

## LOUP GAROU

Professeurs référents :  
M.CATELOIN Stéphane



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Présentation générale . . . . .	2
1.2	Principe du jeu . . . . .	2
1.3	Les personnages du Jeu . . . . .	3
1.4	Déroulement du Jeu . . . . .	5
1.5	Algorithme du Jeu . . . . .	6
<b>2</b>	<b>Les éléments clés</b>	<b>7</b>
2.1	Section Programmation . . . . .	7
2.1.1	Programmation . . . . .	7
2.1.2	Algorithmique . . . . .	7
2.1.3	Interaction . . . . .	8
2.2	Section Base de donnée . . . . .	8
2.3	Section Réseau . . . . .	9
2.3.1	Protocole de Communication . . . . .	9
2.3.2	Développement des clients et du serveur . . . . .	9
2.3.3	La sécurité . . . . .	9
2.4	Section Communication . . . . .	10
<b>3</b>	<b>Les technologies employées</b>	<b>11</b>
3.1	Partie Front-End . . . . .	11
3.2	Partie Back-End . . . . .	11
3.3	Partie Base de donnée . . . . .	12
3.4	Partie Réseau . . . . .	12
<b>4</b>	<b>Votre vision du rôle de chef de projet</b>	<b>13</b>
<b>5</b>	<b>Tâches et équipes</b>	<b>14</b>
5.1	Les tâches . . . . .	14
5.2	Les Livrables . . . . .	16
5.3	Le diagramme de Gantt . . . . .	16

# Chapitre 1

## Introduction

### 1.1 Présentation générale

Le "Loup-Garou" est un jeu de société d'ambiance dans lequel chaque joueur incarne un rôle. Celui-ci est édité pour la première fois en 2001 par la société *Lui-même*, éditeur français. Une partie peut être lancée avec un minimum de 5 joueurs et peut s'étaler sur une vingtaine de minutes.

Le "Loup-Garou" est un jeu de rôles populaires qui implique une bonne dose de stratégie, de communication et de méfiance.

### 1.2 Principe du jeu

Le concept de base du jeu "Loup-Garou" est de déterminer qui est le loup-garou parmi les joueurs. Les joueurs sont divisés en deux groupes : les villageois et les loups-garous. Les villageois cherchent à identifier et à voter pour les loups-garous pour les éliminer, tandis que les loups-garous cherchent à tuer tous les villageois sans être découverts.

Une partie de "Loup-Garou" se déroule selon une alternance de nuits et de jours. **Durant les nuits**, les joueurs ont tous les yeux fermés, puis à tour de rôle le meneur demandera aux personnages nocturnes de se réveiller pour utiliser leur pouvoir. Lorsque tous les rôles nocturnes auront été appelés, le meneur demande à tout le village de se réveiller.

**Chaque jour, au réveil du village**, Les victimes éventuelles des personnages nocturnes sont révélées. Puis tous les joueurs survivants débattront pour éliminer par vote l'un d'entre eux dans le fragile espoir de garantir la survie du Village...

Le but de chaque joueur dépendra de son identité secrète.

La partie peut prendre fin à différents moments (selon les rôles présents dans la partie), basiquement si :

- Tous les loups ont été démasqués et exclu du village (le village l'emporte)
- Les loups ont tué assez de villageois pour prendre le contrôle du village (les loups l'emportent)

Le jeu de cartes se joue avec un maître du jeu qui permet les interactions entre les joueurs. Une version informatique simulerait donc cette tâche afin de dérouler les différentes actions de la partie.

## 1.3 Les personnages du Jeu

**\*Les Loups-Garous :** Chaque nuit, ils dévorent un Villageois. Le jour, ils essaient de masquer leur identité nocturne pour échapper à la vindicte populaire. Leur nombre peut varier suivant le nombre de joueurs. En aucun cas un loup-garou ne peut dévorer un autre loup-garou.



**\*Le Simple Villageois :** Il n'a aucune compétence particulière. Ses seules armes sont la capacité d'analyse des comportements pour identifier les Loups-Garous, et la force de conviction pour empêcher l'exécution de l'innocent qu'il est.



**\*La Voyante :** Chaque nuit, elle voit la carte d'un joueur de son choix. Elle doit aider les autres Villageois, mais rester discrète pour ne pas être démasquée par les Loups-Garous.



**\*Cupidon** : En décrochant ses célèbres flèches magiques, Cupidon a le pouvoir de rendre 2 personnes amoureuses à jamais. La première nuit, il désigne les 2 joueurs amoureux. Cupidon peut, s'il le veut se désigner comme l'un des deux Amoureux. Si l'un des Amoureux est éliminé, l'autre meurt de chagrin immédiatement. Un Amoureux ne doit jamais éliminer son aimé, ni lui porter aucun préjudice.



**\*La Sorcière** : Elle sait concocter 2 potions extrêmement puissantes : **Une potion de Guérison**, pour ressusciter le joueur dévoré par les Loups-Garous, **Une potion d'Empoisonnement**, utilisé la nuit pour éliminer un joueur. La Sorcière doit utiliser chaque potion **une seule fois** dans la partie. Elle peut se servir de ses deux potions la même nuit. Le matin, suivant la ou les potions utilisées, il pourra y avoir deux joueurs éliminés ou aucun ! La Sorcière peut également utiliser la potion de guérison à son profit, et donc se guérir elle-même si elle vient d'être dévorée par les Loups-Garous.

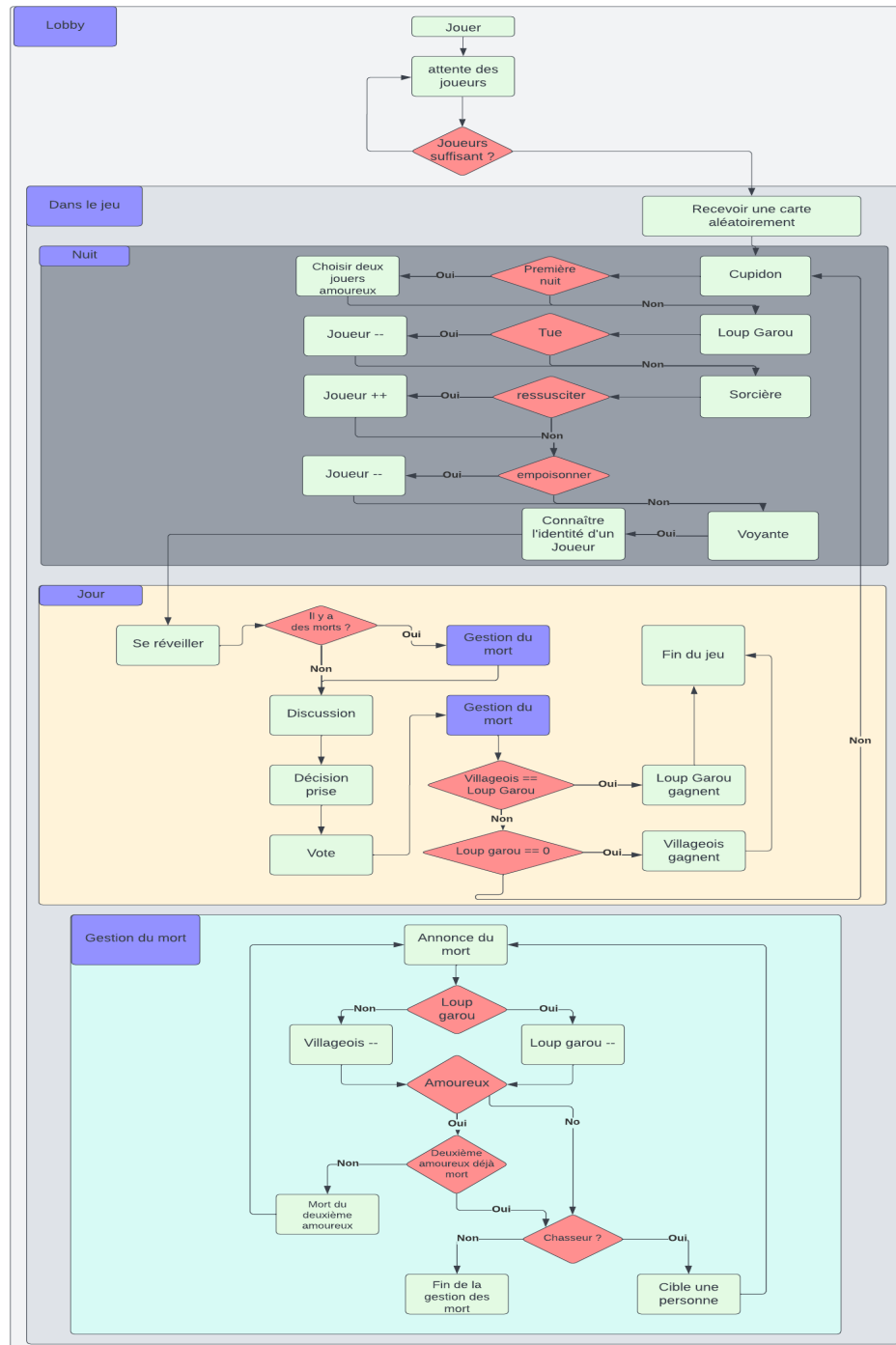


**\*Le Chasseur** : S'il se fait dévorer par les Loups-Garous ou éliminer malencontreusement par les villageois, le Chasseur doit répliquer avant de rendre l'âme, en éliminant immédiatement n'importe quel autre joueur de son choix.



## 1.4 D roulement du Jeu

L'UML suivant permet de repr senter les diff rentes  tapes du jeu du "Loup-Garou".



## 1.5 Algorithme du Jeu

```
Attendre des joueurs (minimum 5) [attendre d'avoir minimum 5 clients]
Distribution aléatoire des rôles [envoyer à chaque client son rôle]

TANT QUE (condition de fin == 0) {
    Nuit [envoyer aux clients que c'est la nuit] [afficher un thème « nuit »]
    | • Appeler les rôles dans l'ordre d'apparition des joueurs en vie
    |   [appeler les rôles dans l'ordre] [afficher la carte de celui qui joue]
    | • Effectuer l'action des rôles
    |   [envoyer l'action au serveur] [recevoir l'action et l'interpréter]
    Jour [envoyer aux clients que c'est le jour] [afficher un thème « jour »]
    | • Annoncer le mort (ou non) et son rôle
    | • Si (1er jour) -> Élire un maire
    |   [afficher dynamiquement les votes des joueurs]
    |   [interpréter les votes à la fin du timer]
    | • Voter pour quelqu'un à écarter du village
    |   [afficher dynamiquement les votes des joueurs]
    |   [interpréter les votes à la fin du timer]
    | • Annoncer le rôle du mort voté
    |   [Retirer le(s) joueur(s) mort(s) de la liste des joueurs à appeler]
    Si (plus de loups) -> condition de fin = 1
    Si (nombre de villageois == nombre de loups) -> condition de fin = 1
    ## plein de possibilités de fin supplémentaires en fonction des rôles ajoutés ##
}

Légende : [Action serveur] [Action client]
```

# Chapitre 2

## Les éléments clés

### 2.1 Section Programmation

#### 2.1.1 Programmation

La partie programmation d'un jeu vidéo est la tâche complexe et chronophage qui consiste à créer un jeu vidéo à partir de zéro. D'un choix collectif, nous avons décidé que la programmation se fera en C# afin de bénéficier des nombreux avantages apportés par le moteur graphique Unity. Par exemple, celui-ci nous permettra de facilement exporter notre projet PC en une version APK et/ou web. Cet outil nous permettra aussi d'établir une interface claire et simple à comprendre afin de conforter la jouabilité et l'expérience de jeu des joueurs. Cet objectif passera aussi par la « colonne vertébrale du jeu », la partie algorithmique.

#### 2.1.2 Algorithmique

La partie algorithmique désigne l'ensemble des instructions qui déterminent le déroulement du jeu. Elle tient compte des actions des joueurs et décide des réponses appropriées. Dans notre cas, un exemple d'algorithme généraux a été donné ci-dessus 1.5. Les divers rôles seront développés afin d'offrir aux joueurs des possibilités de jeu variées, ce qui garantit que chaque partie sera différente et implique des stratégies et des décisions uniques. Aussi, dans le but d'améliorer la jouabilité, des intervalles de temps seront laissés aux joueurs pour qu'ils débattent et se forment une opinion avant d'émettre un vote qui fera avancer la partie, à leur avantage ou non... Ainsi, un aspect collaboratif sera également mis en avant notamment à l'aide de discussions sur un chat écrit pour tout le monde, un chat écrit pour le couple, un chat écrit pour les loup-garou (la nuit), etc...



### 2.1.3 Interaction

Le joueur aura plusieurs interactions possibles, dès le lancement du logiciel. En effet, pour le bon déroulement du jeu, le joueur aurait besoin dans un premier temps de se connecter ou de créer un compte. Depuis cet écran, le joueur aura la possibilité d'accéder aux paramètres du jeu, qui lui permettront notamment de changer de langue. Il aurait ensuite besoin de rejoindre la partie souhaitée, puis la partie pourra commencer s'il y a tous les joueurs. Une fois le jeu commencé, un mécanisme de sélection des joueurs devra être possible, système qui sera utile durant la nuit pour les rôles à actions (ex : Loups-Garous, Sorcière, Voyante...), cette sélection sera également utile lors du jour afin de pouvoir voter chaque jour pour le joueur à éliminer, raison pour laquelle il faudrait que les joueurs soient représentés de manière suffisamment simple afin de pouvoir les sélectionner aisément. L'écran principal devra également posséder une boîte de communication permettant notamment aux joueurs de débattre chaque jour, ou aux loups-garous afin de pouvoir communiquer durant la nuit. Lorsque le joueur sera mort, il aura la possibilité de voir le rôle des autres joueurs en les sélectionnant. Les morts disposent d'une boîte de communication à part. Une fois la partie terminée, un écran annoncera le vainqueur et révélera le rôle de chacun, et celui-ci devra permettre de soit, quitter la partie et de retourner à l'écran titre, soit de relancer la partie dans le même lobby. Unity facilitera la mise en place de ces différents écrans et interactions qui leur seront liées.

## 2.2 Section Base de donnée

Il est indéniable que la mise en place d'une section consacrée à la gestion de base de données est cruciale pour garantir le bon fonctionnement du jeu en question. À cet égard, il convient de souligner que cette section consistera principalement en la sauvegarde des données relatives aux utilisateurs inscrits, telles que leur adresse mail, leur identifiant et leur mot de passe, ainsi qu'en la sauvegarde des scores obtenus par les joueurs. Il est également prévu de mettre en place des mesures de sécurité efficaces, notamment en matière de chiffrement et de hachage des mots de passe et des données, afin de protéger les données des utilisateurs. Enfin, il convient de noter que cette section sera mise en place en collaboration étroite avec l'équipe réseau, afin de garantir une communication fluide avec les clients, tout en respectant les contraintes imposées par le jeu.

## 2.3 Section Réseau

### 2.3.1 Protocole de Communication

Il est crucial de mettre en place un système de communication fiable entre le serveur et les clients pour garantir le bon fonctionnement du jeu. C'est pour cette raison que le protocole TCP a été choisi pour assurer cette communication.

En utilisant TCP, non seulement nous garantissons la livraison des données en vérifiant la correcte réception des informations et en évitant la perte de données, mais nous pouvons également éviter une surcharge du réseau en gérant efficacement le flux de données entre le serveur et les clients.

De plus, il existe de nombreuses bibliothèques qui facilitent l'utilisation du protocole TCP, ce qui facilite le développement et la maintenance du système.

En somme, l'utilisation de TCP pour la communication entre le serveur et les clients est une décision stratégique qui assure une communication fiable, un contrôle efficace du flux de données et une facilité d'utilisation pour les développeurs tout en assurant le bon fonctionnement du jeu.

### 2.3.2 Développement des clients et du serveur

Le client est chargé de traduire les interactions effectuées en messages compréhensibles pour le serveur, qui, à son tour, interprète ces messages et les transmet à l'algorithme de jeu. Le serveur est également responsable de renvoyer aux clients connectés des informations sur l'évolution de la partie, telles que l'état du jeu (jour ou nuit), l'état des cartes, les décès, les rôles et les votes. En tant que cerveau du jeu, le serveur est chargé de faire évoluer la partie en transmettant les mises à jour aux clients pour qu'ils puissent les afficher à l'écran.

### 2.3.3 La sécurité

Dans le but de garantir un niveau élevé de sécurité, il a été décidé d'utiliser un certificat basé sur le système de cryptographie RSA. Ce choix est justifié par plusieurs raisons. Tout d'abord, RSA assure une sécurité accrue en cryptant les données de manière à ce qu'elles ne puissent pas être lues par des tiers non autorisés. Cela protège efficacement les informations sensibles telles que les informations de compte. Ensuite, RSA est un algorithme de chiffrement bien établi et largement utilisé, il existe donc de nombreuses bibliothèques logicielles et des implémentations pour différents systèmes d'exploitation et langages de programmation, ce qui facilite son utilisation. De plus, RSA est un algorithme efficace qui permet de chiffrer et déchiffrer des données rapidement, même pour des tailles de clé élevées.

Enfin, nous avons eu l'occasion de manipuler ce système RSA, ce qui facilitera encore plus son usage.

## 2.4 Section Communication

Notre projet vise à atteindre une cible constituée principalement d'adolescents et de jeunes adultes. Pour atteindre cet objectif, nous avons mis en place une stratégie de communication qui vise à informer les membres de cette cible de l'existence de notre projet, à susciter leur intérêt et à recueillir leurs avis sur le jeu. Pour ce faire, nous créerons des comptes sur les réseaux sociaux pertinents, réaliser des affiches et des visuels standard et interactifs (sondages, questionnaires), partager les évolutions de notre projet, diffuser une vidéo de démonstration du jeu et distribuer des flyers. Toutes ses actions ont pour but de créer une communauté autour de notre Jeu du "Loup-Garou".

# Chapitre 3

## Les technologies employées

### 3.1 Partie Front-End

Le Front-End sera responsable de l'apparence et de l'expérience visuelle du jeu. Le but sera de concevoir une interface utilisateur claire et facile à utiliser pour que les joueurs puissent naviguer aisément. Nous avons décidé d'utiliser Unity en combinaison avec le C# choisi pour le développement Back-end. Nous avons choisi Unity pour sa compatibilité avec C#, ainsi que pour ses fonctionnalités de création du design facile à utiliser, telles que l'aperçu en temps réel et la possibilité d'importer des formats 2D et 3D, des ressources audio et vidéo. En utilisant les plugins disponibles dans Unity, nous avons également la possibilité de faire une exportation facile vers des formats tels que Windows et Android. En outre, Unity est très bien documenté et dispose de nombreuses ressources, comme des tutoriels, ce qui facilite la mise en place de la partie graphique de notre projet. Il est également intéressant de noter que selon le site officiel d'Unity, environ 50 % des jeux sur tous les appareils sont créés avec Unity.

### 3.2 Partie Back-End

La partie technique et fonctionnelle du jeu est mise en place dans cette étape. Pour cette partie, nous avons choisi d'utiliser le langage C# pour le back-end car il est polyvalent et facile à apprendre, offrant une variété d'outils pour la programmation de jeux vidéo. De plus, notre expérience en Programmation Orientée Objets nous permet de nous adapter rapidement à ce langage. Enfin, C# nous permet également de facilement intégrer la partie front-end en utilisant des moteurs graphiques.

### 3.3 Partie Base de donnée

Pour gérer les données de tous les utilisateurs inscrits dans notre jeu, comme leurs scores, identifiants et mots de passe, ainsi que d'autres données nécessaires à la sauvegarde, il est important d'utiliser des outils performants et adaptés. C'est pour cette raison que nous avons choisi d'utiliser MySQL en tant que système de gestion de base de données (SGBD) pour notre projet. Cette décision s'explique par plusieurs raisons.

Tout d'abord, MySQL offre une architecture client/serveur qui assure la sécurité des données avec un accès multi-utilisateur.

De plus, MySQL est le SGBD le plus populaire sur le marché, ce qui facilite l'accès à des documents de référence en cas de problème. Il prend en charge plusieurs langages de programmation et sa gestion est plus facile.

Enfin c'est un outil adapté pour l'équipe car tous les membres sont familiers avec ce système de gestion de base de données.

### 3.4 Partie Réseau

Pour la partie réseau de notre projet, nous avons choisi d'utiliser les bibliothèques et frameworks dédiés à C#, notamment la bibliothèque Xamarin.Essentials. Cette dernière est basée sur le framework Xamarin et contient toutes les méthodes nécessaires pour gérer les différents protocoles réseau courant, tels que TCP. Elle offre également une large gamme de classes pour gérer différents protocoles réseau courant, même si dans notre projet nous nous baserons sur le protocole TCP pour toute la transmission.

En utilisant Xamarin.Essentials, nous bénéficions également d'une facilité de maintenance et de mise à jour des applications en utilisant une seule base de code pour les différentes plateformes. Cela permet également de réutiliser le code existant écrit en C# pour les fonctionnalités réseau sur les applications mobiles, étant donné que l'application sera exportée en version mobile. Si nécessaire, nous pourrions éventuellement utiliser les technologies .NET avec les fonctionnalités compatibles avec la plateforme mobile.

# Chapitre 4

## Votre vision du rôle de chef de projet

**Chef de Projet :** BOUCHTITA Achraf

\*Un chef de projet doit posséder des compétences informatiques et techniques pour comprendre les éléments du code et coordonner l'équipe en cas d'incohérences ou de malentendus.

\*Il dirige les réunions du groupe et définit l'ordre du jour en fonction de l'avancée du projet et des délais.

\*Il tranche les débats au sein du groupe et avoir suffisamment de recul pour prendre des décisions réalistes en ayant une vision globale du projet.

\*Il doit être capable d'imposer des choix nécessaires et de gérer les conflits au sein de l'équipe.

\*Il doit s'adapter et être flexible pour faire face aux contraintes et exiger le travail de l'équipe dans les délais impartis.

\*Il doit faire des comptes rendus hebdomadaires pour mettre à jour sur l'avancée du projet.

\*Il doit être le lien entre le tuteur de projet et l'équipe de développement pour assurer une communication efficace et une coordination adéquate.

# Chapitre 5

## Tâches et équipes

### 5.1 Les tâches

Le partage des tâches est un processus important pour une équipe efficace. Dans le cadre de notre projet, nous avons identifié cinq parties majeures pour lesquelles nous avons attribué différents membres de notre équipe. Cette répartition des tâches est essentielle pour une mise en œuvre efficace de notre projet. Elle permet de maximiser les compétences de chacun de nos membres d'équipe et de réduire les risques de surcharge de travail pour un individu. Cette répartition permet également de favoriser la communication et la collaboration au sein de notre équipe, facteurs fondamentaux pour la réalisation réussie de notre Jeu du "Loup-Garou". Il est important de noter que cette répartition est effectuée de manière équitable et que tous les membres de notre équipe sont consultés et impliqués dans le processus de décision. Il est important de noter que, en plus de la répartition initiale des tâches, des ajustements peuvent être apportés au fur et à mesure de l'avancement du projet. Ainsi, si une partie est achevée, les membres de l'équipe concernés peuvent être redéployés sur d'autres parties, tout cela sous la supervision du chef de projet qui assure la gestion globale du projet. Cette flexibilité permet une adaptation efficace aux éventuels changements ou imprévus qui peuvent survenir dans la réalisation d'un projet.

Sous la direction du **Chef de Projet** : BOUCHTITA Achraf

**Partie Général** : Toute l'équipe

- Validation du Cahier de charge
- Spécification des différentes parties

**Partie FRONT-END :** KLOTZ Jennifer & DELALA Wassim

- Formation en Unity et C#
- Création des maquettes de l'interface
- Graphisme Obligatoire
- Mise à jour des interfaces
- Adapter en Version Mobile
- Test et Optimisation

**Partie BACK-END :** HENTZ Kévin & EL BOUHI Houssam & DIDIER Benjamin

- Formation en C#
- Création des éléments du jeu (Définition des classes ...)
- Implémentation du jeu de base (Avec 2 rôles seulement)
- Implémentation de l'intégralité des rôles
- Compléter le déroulement du jeu
- Test et Optimisation

**Partie Réseau :** ATMIMOU Mohammed Arezki & BOUZIDI Redha

- Formation en C# et en Xamarin.Essentials
- Création Client/Serveur de base
- Configuration du Multi-Client
- Configuration du Multi-Partie
- Sécurisation des échanges de paquets
- Sécurisation du jeu (Anti-Triche)
- Adaptation multiplateforme
- Test et Optimisation

**Partie Base de Donnée :** DIOP Sidy & COLAKOGLU Cenk

- Familiarisation avec MySQL et Formation C#
- Schéma entité association
- Schéma relationnel
- Modélisation / création de la BDD
- Connexion de la base de donnée au Serveur
- Optimisation et Test

**Partie Communication :** KLOTZ Jennifer & DELALA Wassim

- Création des comptes sur les réseaux sociaux
- Réalisation d'affiches et visuels
- Partager l'évolution du développement du jeu
- Partager la sortie de la version Alpha
- Partager la version mobile du jeu

**Partie Integration :** Toute l'équipe

- Integration version Alpha
- Integration version final



## 5.2 Les Livrables

**Version Alpha** : Les livrables de ce projet comprennent la mise en place des fonctionnalités clés pour la version alpha. Cela inclut la création d'un système de compte utilisateur, avec les fonctionnalités de connexion et déconnexion. La communication entre les joueurs sera possible grâce à un système de chat intégré. En outre, un algorithme sera implémenté pour l'attribution aléatoire des rôles. Le système permettra également de réveiller ou d'endormir les joueurs. Il sera possible de voter pour choisir un joueur spécifique, avec un suivi du déroulement du jeu dans le chat. Le système d'affichage affichera les pseudos des joueurs ainsi que leur carte pendant le jeu. Enfin, la gestion des morts sera intégrée pour assurer la fluidité du jeu.

**La version finale** : La version finale du projet inclura une série de fonctionnalités supplémentaires pour améliorer l'expérience de jeu. Les utilisateurs pourront choisir et modifier leur pseudo. Un système de temps limité sera mis en place pour les actions et les débats. Les statistiques des joueurs seront également disponibles ainsi qu'un classement. Il sera possible de jouer à plusieurs parties de jeu, avec la possibilité de choisir les paramètres de jeu. De nouveaux rôles seront ajoutés pour une expérience de jeu plus variée. Un chat vocal sera intégré pour une communication plus immersive (Optionnelle). La gestion des déconnexions sera améliorée pour assurer la fluidité du jeu. Enfin, une option pour une partie privée avec un mot de passe sera disponible pour les utilisateurs souhaitant jouer en toute confidentialité.

## 5.3 Le diagramme de Gantt

Vous trouverez ci-joint le diagramme de Gantt, qui présente une estimation approximative de la durée de chaque tâche de notre projet. Ce diagramme est un outil efficace pour organiser les tâches et suivre l'avancement du projet dans le temps. Il permet de visualiser les tâches à réaliser, leur ordre d'exécution et les dépendances entre elles. Il est cependant important de garder à l'esprit que ces estimations peuvent être modifiées en fonction des événements imprévus ou des changements survenant dans la réalisation du projet.