

ORACULAR LANDS

By **Dorian Laurancy**

A web application for managing sessions of my tabletop role-playing game
"Oracular Lands"

GitHub:

<https://github.com/Laurancy-Dorian/OracularLands>

Polytech Montpellier – 2018 / 2019

SUMMARY

Glossary	2
Introduction	3
I – Functionalities	4
1.1. Document saving	4
1.2. Preparing a game	4
1.3. Play a session	4
II – Project Design	5
2.1. Database	5
2.2. Architecture and technologies	6
2.2.1. Backend	6
2.2.2. Frontend	6
III – Problems	7
IV – Post mortem	7

GLOSSARY

Tabletop role-playing game (RPG): A form of board game where the participants are divided in two groups, the game master (GM) and the players. During a game, the game master tells a story and exposes a situation in which the main characters are embodied by the players. The GM describes the game world and its inhabitants while the other players describe the intended actions of their characters. Some outcomes are determined by the game system, through dice or rules, and some are chosen by the GM. All of these are defined actions through speech. Participants determine the actions of their characters based on their characterization, and the actions succeed or fail according to a set formal system of rules and guidelines. Within the rules, players have the freedom to improvise, and their choices shape the direction of the scenario. The success or failure of an action are often determined by dice roll.

Session: Period during which a game takes place

Story arc: Extended or continuing storyline that gradually unfolds over several episodes (or scenarios in the case of RPG). We can compare it to seasons in TV shows.

Scenario: One episode of a story arc. It is the frame that will serve the development of the story. The game master writes it before starting a session. In TV shows, it would be an episode.

Situation: An element of the plot. It expose an event that happens in the game: dialogue, fight, scenery, etc. In a TV show, a situation would be an event in the episode.

Character sheet: Record of a player character in a role-playing game, including whatever details, notes, game statistics, and background information a player would need during a play session. Character sheets can be found in use in both traditional and action role-playing games. Almost all roleplaying games make use of character sheets in some fashion; even "rules-light" systems and freeform role-playing games record character details in some manner.

INTRODUCTION

Since I was 12 years old, I have been playing tabletop role-playing games (RPG), and it has become one of my favorite hobbies.

A tabletop role-playing game, or RPG, is a social activity where people play through a story or adventure experience under a certain set of rules set by another person. This other person is called the Game Master and its job is to make sure that the rules are followed and the game's story is flowing smoothly. The players embody the main characters, and together everyone helps telling the story and experience the adventure ([see the full definition in glossary](#)).

After participating as a player for several years, I decided to manage some sessions and became Game Master. Eventually, I ended up creating my own RPG. During my sessions, I like launching music and showing images and documents to my players, but I happen to get lost in my files, breaking the rhythm of the game. I then had an idea to make a tool that could help me managing my sessions and I decided to use the web project to make it.

I – FUNCTIONALITIES

This project is a web application that will help me and other game masters managing a tabletop role-playing game. All the functionalities are described in [use case scenarios in annex 1](#) and [2](#).

The main features that the app will offer are preparing games, saving documents and showing them during a session, and playing a session.

1.1. Document saving

Game masters can save documents that are images, videos, music or text. They will be used in sessions and showed to the player to help them imagine the event that is occurring or add a new element. For instance, during a session, the game master can launch a music, show an image to the player and give them a text that brings a new element to the plot.

1.2. Preparing a game

The game master prepares a session by defining story arcs, scenarios and situations ([see glossary](#)). A story arc contains a set of scenarios, which contain a graph of situations. Each situation can have previous situations and next situations. A situation contains a text that describes the event. A story arc has a set of documents attached to it that will be displayed in the session.

When they created everything, he creates a new session and attaches it to the story arc. A password is displayed and he has to give it to his players so that they can subscribe to this session and play.

1.3. Play a session

During a session, the users are connected between each other. The behavior of the app depends of the type of user.

If the user is the Game Master, then a lot of details are displayed:

- The script or description of the current situation
- A graph of all the situations in the scenario, that are the different outcomes and events the GM planned. The user can go from a situation to another with this graph.
- A file tree that contains all the documents related to the scenario. The user can choose to show them to the players (“show” means display the document is an image, start if it is a music...)
- A chat

If the user is a player, the only things that are displayed are the documents that they is authorized to see, and the chat.

II – PROJECT DESIGN

2.1. Database

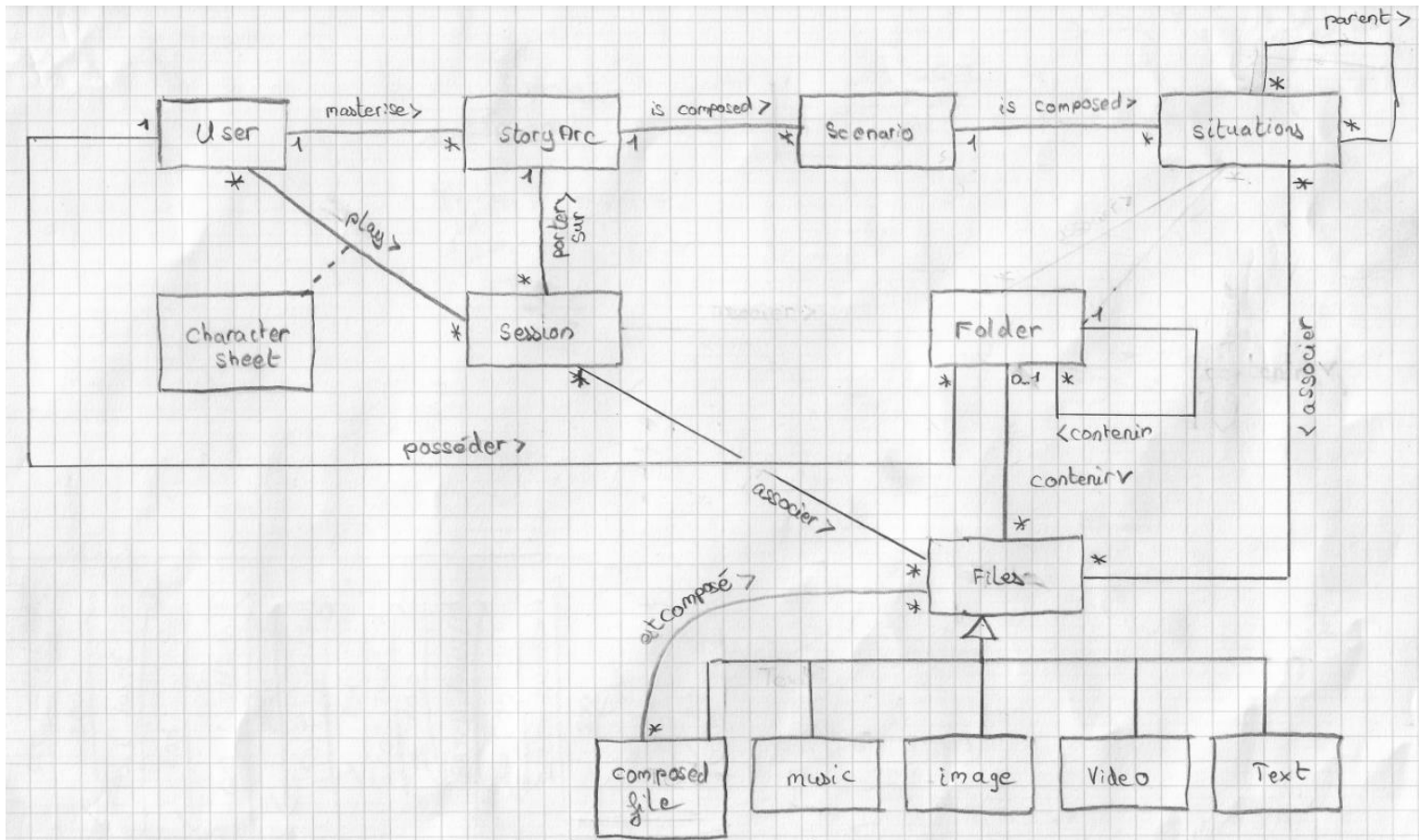


Figure 1 - Database Class Diagram

In my database, we can see that the folders and the files are linked, because a folder is composed of files. Nevertheless, there were some problems with deletions. Therefore, I implemented a trigger to remove all the files and folders if we delete their parent folder.

It is possible that a composed file is composed with anonymous files OR other files that already were there. In the first case, we need to delete the subfiles if we delete the composed files. In the second case, we must not delete it. My second trigger concerns these cases.

2.2. Architecture and technologies

2.2.1. Backend

I decided to run my app on nodeJS with the famous ExpressJS framework. I chose this technology for its performance and scalability. Indeed, even if it is single-threaded, it has also an event-driven architecture that permit handling multiple simultaneous connections efficiently. Beside, since it is single-threaded, it does not have to allocate more RAM than it needs creating a new thread each time a client connects. It uses of the event loop and callbacks for I/O operations, delegating tasks such as database operations as soon as possible. This allows handling a lot of connections at the same time.

2.2.2. Frontend

I used vue.js for frontend management because it is a stable and progressive framework and I find it easy to use once I learned how to use it. With vue.js, we can build simple and reactive web apps. Finally, I wanted to try to use a frontend framework, because I had always used native javascript or JQuery, and vue.js seemed to be easy enough to learn.

I also used Bootstrap as a css framework its classes allows us to design pretty and responsive web pages very quickly.

III – PROBLEMS

The main problem I faced is the fact that the project is very large. Thus, I lost a lot of time designing my application. My first database diagram had about 30 tables, because I wanted to include all the rules of my RPG. I knew it was too much so I kept redesigning the app. Unfortunately, it was still too big so I was not able to finish my project as I wanted.

Even if though I knew javascript, it was the first time I used NodeJs and Vue.js, so I had to learn and check the documentation regularly and I could not be as fast as I would have been if I was coding in PHP. However, I do not regret my choices of technologies, I find them very effective and I'm glad I learned them.

Apart from that, I did not have any big problems, only some minor problems that I solved quickly in most cases.

IV – POST MORTEM

In this project, I learned many new technologies, acquired new skills and I am sure it will help me in the future.

Even if I could not finish this project, I think it's globally well designed and the features that are already implemented work properly. For now, one can register and create story arcs.

I intend to continue and finish my project. I think it will be a fantastic tool when it will be usable. I will be able to manage my RPG sessions efficiently and it will bring a new dimension to it. I am confident on the outcome of my application because I think I really designed it the way I wanted.

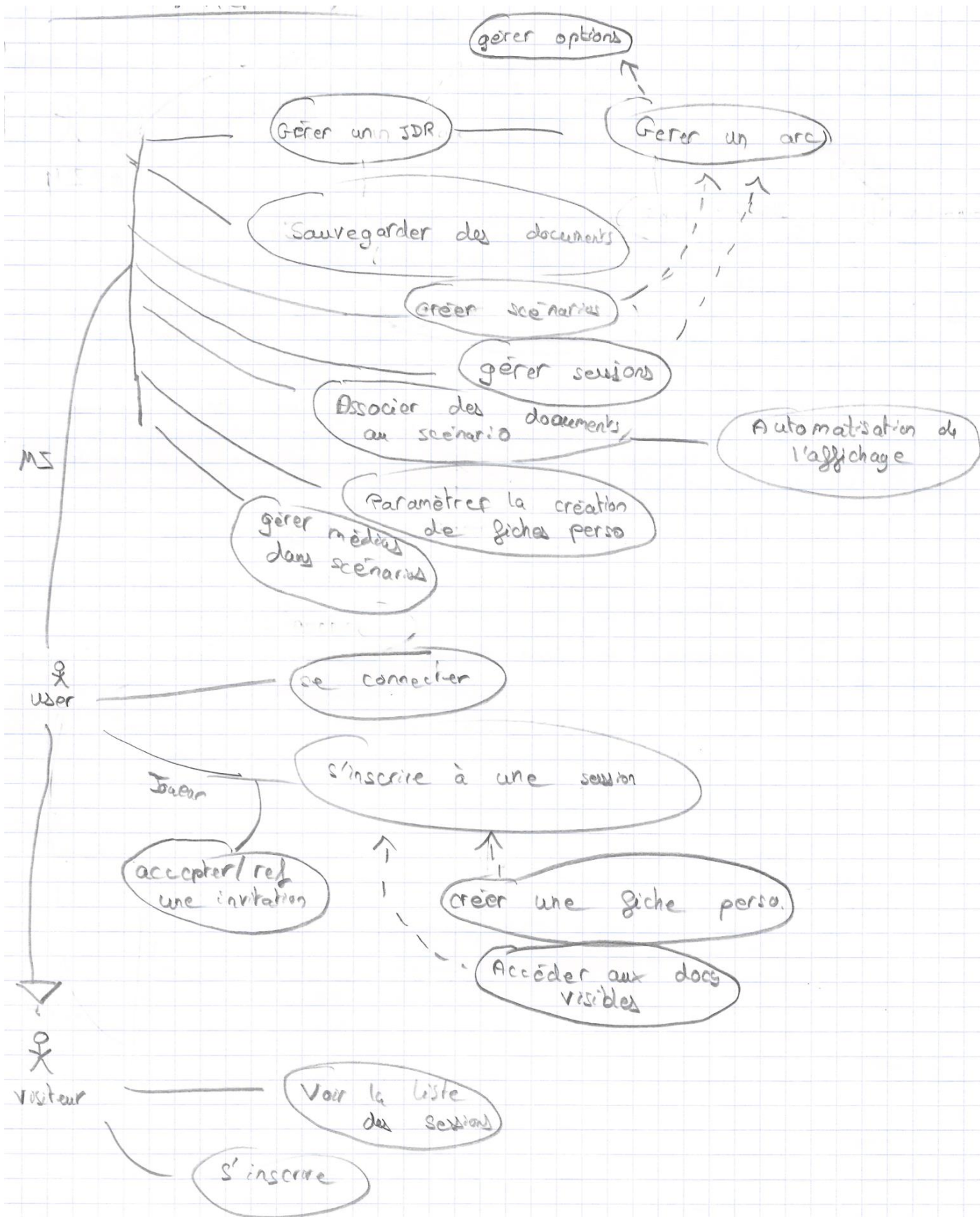
When I will be done, I will add the features I intended to make in the first place, such as all the rules of my game and try to make my app suitable for all tabletop role-playing games.

ANNEXES

Summary

Annex 1: Use case Diagram 1 – Preparing a game	II
Annex 2: Use case Diagram 2 – Play during a session	IV

ANNEX 1: USE CASE DIAGRAM 1 – PREPARING A GAME



1. Visiteur - Un visiteur est un utilisateur qui ne possède pas de compte ou qui n'est pas connecté.

S'inscrire - Le visiteur peut s'inscrire en renseignant son pseudo, son mot de passe, et son adresse e-mail. Il reçoit alors un mail pour confirmer son inscription en cliquant sur un lien, son compte est alors créé.

Se connecter - Quelqu'un qui possède un compte mais n'est pas connecté est un visiteur. Il peut renseigner son pseudo et son mot de passe pour s'authentifier. Après s'être connecté, il devient Utilisateur.

Voir la liste des sessions - Affiche la liste des sessions de jeu disponibles. Un visiteur ne peut pas s'inscrire à une session, mais il peut voir sa description et ses caractéristiques.

2. Utilisateur - Chaque utilisateur peut être à la fois joueur ET maître de jeu (MJ). Chaque utilisateur possède donc un espace personnel où il peut gérer son JDR, mais peut aussi participer à des séances proposées par d'autres utilisateurs.

Se déconnecter - Déconnecte l'utilisateur. Celui-ci devient alors Visiteur.

2.1. Côté Joueur

Voir la liste des sessions - Affiche la liste des sessions de jeu disponibles. Un utilisateur peut voir la description et les caractéristiques d'une session.

S'inscrire à une session - S'il possède le mot de passe de la session, le joueur peut s'y inscrire. Il fera partie du groupe de joueur pour cette session

Créer une fiche perso - Une fois inscrit à une session, l'utilisateur peut créer, afficher et modifier sa fiche perso.

Accéder aux documents visibles - Le joueur peut afficher les documents que le MJ a rendu visibles.

2.2. Côté Maître du Jeu

Gérer des documents - Un document peut être : Une Image, Une musique (Ne peut pas être visible, mais peut être « affichée »), du texte simple, une vidéo ou un document composé des éléments ci-avant.

Un dossier peut comporter des dossiers et / ou des documents. L'utilisateur peut créer, afficher, modifier, déplacer ou supprimer ces documents ou dossiers. Un document peut être visible ou caché. S'il est visible, les joueurs ont accès à ce document. Un document peut être affiché : selon son type, le document sera mis en évidence (musique qui se lance, image ou texte mis en premier plan ...)

Gérer les Arc Narratif - Un Arc est une histoire du JDR. Il porte sur le même groupe de personnages du début à la fin, dans le même contexte. Il est composé de plusieurs scénarios qui se suivent. Un Arc est le modèle d'une session (voir gérer sessions).

Gérer les options d'un Arc - Les Arcs peuvent avoir des différences entre eux : la création des fiches perso (stats, noms, classes, races...), les documents de base (règles, cartes...) qui sont visibles ou non, le nombre de joueurs.

Paramétrer la création des fiches persos dans un Arc - Une fiche perso décrit un personnage, ses stats, ses pv ... Paramétrer la fiche perso signifie paramétrer les races disponibles, les stats en fonction des races, les compétences disponibles, etc.

Gérer les scénarios - Un scénario est un chapitre d'un arc. Elle comprend les caractéristiques de l'arc, et peut modifier certaines propriétés des fiches perso (cas de level up par exemple). Un scénario est composé de situations.

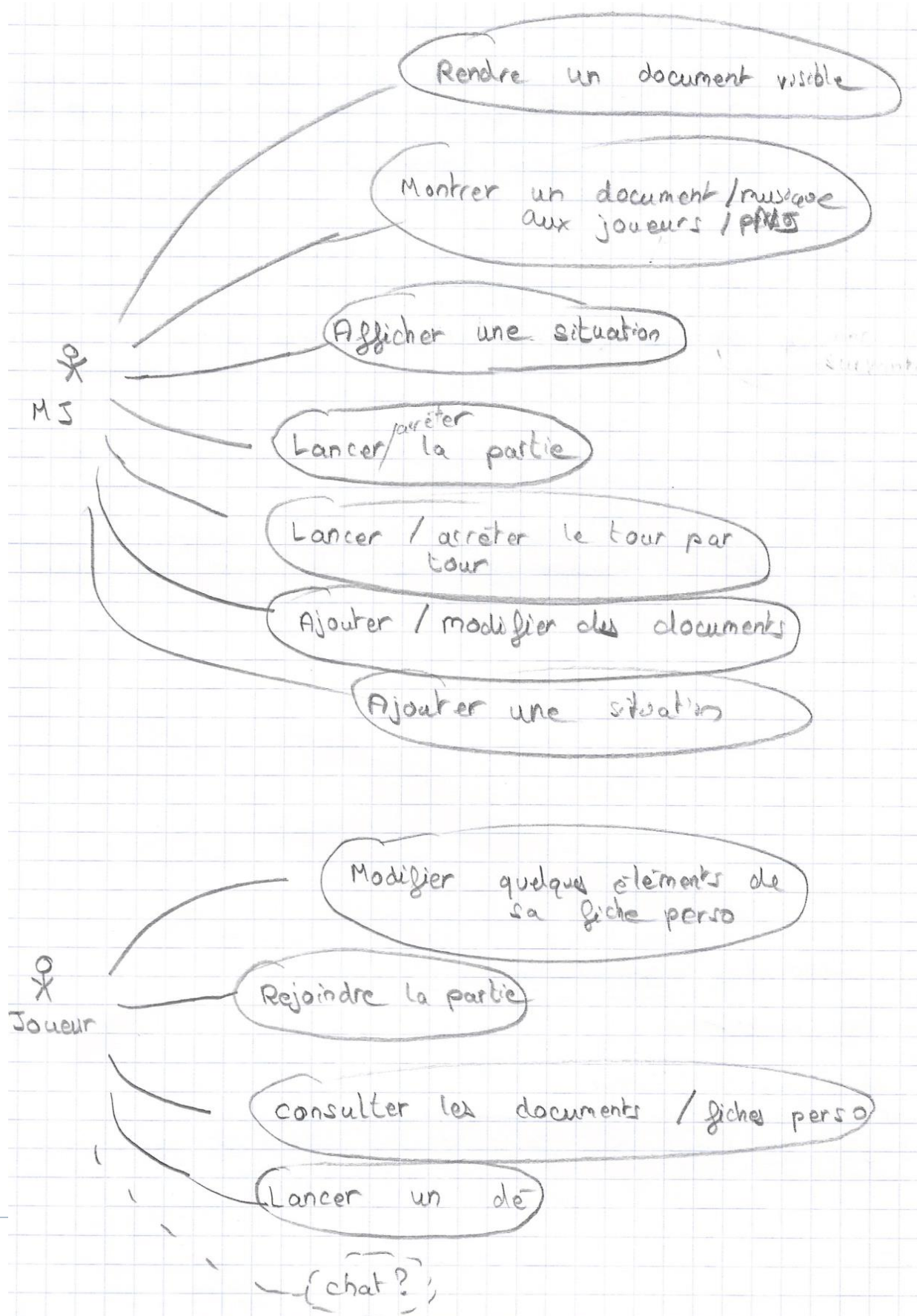
Gérer les situations - Une situation est un moment d'un scénario. Elle comprend les caractéristiques du scénario, et peut rendre visible ou cacher des documents. Elle peut définir les documents visibles, cachés et affichés de base (lors de l'entrée dans cette situation) La situation possède une description qui la décrit de manière exhaustive. Elle peut également définir de nouvelles règles d'affichage sur certains types de documents : Musique -> playlist, Image -> background

Gérer les sessions - Pour un Arc donné, le MJ peut faire jouer plusieurs groupes de joueurs. Ces groupes seront dans des situations différentes et peuvent donc avoir des documents visibles différents.

Une session est donc un Arc + un groupe de joueur. La session comporte toutes les caractéristiques de bases de l'arc : Les scénarios, les documents, le paramétrage. Cependant, ces caractéristiques sont susceptibles d'évoluer au fur et à mesure des parties : des documents se cacher ou être rendus visibles pour cette session seulement (l'Arc ne sera pas affecté).

La session est reliée à un arc et des joueurs, et possède la situation actuelle, ainsi que la liste des documents visibles dans cette session.

ANNEX 2: USE CASE DIAGRAM 2 – PLAY DURING A SESSION



1. Joueur + MJ

Lancer un dé - Lance un dé choisis par le joueur (d4, d6, d8, d12, d20, d100). Le résultat est affiché pour tout le monde.

2. Joueur

Modifier certains éléments de sa fiche perso - Le joueur peut modifier sa fiche perso pendant la partie (inventaire, niveau...). Les compétences et statistiques ne sont pas changeables par le joueur.

Consulter les documents – Les joueurs peuvent consulter les documents que le maitre du jeu a définis comme « visibles »

3. Maitre du jeu

Rendre un document visible - Rend un document visible pour les joueurs

Afficher un document - Affiche le document (voir Use case Preparation - Gérer Documents)

Changer de situation - Le MJ change de situation, les paramètres pour cette situation s'appliquent (voir Use case Preparation - Gérer Situations)

Changer de scénario - Le MJ peut changer de scénario

Lancer le tour par tour - Le MJ lance le tour par tour, les PNJ et le joueur qui jouera sera affiché.

Arrêter le tour par tour - Le MJ arrête le tour par tour

Lancer un dé caché - Lance un dé choisis par le joueur (d4, d6, d8, d12, d20, d100). Le résultat n'est affiché qu'au MJ

Modifier une fiche perso - Le Mj peut modifier n'importe quel élément d'une fiche perso

Ajouter ou modifier un document dans une situation - Ajoute ou modifie un document dans l'arc. Cela affecte toutes les sessions de cet arc

Modifier - ajouter un arc / scénario / situations - Le MJ peut ajouter ou modifier un arc, un scénario ou une situation à tous moments pendant une partie. Cela affectera TOUTES LES SESSIONS de l'arc