acheter des packs d'assets pour nous aider avec la conception des différentes cartes. Nous avons ensuite rassemblé ces éléments pour pouvoir créer des cartes entières représentatives du monde médiéval-fantastique dans lequel se déroule le jeu. Pour ce qui est du graphisme des interfaces, nous nous sommes aidé d'images correspondants à nos besoins récupérés sur internet et avons utilisés le type TextMeshPro pour un rendu visuel plus satisfaisant.

### 2.5.1 Maps

Notre jeu est donc composé de plusieurs zones, 3 niveaux et une zone annexe qui sert comme intermédiaire entre les différents niveaux et qui permet au joueur un peu de repos. Nous avons donc prévu de faire un premier niveau qui se situera dans des cachots et dans des grottes. Ce niveau n'est pas encore fini mais est en bonne voie. Concernant le niveau 2, nous avons décidé de le faire dans une forêt et pour le niveau 3, un château. Nous avons quelques morceaux de ces deux derniers mais nous ne les avons pas encore assemblés. Le niveau 1, qui se déroule dans un

cimetière, n'est quant à lui pas du tout commencé. Pour finir, la zone annexe, qui est un village, est totalement finie même si il faudrait rajouter quelques arbustes et des décors afin de le rendre plus vivant.

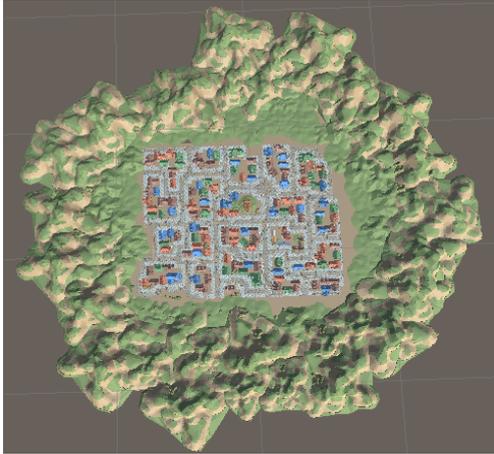


FIGURE 2.17 – Carte : Village



FIGURE 2.18 – Boss : Noopy

## 2.6 Animations

En ce qui concerne les animations, nous avons utilisé le site Mixamo pour les générer, afin d'animer les joueurs, les PNJs et les ennemis. Bien que ces animations ne soient pas encore parfaitement fluides, elles permettent déjà au joueur de se mouvoir avec aisance dans l'environnement du jeu. Nous envisageons d'effectuer des ajustements supplémentaires en modifiant les animations déjà mises en place pour améliorer l'aspect visuel du jeu.

---

## Bilan sur l'avancement

Tâches	Prévu	Réalisé
Développement du joueur	50%	65%
Scénarios et Dialogues	75%	75%
Musiques et sons	0%	0%
Multijoueur	10%	25%
PNJ	50%	50%
Ennemis basiques	25%	45%
Boss	20%	30%
Graphismes et Animations	30%	40%
Site Web	0%	0%

Si nous comparons notre avancement par rapport aux prévisions faites dans le cahier des charges, nous sommes plus ou moins en phase avec celle-ci.

Au niveau du développement du joueur, il reste quelques fonctionnalités à développer comme la mise en place d'un système d'expérience ou le fait de pouvoir équiper l'armure. Avec ce que nous avons déjà réalisé, nous avons une meilleure vue d'ensemble sur ce qu'il faudra faire pour améliorer le joueur. C'est surtout l'ajustement des fonctions actuelles qui vont nous prendre le plus de temps sur ce qu'il reste à faire.

Le scénario est bien avancé mais c'est surtout les dialogues sur lesquels nous n'avons pas encore beaucoup réfléchi. Au niveau de la musique et du son, comme prévu nous nous y sommes pas du tout attelé pour le moment. En effet, il était nécessaire d'avoir les bases de notre projet afin de mettre en place des musiques cohérente avec l'ambiance du jeu.

Le développement du multijoueur représente l'un des défis les plus importants pour la suite de notre projet. Pour le moment, les joueurs peuvent se connecter ensemble mais tout le reste n'est pas encore géré que ce soit le comportement des intelligences artificielles ou encore la gestion des menus.

Pour les intelligences artificielles, nous sommes assez bien avancé. Nous avons déjà créé des PNJs avec divers comportements, ce qui nous permettra de nous baser sur cela afin de faire les suivants selon nos envies. Pour les ennemis basiques, nous avons pour le moment un seul comportement et une attaque. Comme pour les PNJs, il nous suffira de prendre ce modèle et de l'ajuster en fonction des types d'ennemis. Certains nécessiteront sûrement plus de travail mais nous sommes globalement avancés à ce niveau. Enfin, pour les boss, nous avons pour le moment réalisé une seule attaque. Il nous faudrait donc implémenter plusieurs autres types d'offensives.

Enfin, nous avons déjà une map totalement finie et les autres sont en bonne voie de réalisation. Pour les personnages, les ennemis et les objets du jeu, ils sont en accord avec les prévisions établies dans le cahiers des charges. En ce qui concerne les animations, ils nous reste surtout des ajustements à faire mais l'ensemble global est fait. Et pour finir la création du site web n'a pas encore commencé.

---

## Développement futur

Tâches	2ème soutenance
Développement du joueur	80%
Menus	90%
Interfaces	80%
Scénarios et Dialogues	100%
Musiques et sons	20%
Multijoueur	30%
PNJ	100%
Ennemis basiques	60%
Boss	40%
Graphismes et Animations	65%
Site Web	60%

Voici un rappel des pourcentages fixés pour la deuxième soutenance dans le cahier des charges avec des précisions sur les menus et les interfaces.

### 4.1 Fonctionnalités à implémenter dans le futur

Pour cette première soutenance nous avons donc réussi à atteindre tous nos objectifs. Nous devons maintenant penser à toutes les fonctionnalités restantes à implémenter.

Au niveau du développement du joueur, il nous reste à réaliser un système d'expérience avec différents domaines, l'action de pouvoir parer avec son bouclier et de pouvoir équiper son armure. L'utilisation de tous les items sera aussi un point important à développer. Il faut également finaliser les différents magasins et ajouter des possibilités pour que le joueur puisse gagner de l'argent. Nous allons aussi ajouter d'autres types d'items tels que la nourriture ou encore les artefacts. Le lock on a déjà été implémenté comme précisé plus haut mais il n'est cependant pas totalement opérationnel, nous allons donc revoir sa conception.

En terme d'affichage, nous allons rendre la boîte de dialogue dynamique. Les sauvegardes, elles, ne peuvent pour le moment sauvegarder qu'une partie des éléments à sauvegarder. Nous allons donc améliorer le système pour que l'ensemble du jeu soit sauvegardé. Les scénarios et les dialogues sont à finaliser et l'ensemble du multijoueur est encore à faire même si il est déjà entamé.

Pour les graphismes, la carte du village doit être revue pour ajouter des décors. Pour les autres niveaux, il faut encore finir le processus de création. Nous avons déjà la plupart des modèles 3D pour les PNJs, néanmoins, ceux des ennemis restent à trouver. Le personnage principal que le joueur incarne doit aussi être changé.

Au niveau des intelligences artificielles, nous devons diversifier certains comportements et surtout ajouter plusieurs types attaques au boss. Pour les PNJs, il nous faut implémenter un modèle qui puisse donner des objets au joueur.

Nous allons aussi commencer à implémenter des musiques et des sons dans le jeu et à faire le site internet. Pour finir, nous allons ajuster certaines choses et régler les problèmes.

## 4.2 Améliorations possibles

Nous avons donc une liste d'éléments prédéfinis à implémenter. Cependant nous ne sommes pas fermes à des améliorations possible pour faire évoluer notre jeu. En effet, nous avons quelques idées qui n'étaient pas prévu mais qui pourrait améliorer l'expérience du joueur.

L'idée d'une interface où les quêtes du joueur seront stockées et mises à jour nous est venue. Cette interface lui permettrait de visualiser le chemin qu'il a parcouru et les quêtes en cours. Cela pourrait aussi faciliter la compréhension de l'histoire.

Le joueur a accès à certaines informations tel que sa barre de vie ou encore sa monnaie. L'ajout d'une Mini Map pourrait améliorer sa vision des cartes et faciliter ses déplacements.

Pour les armes possible du joueur, nous avons initialement prévu des épées et des haches. Pour améliorer sa capacité et diversité d'attaques, nous avons pensé à ajouter un arc. Cela permettrait aussi de pouvoir implémenter des ennemis et boss différents des types actuels.

Les sauvegardes étant actuellement modifiables en dehors du jeu très facilement, nous pourrions ajouter l'encryptage des données sauvegardées.



---

## Conclusion

En somme, nous avons mis en place les fondations nécessaires pour un gameplay d'action-RPG satisfaisant, mais nous prévoyons d'ajouter d'autres fonctionnalités pour améliorer encore l'expérience de jeu.

De part la construction séparée des divers éléments présentés, nous pourrions implémenter les niveaux puis en découlera celle du scénario pour achever l'élaboration de la campagne du jeu et de ses diverses fonctionnalités.

Nous sommes assez confiant pour la suite du projet vu notre avancée actuelle.