



LE JEU DU LOUP GAROU

Projet mené par Adriana GIROL, Nina DURIN et Lucas VALLADON

SOMMAIRE :

<u>I – Introduction</u>	p.3
<u>II – Présentation du projet</u>	p.3
<u>III – Réalisation du projet</u>	
A) Organisation du groupe	p.4
B) Structure globale	p.4
C) Explication du code de référence	p.5
D) Sur le plan sociétal et juridique	p.9
<u>IV – Conclusion</u>	p.10
<u>V – Annexes</u>	p.11

I – Introduction

Lors de l'année de terminale S au lycée Sainte Marie Bastide, mes camarades et moi-même avons eu l'occasion de choisir la spécialisation ISN qui a pour objectif éducatif l'apprentissage du codage et la réalisation d'un projet de fin d'année présenté au baccalauréat. En effet, notre groupe composé de GIROL Adriana, DURIN Nina et VALLADON Lucas a choisi de réaliser un site internet ainsi d'un jeu autour du célèbre jeu du « Loup Garou ». Nous nous sommes alors répartis les tâches suivantes :

- Gestion du site internet
- Gestion du code de référence
- Gestion de l'affichage

Notre projet est donc divisé en deux parties, l'une concernant le site internet et l'autre concernant le jeu vidéo.

II – Présentation du projet

Le loup garou est, à l'origine, un jeu de société dans lequel les joueurs se voient recevoir une carte qui les attribue à un groupe. Soit loup garou, soit villageois. Le but étant pour les villageois d'éliminer tous les loups garous pour remporter la partie, et vice versa. Rappelons qu'il existe plusieurs types de villageois tels que le simple villageois, la voyante, le chasseur, cupidon, la sorcière, le voleur, la petite fille et enfin le maire. Le but pour les loups garous est d'éliminer tous les villageois et le but pour les villageois est d'éliminer des loups garous, sachant que les joueurs ne connaissent pas les rôles des autres joueurs.

Nous avons décidé de mettre le jeu à disposition de n'importe quel joueur depuis un site internet. Ainsi nous avons eu recours non seulement à python pour coder le jeu mais aussi au html et au css afin de coder notre site internet.

III - La réalisation du projet

A) Organisation du groupe

A partir du moment où nous avons décidé ce que nous voulions réaliser, il a fallu se répartir les tâches. En effet, le codage en python demandant plus de volume de travail nous pensions que deux personnes étaient nécessaires à sa création afin de ne pas avoir de charge trop lourde. Lucas et Nina se sont donc divisé le travail en python. Dès lors Lucas s'est occupé du code de référence servant à la création du jeu alors que Nina elle, s'est occupée de l'affichage du jeu ainsi que sa mise en forme. Adriana s'est quant à elle occupée du site internet.

B) Structure globale

Le jeu du loup garou fonctionne en cycles. L'un le jour et l'autre la nuit. Dans un premier temps, il est nécessaire de donner aux différents joueurs leur rôle. Puis, le jeu initialise certains personnages possédant des aptitudes particulières :

- 1 – Le voleur (voir aptitudes en annexe) se réveille et choisit son rôle
- 2 – Cupidon (voir aptitudes en annexe) se réveille et choisit son rôle
- 3 – Les amoureux se réveillent et prennent connaissance de leur amant

Ensuite, le cycle jour / nuit peut débuter. La nuit, les loups garous se concertent et votent pour un joueur à éliminer et le jour, tous les joueurs votent pour éliminer un joueur soupçonnée d'être loup garou.

C) Explication du code de référence

Le code de référence qui va être expliqué est la version de travail qui nous a permis de réaliser le programme final. Ce code de référence a été codé par moi-même puis utilisé et simplifié par Nina afin de créer un affichage plus léger et de qualité. Ainsi ce code n'est pas celui de notre programme final mais celui que j'ai rédigé afin d'aboutir à ce projet. Cette version étant une version dite de « test », elle ne réalise qu'un affichage en lignes de commandes. Par ailleurs, il se peut que cette version, quelques fois, comporte quelques erreurs aléatoires dues à la densité de son code.

Dans un premier temps il est nécessaire d'écrire la formule `#-*-coding:utf8 -*-` afin d'éviter les caractères spéciaux. Ensuite, il faut importer divers modules car ces derniers sont utiles dans l'utilisation de fonctions appartenant à leur banque ou alors à l'utilisation de packages. Ainsi on importe les modules suivants grâce aux commandes :

```
from classes.locals import contenu
import random
import os
from classes.Roles import Roles
import time
```

Les modules `classes.locals` et `classes.Roles` sont des packages contenant les textes du jeu.

- La banque de fonctions « random » permet d'utiliser les fonctions :
 - `random.sample()` qui permet dans une liste de choisir un ou plusieurs éléments au hasard sans remise
 - `random.choice()` qui permet de choisir au hasard un élément dans une liste
 - `random.randrange(n)` qui permet d'obtenir un nombre entier aléatoire entre 0 et n inclus.
- La banque de fonctions « os » permet d'utiliser ici la fonction :
 - `os.system(« Pause »)` qui permet simplement un arrêt du programme
- La banque de fonctions « time » permet d'utiliser la fonction :
 - `time.sleep(n)` qui permet une pause de n secondes

Le menu :

La seconde étape est de créer le menu. Pour cela, on crée une **class** « menu » dans laquelle on demande au joueur s'il souhaite voir les règles du jeu. Pour cela on utilise une boucle **while** afin de faire choisir au joueur :

```
voir_regles1 = False
while voir_regles1 == False:
    voir_regles2 = input('Voulez-vous voir les règles du jeu ? \n')
    try:
        voir_regles2 = voir_regles2.upper()
    except TypeError:
        print('Vérifiez que vous entrez une lettre. \n')
    else:
        if voir_regles2.upper() != "O" and voir_regles2.upper() != "N":
            print("Vous devez entrer un 'O' ou un 'N', pour Oui ou Non. \n")
        voir_regles1 = True
```

Une seconde boucle **while** vient ensuite afin de demander au joueur s'il souhaite commencer une partie. Cette seconde boucle fonctionne sous le même principe que la première visant à afficher les règles.

L'initialisation du jeu :

L'initialisation du jeu donne dans un premier temps, grâce à un **def** de la fonction **define_role()**, un rôle aléatoirement au joueur. Cette fonction comporte une liste avec le nombre total de rôles et une gestion de différentes listes permettant un aléatoire :

```
def define_role():

    liste_roles = [ ]

    for i in range(0, 80):
        listeStart = [ ]
        listeEnd = [ ]
        alea = random.randrange(1, 18, 1)
        listeStart = liste_roles[:alea]
        listeEnd = liste_roles[(alea+1):18]
        replace = liste_roles[alea]
        liste_roles[:] = [ ]
        liste_roles.append(replace)
        liste_roles += listeStart
        liste_roles += listeEnd
    pass
```

Le jeu :

La partie jeu, elle, repose sur une `class` « jeu » dans laquelle on fait tourner en boucle deux parties tant qu'aucun des deux camps n'a gagné : une partie correspondant à la nuit et une autre correspondant au jour.

La nuit :

Pendant la nuit, les loups garous ainsi que la sorcière entrent en scène. Dans un premier temps, on affiche les textes correspondants aux loups garous :

```
print(Roles.Loup_garou.Texte_loupgarou[0])
time.sleep(1)
print(Roles.Loup_garou.Texte_loupgarou[1])
```

Rappelons que l'on va chercher les phrases de texte des divers personnages dans le fichier « Roles ».

Ensuite, il faut prendre en compte le fait que le loup garou soit un Bot ou alors qu'il soit un humain. Si ce dernier est un bot, alors sur la liste des 18 joueurs, il est entre 2 et 18 alors que s'il est un humain il est le numéro 1.

Pour faire cette distinction on utilise une boucle `for` :

```
for Bot_loupgarou in Joueurs[-17:]:
    if Bot_loupgarou is "loup garou":
        joueur_a_tuer1 = random.choice(Victimes_loupgarou)
        joueur_a_tuer2 = Joueurs.index(joueur_a_tuer1)
        return joueur_a_tuer2

for Joueur_loupgarou in Joueurs[0:1]:
    if Joueur_loupgarou is "loup garou":
        joueur_a_tuer2 = input("Pour qui votez-vous de tuer ?" + str(Joueurs[:]))
        print("Vous votez pour", joueur_a_tuer2)
        return joueur_a_tuer2
```

On utilise le même principe afin de faire jouer la sorcière dans le cas où elle serait un bot ou si c'est le joueur qui a ce rôle.

Le jour :

Pendant le jour, tous les joueurs (sauf ceux qui sont morts) votent pour un joueur qu'ils trouvent suspect et qui pourrait être loup garou.

Afin de procéder au vote des bots et du joueur, on utilise toujours le même procédé qui est celui d'utiliser une boucle **for** afin de parcourir les éléments de la liste des joueurs.

Dans le cas des bots on effectue un vote aléatoire parmi la liste des joueurs (sauf eux) qui repose sur le principe suivant :

```
for elt_bot in Joueurs[1:]:  
  
    if elt_bot is 'loup garou':  
        vote_loupgarou1_1 = random.choice(Liste_noms)  
        vote_loupgarou1_2 = Liste_noms.index(vote_loupgarou1_1)  
        Somme_vote.append(vote_loupgarou1_1)
```

Ici on peut voir que le bot choisit au hasard dans la liste des noms un nom, puis, il cherche l'indice de ce nom qu'il reporte dans la liste des joueurs. Une fois le joueur choisit, on ajoute le nom de ce joueur à une liste qui est la liste « Somme_vote » grâce à la fonction **append()**.

Cette opération est répétée pour chaque rôles à une différence près, celle du maire pour qui les votes comptent *5 et donc on ajoute 5 fois à la liste des votes le nom du joueur choisit.

De la même manière qu'avec les bots, le joueur lui, choisit le joueur pour lequel il veut voter :

```
for elt_joueur in Joueurs[0:1]:  
  
    if elt_joueur is 'loup garou':  
        VOTE_LOUPGAROU2 = True  
        while VOTE_LOUPGAROU2 is True:  
            vote_loupgarou2_1 = input("Pour quel joueur voulez-vous voter ?" + str(Liste_noms) + "\n")  
            vote_loupgarou2_2 = Liste_noms.index(vote_loupgarou2_1)  
            if Joueurs[vote_loupgarou2_2] is 'MORT':  
                print("Ce joueur est mort, veuillez en choisir un autre !")  
            else:  
                print("Vous votez pour : ", vote_loupgarou2_1)  
                VOTE_LOUPGAROU2 = False  
        Somme_vote.append(vote_loupgarou2_1)
```


La différence avec le joueur est que ce dernier ne doit absolument pas voter pour un autre joueur qui est mort, c'est pour cela que lorsqu'un joueur meurt, on remplace son nom dans la liste par « MORT » afin de vérifier si ce joueur est mort. Ainsi, si ce joueur est mort, le programme demande au joueur de voter pour un autre joueur qui lui n'est pas mort.

Cette opération est effectuée pour chaque rôle et comme pour les bots, le vote du maire compte pour 5.

D) Sur le plan sociétal et juridique

Le jeu « Les Loups-Garous de Thiercelieux » a été créé par Philippe des Pallières et Hervé Marly en 2001 et édité par **Lui-même** une société éditrice de jeux de société. Ce jeu, bien qu'il soit aisé de le reproduire avec de simples petits bouts de papier, est distribué en France par la société **Asmodée**. En cela, reproduire à l'identique ce jeu et s'en proclamer créateur est interdit. Ainsi, nous n'avons pas voulu créer un jeu identique à l'original car certains aspects du jeu tels que le fait de garder en mémoire des événements ou encore les nombreuses extensions, sont compliquées à créer. De plus, puisque ce jeu ne nous appartient pas, nous n'avons pas la permission de le recréer car ce dernier est défendu par la propriété intellectuelle. En effet, ce type d'œuvre semble correspondre au régime de l'œuvre collective définie par l'article L. 113-2 alinéa 3 du code de la propriété intellectuelle. Dans le cadre du jeu de société, l'auteur serait alors considéré comme un chef d'orchestre dans son domaine.

C'est pourquoi nous tenons particulièrement à créditer les réels détenteurs des droits de ce jeu.

IV – Conclusion

Ainsi, notre programme final fonctionne sans problèmes majeurs et nous avons déjà réfléchi à diverses améliorations possibles :

- Création de bots qui réfléchissent, s'accusent entre eux et se souviennent des accusations.
- Ajout des diverses extensions du jeu original.
- Un mode Multi-Joueurs
- Une possibilité de personnaliser sa partie en choisissant le nombre de joueurs, de rôles distribués etc...

Je suis personnellement très satisfait d'avoir choisi la spécialisation ISN car ce choix m'a permis de découvrir quelque chose de nouveau pour moi : la programmation (en langage python en l'occurrence). Cette année, ayant surtout appris les bases de la programmation informatique, il serait très intéressant pour moi de voir encore plus loin et m'entraîner à l'affichage notamment grâce au module PyGame.

J'ai beaucoup apprécié travailler avec mes deux autres camarades GIROL Adriana et DURIN Nina car nous avons su se répartir efficacement le travail afin de mêler un apprentissage du langage python (pour moi) rapide, avec la gestion d'un projet à réaliser en groupe. Pour cela il a fallu respecter des objectifs que nous nous sommes nous-même fixés et que nous avons pu suivre en partageant nos avancées sur GitHub.



V – Annexes

Règles du jeu :

- Le principe est simple :
 - Les Loups Garous doivent tuer ceux qui n'en sont pas.
 - Ceux qui ne sont pas des Loups Garous doivent les tuer.

- Le déroulement d'une partie :
 - Une partie se déroule sur plusieurs "jours". En effet, la nuit, les Loups Garous assassinent un villageois alors que le jour, les villageois se concertent afin de découvrir l'identité des Loups Garous.

- Les personnages :
 - Les Loups Garous :
 - Chaque nuit, ils dévorent un villageois. Le jour, ils tentent de masquer leur identité afin que les villageois ne découvrent pas qui ils sont.
 - Les villageois (ceux qui ne sont pas Loup Garou) :
 - Chaque nuit, l'un d'entre eux est dévoré par les Loups Garous et est éliminé. Le lendemain, ils se concertent afin de trouver des signes trahissant l'identité d'un coupable. Ils décident ensuite de la personne qui sera exécutée.

- Les différents types de villageois :
 - 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 découvre la vraie personnalité d'un joueur de son choix. Elle doit aider les autres villageois, mais rester discrète pour ne pas être démasquée par les Loups Garous.

- Le chasseur :
 - S'il se fait dévorer par les Loups Garous ou exécuter malencontreusement par les joueurs, le Chasseur a le pouvoir de répliquer avant de rendre l'âme, en éliminant immédiatement n'importe quel autre joueur de son choix.
- Cupidon :
 - En décochant ses célèbres flèches magiques, Cupidon a le pouvoir de rendre 2 personnes amoureuses à jamais. La première nuit, il désigne 2 joueurs comme étant amoureux. Si l'un des amoureux est éliminé, l'autre meurt de chagrin immédiatement. Un Amoureux ne doit jamais voter contre son bien-aimé.
- La sorcière :
 - Elle sait concocter 2 potions extrêmement puissantes: Une potion de guérison, pour ressusciter le joueur tué par les Loups Garous, une potion d'empoisonnement, utilisée la nuit pour éliminer un joueur.
 - La sorcière ne peut utiliser chaque potion qu'une seule fois dans la partie. Elle peut se servir de ses 2 potions dans la même nuit. Le matin suivant il pourra, suivant l'usage de ce pouvoir, y avoir 0 mort, 1 mort ou 2 morts. La sorcière peut utiliser les potions à son profit, et donc se guérir elle-même.
- La Petite Fille :
 - La Petite Fille peut, en entrouvrant les yeux, espionner les Loups Garous pendant leur réveil. Si elle se fait surprendre par un des Loups Garous, elle meurt immédiatement (en silence), à la place de la victime désignée. La Petite Fille ne peut espionner que la nuit, durant le tour d'éveil des Loups Garous.
- Le voleur :
 - Après distribution, les deux cartes non distribuées sont placées au centre de la table face cachée. La première nuit, le voleur pourra prendre connaissance de ces deux cartes, et échanger sa carte contre l'une des deux autres. Si ces deux cartes sont des Loups Garous, il est obligé d'échanger sa carte contre un des deux Loups Garous. Il jouera alors ce personnage jusqu'à la fin de la partie.
- Le maire :
 - Son rôle est primordial parmi les villageois. En effet, son vote vaut 5. Cependant, s'il meurt, tout le village à part la sorcière MEURT.