

## Rapport de Projet

# Loup-Garou de Thiercelieux en C

par Joan & Khiem



Legrand Joan

Tran Dinh-Khiem

Année 2020-2021

## Introduction

Face au différent choix proposé nous avons décidé de réaliser au Loup Garou de Thiercelieux en C, étant un jeu que nous connaissions on a trouvé intéressant de le programmer. Etant un jeu complexe dans sa conception, en effet il repose sur la confiance des joueurs et le mensonge, ce sont des particularités qui rendent la réalisation du projet plus complexe au premier abord. Mais après quelques temps de réflexion nous avons compris que cela n'était pas insurmontable.

Nous avons réparti les tâches de cette manière, Khiem a travaillé sur les fonctions des rôles et quelques fonctions secondaires du programme. Joan lui a travaillé sur la majorité des fonctions principales au déroulement de la partie. Dans un premier temps nous travaillons chacun de notre côté avec des rendez-vous hebdomadaires pour avoir une avancée stable. Cependant une fois que nous avons bien avancé dans le programme nous devons travailler ensemble pour avoir une meilleure compréhension des parties de chacun.

## Bibliothèques Importantes

Nous avons décidé de travailler sur l'environnement de développement CodeBlocks .Dans notre programme nous avons utilisé plusieurs librairies que nous avons trouvé importantes dans notre vision du projet. Tout d'abord la librairie String.h, elle est essentielle au bon fonctionnement de notre programme, elle contient la fonction Strcmp qui permet une comparaison du code ASCII de deux ensembles de caractères est utilisé dans la majorité des fonctions des rôles. Ensuite les deux librairies principales sont SDL2.h et Windows.h, elles sont surtout importantes dans la présentation du programme. Windows.h permet les modifications de la console, le changement de couleur par exemple. La SDL est utilisé pour afficher l'ensemble des images des rôles dans la fenêtre SDL créée pour cela.

## Réponses Attendues au Cahier des charges :

L'Objectif premier était " d'implémenter toutes les règles du jeu de base. Elles doivent être complètement et parfaitement implémentées pour qu'on puisse jouer une partie de Loups-Garous du début à la fin, à plusieurs joueurs sur un même écran. "

Notre programme répond à cet objectif en permettant à des joueurs (de 6 à 19), de jouer à une partie de Loup Garou, avec des règles identiques au Loup Garou de Thiercelieux.

Ensuite l'un des objectifs secondaires était d'implémenter une ou plusieurs extensions au jeu. Nous avons donc ajouté les extensions "Village" et "Nouvelles Lune", et avons donc ajouté aux 8 rôles du jeu de base les 8 rôles de ses autres extensions. Ce qui propose une diversité des rôles extraordinaire ; tout en fonctionnant parfaitement.

Le dernier objectif était la création d'une IA pour les Loups-Garous. Nous n'avons pas souhaité faire cet objectif secondaire, par manque de temps pour peaufiner notre travail, et car nous ne voyons pas l'intérêt de cet IA. Les vrais joueurs sauraient tous de suite pour qui voter (=> pour l'ordinateur), rendant la partie sans intérêt.

## Algorithmes et démarche de réflexion

Pour commencer nous avons créé une base solide pour le jeu. Nous avons donc créé la structure Joueur, comprenant les informations de chaque joueur (son nom, le nom de son rôle, son nombre de vie (0 ou 1 [ou 2 pour le cas de l'Ancien]), ainsi que des booléens indiquant les particularités offertes par des Fonctions d'autre rôle (couple, charmed, protected...).

Ensuite un Tableau Dynamique de structure joueur est créé en fonction du nombre de joueur. Il a fallu l'initialiser, c'est le rôle de la fonction CreationJoueurs. Elle va notamment répartir les joueurs aléatoirement à chaque rôle (rôles prédéfinis en fonction de l'extension choisie et du nombre de joueurs). Ainsi la variable rolejoueur de chaque joueur est initialisée caractère par caractère, pour éviter tout bugs ; et cela car par la suite tout le code sera basé sur la fonction strcmp qui va comparer cette variable pour détecter le rôle du joueur.

Après cela la partie peut commencer. La fonction DeroulePartie démarre et va, en premier lieu, appeler la fonction ConnaissanceRoles, qui permet aux joueurs de découvrir leur rôle.

La fonction DeroulePartie contient ensuite la boucle des tours, qui va appeler successivement les Fonctions des rôles (chaque rôle actif possédant sa propre fonction agissant sur le Tableau de Joueurs), et les fonctions de la journée (fonctions d’Affichage des joueurs charmés et des joueurs morts pendant la nuit, et notamment la fonction Vote)

Le programme est structuré de manière simple. Pour commencer nous n'avons pas de fichier header, toutes les fonctions se trouvent dans le fichier main.c. C'est quelque chose qui n'est généralement pas fait ; mais grâce à l'environnement CodeBlocs et à une bonne organisation des fonctions cela n'a posé aucun problème.

On pourrait noter que notre programme ne possède aucune variables globale (excepté le cas du Tableau dynamique de Joueur et de la fenêtre SDL [fenêtre + texture + rendu], qui pourrait être considérées comme globales, même si pas créées comme telles. Et toutes les informations dont toutes les fonctions ont besoin, sont systématiquement en paramètre.

## Avantage et limites du programme :

Notre programme répond au cahier des charges ce qui est le plus important. Notre interface graphique permet une meilleure compréhension du jeu lors de la partie. Les changements de couleur en fonction des rôles et les images correspondants à chaque situation lors de la partie sont un confort de jeu. La possibilité de jouer sans narrateur était une fonctionnalité importante pour nous car cela rendait le jeu plus attrayant et surtout plus intéressant à jouer, tout le monde est dans la partie et aucune personne n'est laissée à part. Notre programme consomme peu d'espace mémoire et donc par la même occasion est rapide.

Dans les limites de notre programme il y a sûrement des bugs dû à la grande quantité de parties et de possibilités dans le jeu. En effet nous avons réussi à déboguer des cas qui étaient arrivés lors de nos tests cependant il reste une certaine quantité de bug possible dans notre programme. Ensuite nous avons aussi la fenêtre SDL où il ne faut pas cliquer sinon celle-ci s'arrête et ne fonctionne plus. Le changement d'image est donc impossible, nous avons essayé de prévenir ce souci en affichant sur les images le texte "ne pas cliquer sur la fenêtre".

## Exemple déroulé :

Nous avons maintenant vous montrer le déroulé d'une partie type, nous avons choisis de faire une partie courte avec 6 joueurs et sans extension car cela est plus représentatif pour une présentation comme celle-ci.

Tout d'abord au lancement de notre programme nous demandons toujours aux joueurs présents s'ils veulent connaître les règles et conditions de victoire de chaque rôle.

```
Loup Garou | Joan & Khiem |
Voulez vous connaître avant de commencer, les specificites de chaque roles ?
entrez la reponse correspondantes :
0: Non
1: Oui
=>
```

Ensuite on demande l'extension choisi, ceci est important en effet certains rôles ne sont pas présents dans certaines extensions. Par la même occasion on demande le nombre de joueur pour pouvoir commencer à jouer.

```
Loup Garou | Joan & Khiem |
Entrez l'extension de votre choix (qui impactera les roles de la partie)
0: Loup Garou Classique (de 6 a 16 joueurs)
1: Extension Village (de 8 a 16 joueurs)
2: Extension Nouvelle Lune (de 10 a 17 joueur)
3: Extensions Nouvelles Lunes + Village (de 12 a 19 joueurs)
=>1
entrez le nombre de joueurs entre 8 et 16
=>
```

```
Loup Garou | Joan & Khiem |
entrez nom du joueur 1
=>Jean
entrez nom du joueur 2
=>Marc
entrez nom du joueur 3
=>Joan
entrez nom du joueur 4
=>Paul
entrez nom du joueur 5
=>Patrick
entrez nom du joueur 6
=>Maxime
```

On demande ici les noms de chaque joueur un par un. Ceci est une priorité pour le bon déroulement du jeu. Chaque joueur aura donc un numéro dans le tableau et sa structure joueur va avoir les changements nécessaires pour le début de la partie. C'est-à-dire l'attribution des rôles, le nombre de vie qui est mis à 1 (Sauf s'ils ont choisi une extension contenant l'ancien).

```

Loup Garou | Joan & Khiem |

Joueur 1 => nom : Jean
vote : 0
maire : 0
nombredevie : 1
Mechant : 0
couple : 0
nom role : LoupGarou
charmed : 0
protected : 0

Joueur 2 => nom : Marc
vote : 0
maire : 0
nombredevie : 1
Mechant : 0
couple : 0
nom role : Cupidon
charmed : 0
protected : 0

```

Ce tableau est un outil de travail que nous utilisons lors de nos tests. Cela nous permettait de savoir les changements au niveau de la structure de chaque joueur, son nombre de vie, ses statuts comme charmé ou s'il est en couple.

```

Loup Garou | Joan & Khiem |

Cupidon se reveille
Qui est la premiere personne amoureuse :

```

Lors du premier tour Cupidon va jouer, et mettre deux personnes en couple. Cette fonction n'est qu'utilisée qu'une fois par partie lors du premier tour. Ensuite la Voyante

```

Loup Garou | Joan & Khiem |

La voyante se reveille
Decouvrir le role de quel joueur ?
Jean
Jean a comme role : LoupGarou
. . 7

```

La voyante va pouvoir prendre une personne et voir son rôle. Pour cela on parcourt le tableau jusqu'à trouver le nom du joueur puis on prend grâce à la structure son rôle.

Ici pour montrer la fonctionnalité des noms incorrects nous avons choisi de mettre un nom qui n'était pas présent dans la partie. En effet si le nom de la personne choisi par les LoupGarous est une personne morte ou bien une personne qui n'est pas présente dans la partie. On dit alors au joueur qu'il est impossible de rentrer ce nom.

```

Loup Garou | Joan & Khiem |

Les loups garous se reveillent
Qui avez vous choisis de tuer?
=>Michel
Nom incorrect ou joueur protege ou mort, veuillez reessayer
Qui avez vous choisis de tuer?
=>

```

La sorcière possède une structure, elle permet de savoir si elle a toujours ses potions de vie ou de mort. Lors d'une utilisation, le booléen va passer en false.

```
Loup Garou | Joan & Khiem |
La sorciere se reveille
Paul est mort
Tu as encore du poison veux tu tuer quelqu'un ?
(si tu dis oui, tu ne pourras pas utiliser ta potion de ressurrection)
Ecrivez Oui ou Non
=>
```

Après l'affichage des morts le chasseur qui avait été tué va donc pouvoir jouer. Il a choisi Marc qui est déjà mort donc il doit rechoisir une autre personne ici Maxime.

```
Loup Garou | Joan & Khiem |
Le chasseur est mort il va donc se venge
Qui veux tu tuer?
Marc
Nom incorrect veuillez reesayer
Qui veux tu tuer?
Maxime
```

Pour élire le maire nous avons opté pour une solution simple les joueurs débâtent dans la vraie vie puis rentrent le nom du maire.

```
Sélection Loup Garou | Joan & Khiem |
Vous allez elire le maire,
procedez a un vote puis rentrez le nom du maire choisi
=>Patrick
Patrick est desormais le maire
. . 7
. . 6
. . 5
. .
```

Les votes du village se font chaque jour, il ne restait que deux joueurs donc Jean et Patrick, ils votent donc l'un pour l'autre. Sauf que Patrick est le maire son vote compte donc double.

```
Loup Garou | Joan & Khiem |
Vous allez voter pour tuer quelqu'un. Au prealable discutez et faites un debat
Jean, a ton tour de voter
Pour qui veux tu voter
=>Patrick
Patrick, a ton tour de voter
Pour qui veux tu voter
=>Jean
```

C'est donc Jean qui meurt lors du vote. La partie est donc finie avec la victoire de Patrick et du village avec lui car il ne reste aucun Loup Garou, ni rôle antagoniste.

```
Sélection Loup Garou | Joan & Khiem |
Jean est mort suite au vote du village il etait LoupGarou
. . 6
. . 5
. . 4
. .
```

```
Toutes les menaces du village ont ete elimines,
Victoire des villageois !
. . 5
. . 4
. . 3
. . 2
. . 1
. . 0
```

Voici des exemples d'image s'affichant dans la fenêtre SDL



## Conclusion

Grâce à ce projet ayant démarré il y a environ deux mois, nous avons tout d'abord dû apprendre à travailler en groupe. Ici un binôme nous avons choisi de nous mettre ensemble car nous travaillons déjà sur le TIPE ensemble. La grande peur à l'annonce de ce projet était la limite temporelle imposée qui nous semblait très courte cependant avec des efforts nous avons su travailler efficacement.

Nous avons aussi appris au niveau programmation, la SDL qui nous était jusqu'à lors inconnue après l'avoir comprise, elle est devenue un atout notable de notre programme. Apprendre à coder un programme entier avec des objectifs et attentes précis a été pour nous la notion la plus importante de ce projet. Pour nous ce projet était une réalisation qui nous a apporté notre premier contact à un travail assez conséquent et ce dans un temps réduit.