

# APG Helper

Un logiciel PC et une application mobile pour faciliter vos parties de jeux de rôle

Loïc LE COZ

Arthur EPAUD

Louis OLIVIER

Jean-Loup MONNIEA



Printemps 2018

---

## Amis rôlistes !

Vous qui aimez passer des soirées autour d'une table, que ce soit pour raconter des histoires ou pour les vivre, le Maître du Jeu (MJ) vous racontant son dernier scénario qu'il a préparé lui-même, vous ne cherchez qu'à passer une bonne soirée avec des amis, sans prise de tête. Néanmoins, les systèmes de règles de certains jeux de rôle peuvent se révéler touffus : qui peut se targuer de respecter toutes les règles à la lettre ? De bien respecter tous les effets et bonus temporaires ? De ne pas faire des erreurs de calcul lors de retrait ou gain de statistiques ? Les joueurs et le MJ doivent parfois jongler avec de nombreuses informations en continu et il est parfois compliqué de ne pas s'emmêler les pinceaux.

Notre projet, séparé en un logiciel PC pour le MJ et une application mobile pour tous les joueurs, permet d'automatiser certains procédés laborieux, de simplifier les recherches d'informations utiles, et de rassembler fonctionnalités utiles pour le MJ (telles que de la génération de monstres ou d'objets) afin de simplifier vos parties !

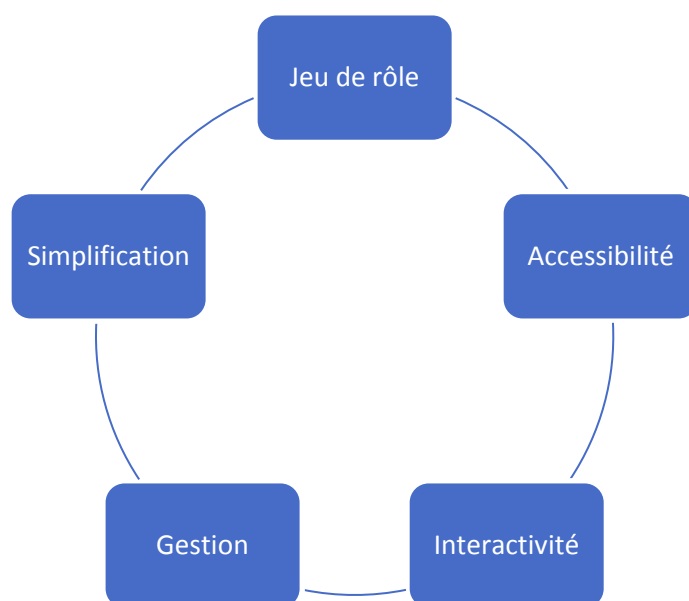
---

## Role players !

You who like to spend your evening around a table, telling stories or living them, the Dungeon Master (DM) narrating his last self-made scenario: you only want to have a good time with your friends. But sometimes, some RPG's rules can be unclear: who can brag about following all the rules to the letter? Respecting all temporary effects or bonuses? Calculating without mistakes every time? RPG players must deal with a lot of informations during a game and it's sometimes easy to get things mixed together.

Our project, (split into a PC software for the DM and a mobile application for the player), allows to computerize some laborious processes, simplify information researches, and gather functionalities useful for the DM (such as monster or objet randomization), to ease your game nights!

---



---

# Table des matières

Avant toute chose: Qu'est-ce qu'un Jeu de Rôle?.....	4
Être Maître du Jeu, c'est du boulot... ..	4
Un logiciel et une application à votre rescousse !.....	5
1 ~ Présentation globale du logiciel.....	5
2 ~ Présentation des fonctionnalités du logiciel .....	7
3 ~ Présentation de l'application.....	10
Avant de coder, bien modéliser.....	12
1 ~ Diagrammes de cas d'utilisation.....	12
2 ~ Scénarii.....	14
3 ~ Diagrammes de séquence .....	18
4 ~ Diagrammes de classe .....	28
5 ~ Diagrammes d'état-transition.....	32
6 ~ Diagrammes d'activité.....	34
Les outils de développement.....	36
Qui sommes-nous? .....	37
Gantt .....	38
1 ~ Prévisionnel P1.....	38
2 ~ Définitif P1 & Prévisionnel P2.....	39
Répartition des tâches.....	40
Conclusion.....	41



---

# Avant toute chose : Qu'est-ce qu'un Jeu de Rôle ?

Un jeu de rôle (JDR) est un jeu de société dans lequel les participants conçoivent ensemble une fiction par l'interprétation de personnages et par la narration, dans un univers précis, avec un ensemble de règles à respecter. N'importe quel univers est propice à une partie de jeu de rôle, mais par la suite nous ferons référence à un univers très classique d'heroic-fantasy : celui de Donjons et Dragons (D&D). Nous nous baserons sur son système de règles afin de définir les informations à prendre en compte lors d'une partie.

Lors d'une partie de JDR, il y a 2 rôles bien distincts : celui du Maître du Jeu (MJ) et celui des joueurs. Le rôle du MJ est d'imposer un univers, de raconter une histoire dans laquelle les joueurs sont les personnages centraux, et donc de décrire les conséquences qu'ont les actions des joueurs. Le MJ a également pour rôle d'incarner les personnages non joueurs (PNJ) et de décider de leurs actions. Ces PNJ peuvent être sympathiques, ou au contraire attaquer les joueurs. Les joueurs quant à eux sont libres de faire ce qu'ils veulent (avec pour seule limite celles imposées par le MJ) afin de répondre aux différentes péripéties qui s'offrent à eux. Les tours de paroles doivent être distribués par le MJ, afin que tout le monde puisse s'exprimer et expliciter ses actions. Les aventures sont souvent parsemées de combats, qui se résolvent alors au tour par tour.

Le MJ doit donc faire vivre une histoire à ses joueurs, et les actions de ces derniers seront décisives pour la résolution du scénario concocté par le maître du jeu !

Une partie de JDR est donc propice à l'imagination et aux fous rires, parfait pour s'amuser entre amis, et surtout pas pour se prendre la tête !

## Être Maître du Jeu, c'est du boulot...

Évidemment, la clé d'une partie réussie est la préparation en amont du MJ : il doit créer une histoire, des personnages, des situations, des combats... Le plus souvent, le MJ crée son univers sur des diverses notes, papiers ou traitements de texte, les différentes rencontres sur des tableurs... Chacun prépare ses parties à sa manière, sur les supports qu'il préfère. Il existe également des scénarios « pré-faits », disponibles dans certains JDR.

De plus, les jeux de rôles ont la réputation d'être « compliqués » et obscurs pour les non pratiquants, ce qui dans certains cas, n'est pas totalement faux...

En effet le MJ a le rôle d'arbitre du bon respect des règles, doit pouvoir jouer la comédie, faire en sorte que les joueurs s'amuse à suivre l'histoire, et improviser dans le cas contraire. Cela fait donc beaucoup d'informations à prendre en compte et il n'est pas rare de se retrouver submergé.

Les joueurs doivent aussi jongler avec de nombreuses variables : les points de vie (PV), les différentes caractéristiques, les compétences, les bonus et malus qui s'appliquent pendant un temps limité... Bien souvent, les joueurs débutants qui ne connaissent pas encore bien les règles sont perdus parmi tous ces chiffres et omettent de prendre en compte certaines informations, ce qui peut mener à un non-respect des règles handicapant.



# Un logiciel et une application à votre rescousse !

Le projet se décompose donc en 2 parties distinctes : un **logiciel PC pour le MJ** et une **application mobile pour les joueurs**.

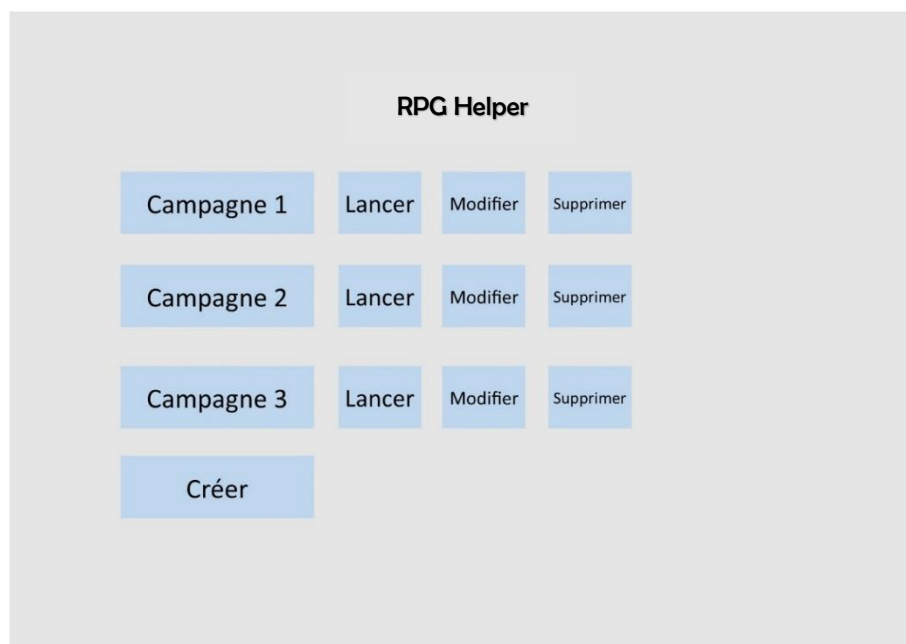
## 1 ~ Présentation globale du logiciel

Ce logiciel est à destination du MJ, il lui servira à créer sa campagne<sup>1</sup> en amont d'une partie et restera ouvert et disponible tout le long d'une partie en cours. En effet il est fréquent qu'un MJ garde un ordinateur à proximité afin de pouvoir vérifier des informations en cours de partie, ou tout simplement pour accéder à ses notes : l'utilisation du logiciel lors d'une partie ne sera donc pas gênante.

Lors d'une partie, le logiciel sera en communication constante avec les téléphones des joueurs : le MJ pourra alors modifier leur affichage afin d'afficher les informations importantes ou des images.

Hors partie, le logiciel sera disponible à tout moment et servira à préparer sa campagne : préparer l'histoire, les différentes rencontres, les monstres et personnages présents...

### 1.1 ~ Écran de démarrage



<sup>1</sup> Une campagne est une histoire à long terme ou une série d'aventures, qui impliquent généralement les mêmes personnages. Elle est préparée par le MJ et souvent découpée en plusieurs sessions de jeu.

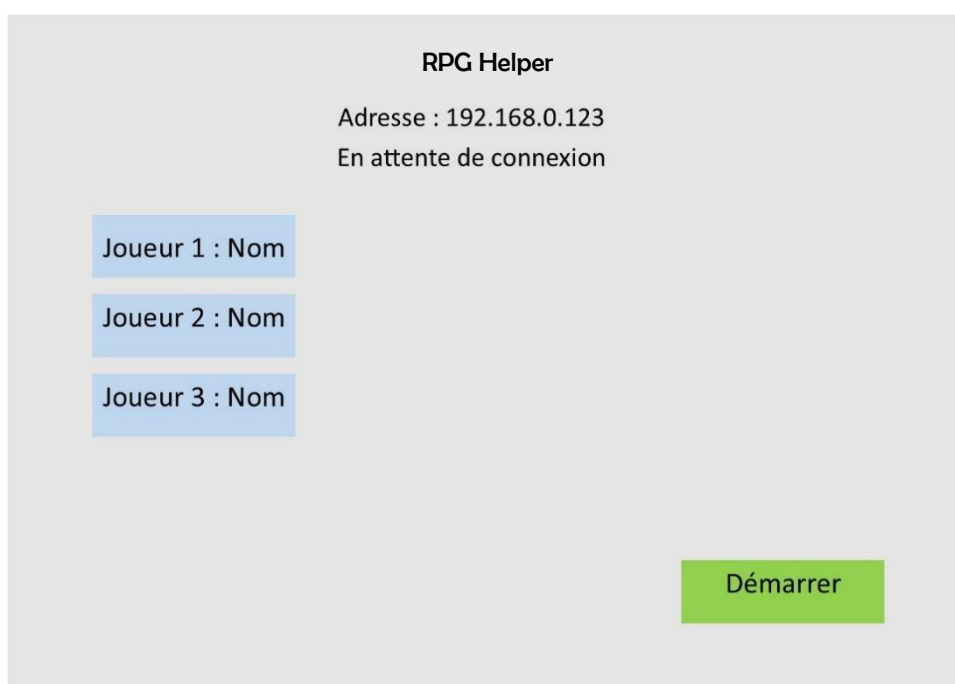


C'est cet écran qui sera présent au lancement du logiciel. Il permet au MJ de :

- ~ Créer une nouvelle campagne
- ~ Modifier/supprimer une campagne existante : la modification a lieu en amont d'une partie, lorsque le MJ prépare l'histoire qu'il va faire vivre à ses joueurs.
- ~ Lancer une campagne : cette option ne s'applique que lorsque tous les joueurs sont présents, prêts à jouer.

Cliquer sur **Lancer** fera apparaître l'écran de connexion suivant :

## 1.2 ~ Écran de connexion



C'est à ce moment que les joueurs pourront se connecter en rentrant l'adresse IP<sup>2</sup> présente sur le logiciel ainsi que leur nom et informations.

<sup>2</sup> Une adresse IP (avec IP pour Internet Protocol) est un numéro d'identification qui est attribué de façon permanente ou provisoire à chaque branchement à un réseau informatique utilisant l'Internet Protocol. Ici elle permettra aux téléphones de se connecter à l'ordinateur.



## 2 ~ Présentation des fonctionnalités du logiciel

Après avoir créé une campagne, il sera possible de modifier ses caractéristiques.

Chaque campagne est indépendante, et est décomposée en onglets, à savoir :

- ~ Histoire
- ~ Combat
- ~ Joueurs
- ~ Règles
- ~ Générateur

Nous allons nous attarder sur la description de chaque onglet.

### 2.1 ~ Onglet Combat

The screenshot displays the 'Combat' tab of a software interface. The interface is organized into several sections:

- Top Navigation Bar:** Contains tabs for 'Histoire', 'Combats', 'Règles', 'Générateurs', and 'Joueurs'. The 'Combats' tab is currently selected.
- Sub-Tab Bar:** Below the main tabs, there are sub-tabs labeled 'rencontre1', 'rencontre2', 'rencontre3', 'rencontre4', and a '+' button. The 'rencontre1' sub-tab is active.
- Combat Grid:** A table with 5 columns labeled 'Monstre 1' through 'Monstre 5' and a '+' button. The first row is labeled 'Stat 1', the second 'Stat 2', and the third 'Stat 3'.
- Monster Details Panel:** On the right side, there are two panels for 'Monstre 1' and 'Monstre 2'. Each panel contains a description of the monster and a small image. The 'Monstre 1' panel is currently selected.
- Player and Monster Selection Panel:** On the far right, there are buttons for 'Joueur 1', 'Monstre 1', 'Joueur 2', and 'Monstre 2'. A yellow dot is visible next to 'Monstre 1'. Below these is a '+' button and a 'Tour suivant' button.
- Bottom Bar:** Contains a 'Commande: ex 1d20' field, a 'Résultat' button, an 'Heure en jeu' field showing '12 h', and a 'Passer' button.

Numbered callouts (1-7) point to specific elements:

- 1: Points to the 'Combats' tab in the top navigation bar.
- 2: Points to the 'rencontre1' sub-tab.
- 3: Points to the 'Monstre 5' column header in the combat grid.
- 4: Points to the 'Joueur 1' button in the player selection panel.
- 5: Points to the 'Heure en jeu' field in the bottom bar.
- 6: Points to the 'Commande: ex 1d20' field in the bottom bar.
- 7: Points to the 'Monstre 1' description panel.



- 1- **Onglets principaux.** Ils seront toujours présents en haut de l'écran.
- 2- L'onglet Combats sera composé en sous onglets « **Rencontre** ». Le MJ peut créer autant de rencontres qu'il le souhaite et les modifier librement.
- 3- Chaque rencontre comporte un **tableau de statistiques des monstres**. Ce tableau permet d'ajouter des monstres ou PNJ (Personnages Non Joueurs : des personnages interprétés par le MJ, qui peuvent être alliés ou hostiles envers les joueurs). Ce tableau regroupe les statistiques les plus importantes des créatures (Point de vie (PV), Classe d'Armure (CA)...). Lors d'une partie réelle, il servira à modifier les statistiques des monstres lorsqu'ils sont touchés, et permet donc de savoir quand les ennemis sont hors combat.
- 4- **Tour de jeu : uniquement accessible en partie réelle.** Cette liste est à générer à chaque début de rencontre<sup>3</sup>. Les combats se déroulant au **tour par tour**, elle explicite quel est le joueur (ou l'ennemi) actif. Le joueur actif est signalé par une pastille. L'appui sur le bouton « Tour Suivant » change le personnage actif (du haut vers le bas). Lorsqu'on n'est pas en partie réelle, cette zone sera juste vide.
- 5- **Heure en jeu.** Cette information sera présente en bas de tous les onglets. En effet, les parties pouvant se dérouler sur plusieurs « jours » fictifs, la notion de l'heure en jeu est très souvent négligée, alors qu'elle peut se révéler très importante dans certaines situations ! L'heure est donc modifiable par le MJ à tout moment par la pression de boutons dédiés.
- 6- **Lanceur de dé.** Le MJ peut par une simple commande (ex : « 2d6 » pour lancer 2 dé à 6 faces) lancer le nombre de dés souhaités et voir le résultat s'afficher. Cela peut s'avérer pratique pour faciliter les nombreux jets de dés ou en cas de manque de certain type de dé<sup>4</sup>.
- 7- **Description des créatures.** Le MJ peut à tout moment ajouter une section pour pouvoir décrire avec plus de précision les monstres ou personnages présents dans le tableau des statistiques : mode de vie, type d'attaques, particularités physiques... Il peut également importer une image.

## 2.2 ~ Onglet Joueurs

Cet onglet sera uniquement disponible en partie réelle. Dans le cas contraire (modification de campagne), cet onglet sera vide.

L'intérêt principal de cet onglet est qu'il **présente les informations affichées sur l'écran des joueurs**. Ces informations seront les caractéristiques principales de chaque joueur, qui seront amenées à changer lors de la partie (PV, CA, bonus, effets temporaires...). Le MJ pourra alors modifier ces informations et elles seront actualisées automatiquement sur l'écran des joueurs en question.

Cet écran permet donc au MJ de garder un œil sur les infos importantes des joueurs et évite aux joueurs d'oublier de modifier leurs caractéristiques quand il le faut : en effet, **les effets temporaires** (ex : un personnage enchante son arme ce qui lui prodigue un bonus de +2 aux dégâts pendant 3 tours), **seront décrémentés automatiquement** et la durée restante affichée sur le téléphone du joueur.

<sup>3</sup> L'ordre des tours de jeu est déterminé par un jet de dé de chaque joueur et dépend de la caractéristique d'Initiative propre à chaque personnage.

<sup>4</sup> Les JDR utilisent souvent des dés bien spécifiques (dés à 4/8/12/20 faces)





1- **Sous onglet joueur.** Lors de la connexion des joueurs, ils sont créés automatiquement, et affiche le nom du personnage. Chaque personnage dispose alors de sa propre page avec ses propres caractéristiques.

2- **Tableau statistiques.** Ce tableau montre les informations importantes du personnage, que les joueurs auront à remplir lors de leur connexion. Elles sont modifiables par le MJ et automatiquement actualisées sur le téléphone.

3- **Tableau Bonus/Effets temporaires.** Ce tableau peut être rempli uniquement par le MJ, il liste les effets temporaires subis par le joueur. Comme précisé précédemment, ces informations sont aussi affichées sur le téléphone du joueur et la durée des effets (ex: «enflammé pendant 3 rounds») sera mis à jour automatiquement.

Bonus /Malus	Durée	Unité
Devient géant	3	heures
...	...	...

Commande: ex 1d20    Résultat    Heure en jeu    12 h    Passer

## 2.3 ~ Onglet Règles

Les règles des JDR sont le plus souvent disponibles au format papier sous forme de livre, ou également sous forme de PDF. Cet onglet permet au MJ d'importer un PDF de son choix et de le consulter directement dans le logiciel, sans avoir à utiliser un logiciel tiers.



---

## 2.4 ~ Onglet Histoire

L'onglet Histoire comportera la possibilité d'ajouter des sous onglets aux noms personnalisables (ex: sous-onglet « Scénario »/ « Lieux »/ « PNJ »...) Ces sous onglets comportent des **zones de textes personnalisables**. Ici le MJ pourra créer son univers, son histoire, et y annoter toutes les informations qui lui seront utiles en cours de partie.

## 2.5 ~ Onglet Génération

Pour un MJ, trouver des idées de rencontres originales n'est pas toujours facile. C'est pourquoi notre logiciel permet de générer des idées de **monstres**, **PNJ** ou **objets**. Le MJ rentre des informations à propos de l'entité qu'il veut créer, en fonction de ses besoins. Exemple: « Générer un monstre de niveau [4], de type [mort-vivant], qui vit dans les [souterrains]<sup>5</sup> ». Nous disposons d'une base de données qui contient toutes les informations présentes dans les différents livres de règles de l'édition 3.5 de Donjons et Dragons, et qui va permettre au logiciel de créer aléatoirement des entités en prenant en compte les informations sélectionnées par le MJ, et qui répond donc à ses besoins. Exemple de réponse: « Un squelette démoniaque, armé d'une cimeterre. Ses caractéristiques sont ... » suivi d'une description détaillée de toutes ses statistiques.

Il en sera de même pour la création d'objet (Exemple: « Générer un [anneau] [non magique] [rare] » → « Anneau en or et diamant d'une valeur de 3000 pièces d'or ») ou de PNJ (Exemple: « Générer un PNJ [Nain] [Barbare] [Homme] de niveau [4] » → « Tordek Markvarth, vêtu d'une cote de maille et armé d'une hache à 2 main. Ses caractéristiques sont: ... »)

## 3 ~ Présentation de l'application

L'application est réservée aux joueurs. Elle est **uniquement utile en cas de partie réelle**. En effet, elle a pour but principal **d'afficher en temps réel les caractéristiques du personnage incarné par le joueur**.

Les joueurs doivent rejoindre une campagne générée en rentrant une adresse fournie par le logiciel. Deux cas de figures sont envisageables :

- ~ C'est la première fois que les joueurs rejoignent la campagne. Ils devront alors rentrer leurs informations<sup>6</sup> importantes. C'est seulement une fois cela fait qu'ils pourront accéder à l'écran principal de l'application.
- ~ Les joueurs reprennent une partie existante<sup>7</sup>. Une liste de noms des personnages s'affiche et chacun des joueurs sélectionne le sien. Les informations relatives à chaque joueur (sauvegardées sur le logiciel PC) seront alors transmises à chaque téléphone. Les joueurs auront alors accès à toutes leurs informations, exactement comme ils les avaient laissées lors de la session précédente.

---

<sup>5</sup> Les éléments entre crochets sont des listes d'éléments sélectionnables et modifiables, que nous avons jugé pertinent pour la création de monstres

<sup>6</sup> Chaque joueur dispose d'une Feuille de Personnage qui contient toutes les informations relatives à son avatar. Elle est très complète, et fourmille d'informations. Le joueur n'aura évidemment pas à rentrer toutes les infos mais seulement celles qui seront amenées à changer.



<sup>7</sup> Une campagne de JDR peut se clôturer en une seule session, mais il est fréquent que les joueurs gardent leur avatar sur plusieurs sessions. Une campagne peut alors s'étaler sur plusieurs semaines voire plusieurs mois.




Voici une vision de l'écran principal de l'application

Non du perso

CompétencesDonsSorts

 24  13

 22 : 56

Malus

Effet 1	Durée 1	Heure(s)
Effet 2	Durée 2	Round(s)

Bonus

Effet 1	Durée 1	Round(s)
Effet 2	Durée 2	Heure(s)

L'appli présente **les statistiques principales** des personnages qui seront amenés à changer (ici les PV et la CA), ainsi que le **tableau des effets temporaires** et **l'heure en jeu**.

**3 autres onglets** sont également présents : Compétences, Dons et Sorts<sup>8</sup>. Ces onglets prendront la forme de **zones de texte**, que le joueur pourra ajouter à souhait, afin de lister et de décrire tous ses moyens d'actions, et aussi d'ajouter des notes personnelles.

<sup>8</sup> Dans Donjons et Dragons, tous les personnages possèdent des Compétences, des Dons et certains des Sorts. Ces caractéristiques sont propres à chaque personnage et bénéficient de longues descriptions qu'il est souvent difficile de retenir sans jeter un coup d'œil dans le livre des règles.



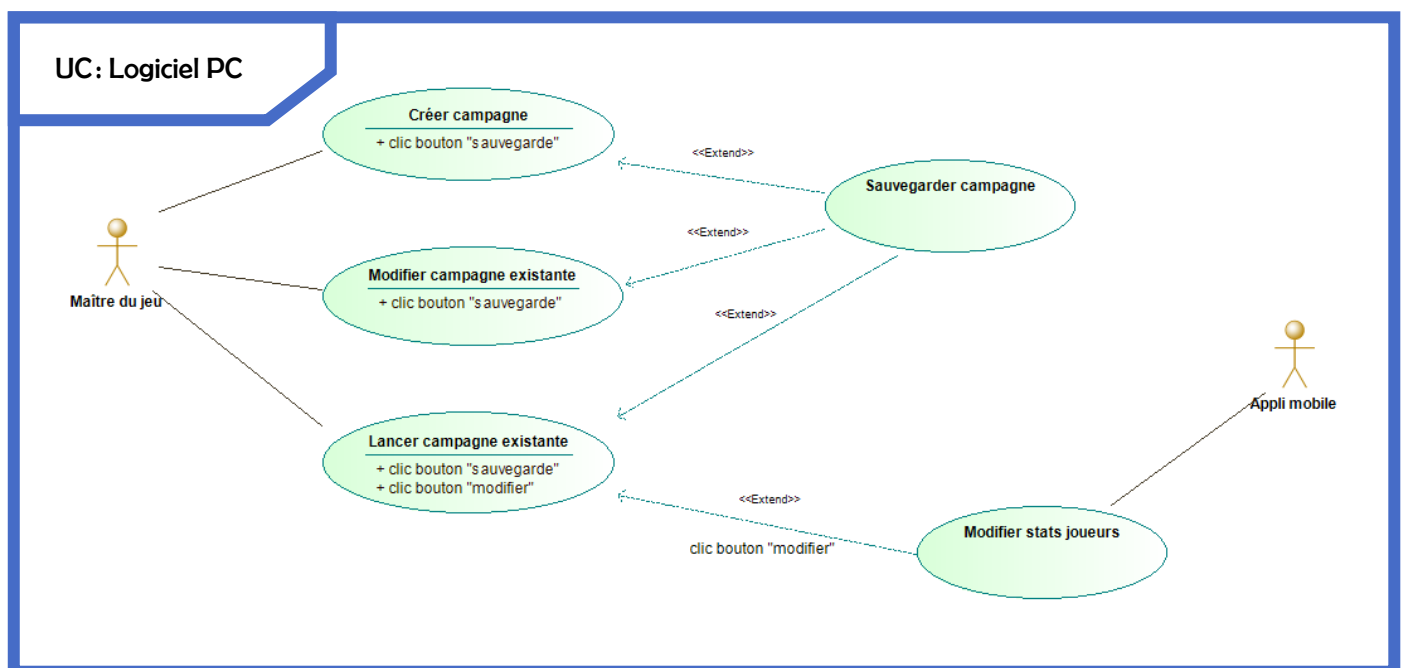
# Avant de coder, bien modéliser

Dans le cas d'un projet informatique en phase de conception comme le nôtre, il est vital de réaliser des diagrammes explicitant les aspects du projet sous différents points de vue. Cela nous permet de bien saisir les différents besoins des utilisateurs et ensuite de détailler les solutions que nous allons implémenter.

Nous utiliserons le langage UML (Unified Modeling Language), couramment utilisé en développement logiciel et en conception orientée objet.

## 1 ~ Diagrammes de cas d'utilisation

### 1.1~ Logiciel PC



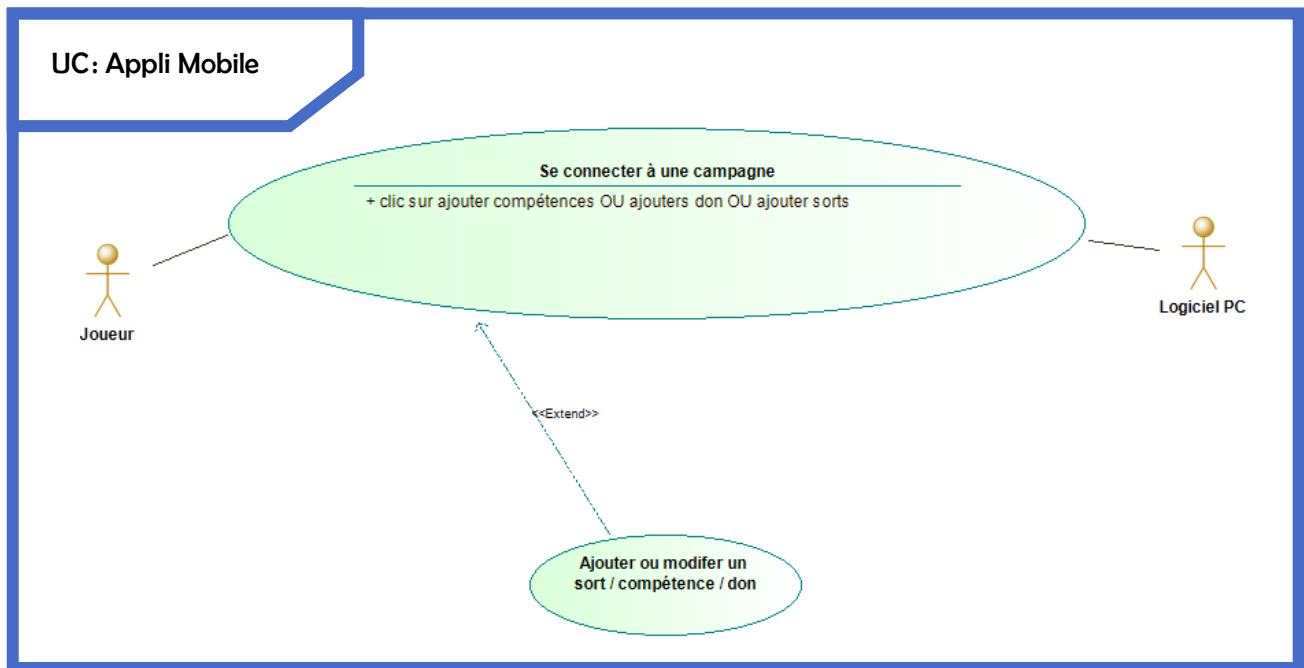
Le logiciel PC est destiné au MJ.

Lors du lancement de l'application, le MJ peut choisir entre plusieurs options de lancement : Il peut **Créer une nouvelle campagne**, **modifier une campagne** créée au préalable ou encore **lancer une campagne** déjà créée. Lorsqu'une campagne est lancée sur l'appli, le MJ peut à tout moment **modifier les statistiques** des joueurs (PV, CA...) ainsi que leur infliger des bonus ou malus de jeu, ces modifications sont ensuite envoyées sur l'application mobile, qui permettra leur affichage auprès des joueurs.

De plus, lorsqu'une campagne est en cours, il peut la **sauvegarder** afin de pouvoir récupérer les différentes informations ultérieurement (Lors d'une prochaine session).



## 1.2~ Application mobile



L'application mobile est quant à elle uniquement utilisée par les joueurs, en cas de partie réelle.

Elle permet aux joueurs de **se connecter à une campagne** générée par le MJ et octroie un accès simple aux informations principales les concernant en cours de jeu. Une fois connecté à la campagne, le joueur a la possibilité de **rentrer des informations** complémentaires concernant son personnage : Compétences, Sorts et Dons.



## 2 ~ Scénarii

### 2.1 ~ Connexion campagne depuis l'application mobile

<b>Scénario Nominal</b> Les joueurs rejoignent une nouvelle partie	<b>Scénario Alternatif</b> Les joueurs rejoignent une partie existante	<b>Scénario Erreur</b> Problème de connexion
1 - Entrer adresse IP	1 - Entrer adresse IP	1 - Entrer adresse IP
2 - Appuyer sur "Rejoindre Partie"	2 - Appuyer sur "Rejoindre Partie"	2 - Appuyer sur "Rejoindre Partie"
3 - Affichage page de renseignement	3 - Affichage page choix personnage existant	<b>3 - Affichage message erreur "Connexion Impossible"</b>
3.1 - Rentrer ses informations dans les zones dédiées	3.1 - Appui sur le bouton personnage de son choix	
3.2 - Appuyer sur "Valider"	3.2 - Appui sur "Valider"	
4 - Affichage page principale	4 - Affichage page principale	
<b>Optionnel :</b>	<b>Optionnel :</b>	
5 - Appui sur "Compétence/Dons/Sort"	5 - Appui sur "Compétence/Dons/Sort"	
5.1 - Rentrer du texte dans les zones dédiées	5.1 - Rentrer du texte dans les zones dédiées	



## 2.2 ~ Création campagne sur l'application PC

<b>Scénario Nominal</b> Le MJ crée une nouvelle campagne	<b>Scénario Alternatif</b>
1 - le MJ clique sur créer	1 - le MJ clique sur créer
1.1 - ouvre un pop-up qui demande le nom de la campagne	1.1 - ouvre un pop-up qui demande le nom de la campagne
1.2 - le MJ rentre le nom et clique sur valider	1.2 - le MJ rentre le nom et clique sur valider
2 - la campagne, vide, est créée d'après le nom donné par le MJ	<b>1.2.1 - Le nom de la campagne est déjà pris : affichage "nom de campagne déjà pris"</b>
<b>Optionnel :</b>	1.2.2 - Le MJ rentre un nom valide
3 - cliquer sur l'onglet Histoire et remplir du texte BBCode <sup>9</sup> puis valider	#Reprise à l'étape 2 du scénario nominal
4 - cliquer sur combat et créer une rencontre	
4.1 - ajouter des monstres avec des statistiques	
4.2 - ajouter une description de monstre (texte, image)	
5 - ajouter un PDF de règles	
6 - générer	
6.1 - monstres	
6.2 - lancer de dé	
6.3 - PNJ	
6.4 - objets	

<sup>9</sup> Le BBCode permet de donner un style (couleur, police d'écriture, italique..) à un texte.



## 2.3 ~ Modification campagne sur l'application PC

<b>Scénario Nominal</b> Le MJ modifie une campagne existante	<b>Scénario Erreur</b>
1 - Cliquer sur une campagne existante	1 - Cliquer sur une campagne existante
2 - Cliquer sur "Modifier"	2 - Cliquer sur "Modifier"
3 - La campagne existante s'affiche	<b>3 - La campagne existante n'a pas pu être récupérée</b>
<b>Optionnel :</b>	3.1 - Pop-up affichant "la campagne n'a pas pu être récupérée"
4 - cliquer sur l'onglet Histoire et remplir/modifier du texte BBcode puis valider	3.1 - Le MJ ferme le pop-up et retour à l'écran de sélection des campagnes
5 - cliquer sur combat et créer une rencontre	
5.1 - ajouter des monstres avec des statistiques	
5.2- ajouter une description de monstre (texte, image)	
6 - ajouter un PDF de règles	
7 - générer	
7.1 - monstres	
7.2 - lancer de dé	
7.3 - PNJ	
7.4 - objets	





## 2.4~ Lancement Campagne sur l'application PC

<b>Scénario Nominal</b> Le MJ lance la campagne	<b>Scénario Alternatif</b>
1 - Le MJ clique sur "lancer" à côté de la campagne souhaitée (recherche réseau)	1 - Le MJ clique sur "lancer" à côté de la campagne souhaitée (recherche réseau)
1.2 - Affichage du lobby de connexion des joueurs	1.1 - Affichage du lobby de connexion des joueurs
1.3 - Attente de connexion des joueurs	1.2 - Attente de connexion des joueurs
2 - Le MJ clique sur démarrer	2 - Le MJ clique sur démarrer
3 - l'interface de gestion s'affiche	<b>2.1 - Il n'y a aucun joueur de connecté</b>
<b>optionnel:</b>	2.2 - Affichage pop-up erreur "il n'y a aucun joueur de connecté"
4 - Modifier statistiques joueurs	2.3 - Le MJ ferme le pop-up et retour à l'écran d'attente de connexion des joueurs
5 - cliquer sur l'onglet Histoire et remplir/modifier du texte BBcode puis valider	
6 - cliquer sur combat et créer une rencontre	
6.1 - ajouter des monstres avec des statistiques	
6.2 - ajouter une description de monstre (texte, image)	
7 - ajouter un PDF de règles	
8 - générer	<b>Scénario Alternatif 2</b>
8.1 - monstres	1 - Le MJ clique sur "lancer" à côté de la campagne souhaitée (recherche réseau)
8.2 - lancer de dé	1.1 - Le PC n'est connecté à aucun réseau (pas d'adresse IP)
8.3 - PNJ	1.2 - Affichage pop-up "connectez-vous à un réseau"
8.4 - objets	1.3 - Retour à l'écran de lancement de la campagne



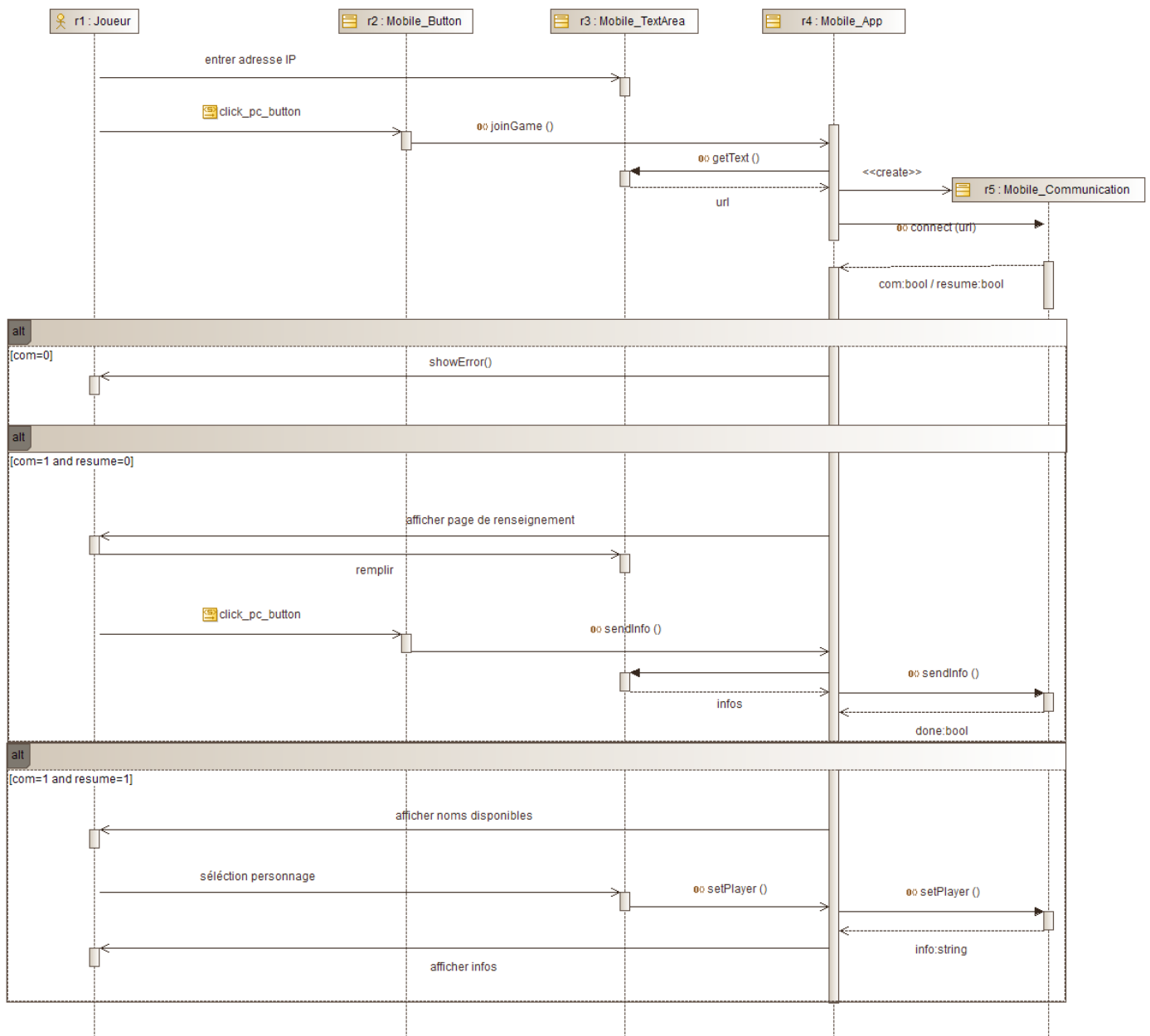
### 3~ Diagrammes de séquence

Nous allons réaliser un diagramme de séquence par scénario nominal.

Les diagrammes de séquence explicitent la communication entre les différentes classes<sup>10</sup> qui entrent en jeu lors d'un scénario.

#### 3.1~ Connexion campagne depuis l'application mobile

##### SEQ : connexion campagne



<sup>10</sup> Une classe est comme un plan de bâtiment, c'est un modèle qui définit les instructions destinées à la construction. Un objet ou une instance de classe est le bâtiment.



---

Ce diagramme représente les interactions entre classes lors de la connexion d'un joueur via l'application mobile. La fenêtre de base de l'application mobile est un simple écran de connexion, contenant une zone de texte dans lequel le joueur vient rentrer l'adresse IP fournie par le logiciel PC. Le joueur appuie ensuite sur le bouton de connexion, ce qui communique l'information à la classe «Mobile\_App». La classe récupère le texte (ici l'adresse IP) contenue dans la zone de texte, puis crée une instance de classe «Mobile\_Communication», qui a pour but de gérer la communication entre l'appli mobile et le logiciel. La communication s'effectue ensuite et le résultat de l'opération est renvoyé sous forme de deux variables booléennes<sup>11</sup> : «com» représente le fait que la connexion a été établie et «resume» sert à savoir si la partie auquel le joueur se connecte est une nouvelle partie ou la suite d'une partie existante. La suite de l'exécution dépend du résultat de l'opération précédente :

~ Si il y a eu un problème avec la connexion (com = 0), l'application affiche une erreur à l'utilisateur, on est alors en cas de scénario d'erreur.

~ Si la connexion est établie (com = 1) et que la partie est nouvelle (resume = 0), l'application affiche au joueur une page de renseignements à remplir afin d'identifier le personnage, le joueur remplit les informations via une zone de texte et, de la même manière que pour l'IP, les informations sont gérées par la classe «Mobile\_Communication».

~ Si la connexion est établie et que c'est une reprise de campagne (resume = 1), l'application affiche au joueur une liste des personnages présents dans cette campagne. Il sélectionne donc le sien, et la classe «Mobile\_Communication» s'occupe d'aller récupérer toutes les informations qui sont stockées sur le logiciel PC afin de pouvoir les afficher sur le téléphone du joueur.

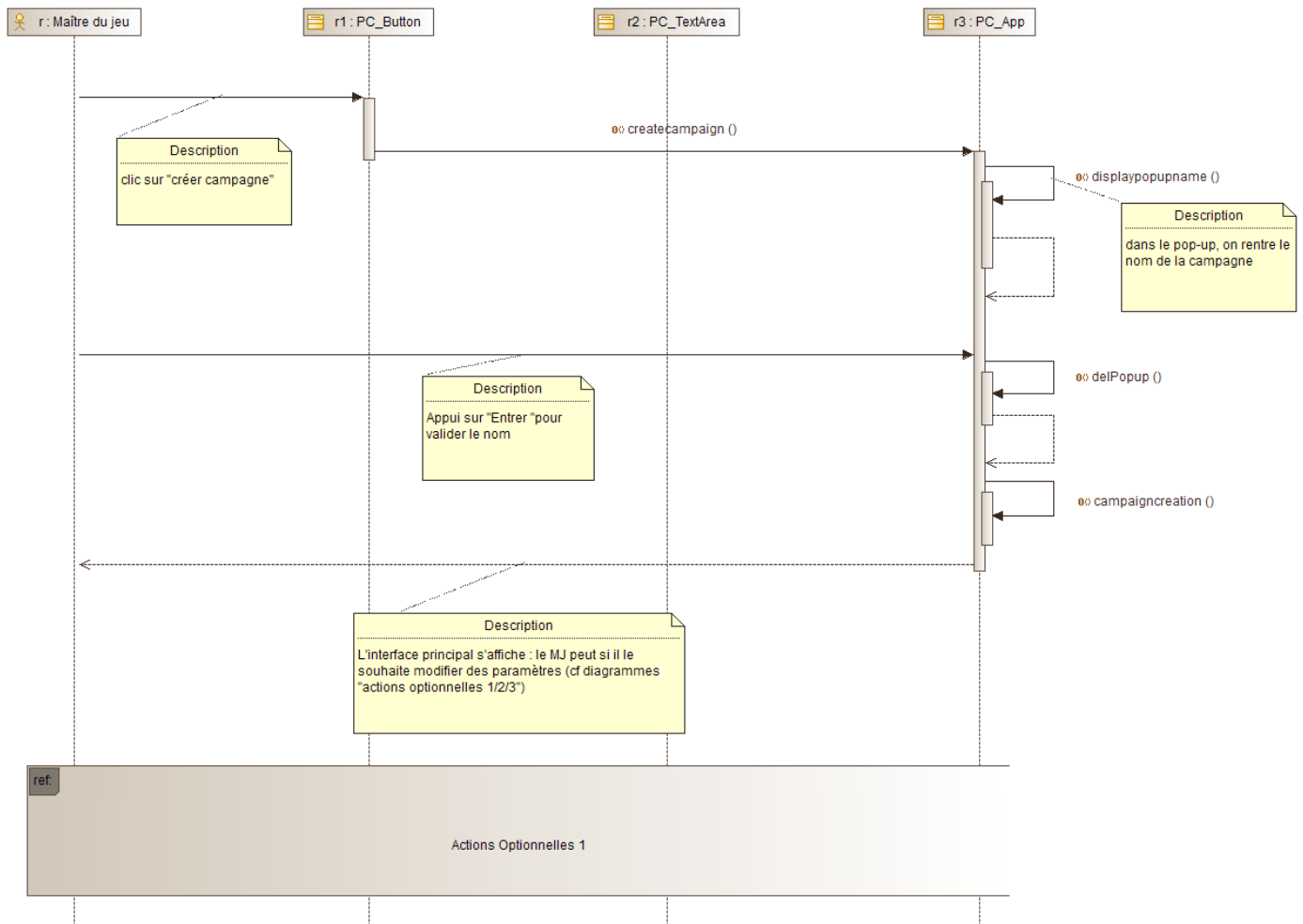
---

<sup>11</sup> Une variable booléenne et une variable qui peut prendre uniquement 2 états : vraie (1) ou faux (0)



### 3.2~ Création campagne sur l'application PC

#### SEQ : création campagne



Ce diagramme représente les interactions entre classes lors de la **création d'une campagne par le MJ**. L'appui d'un bouton dédié fera apparaître un pop-up permettant au MJ de rentrer le nom de sa campagne, et l'appui sur la touche «Entrée» le fera disparaître.

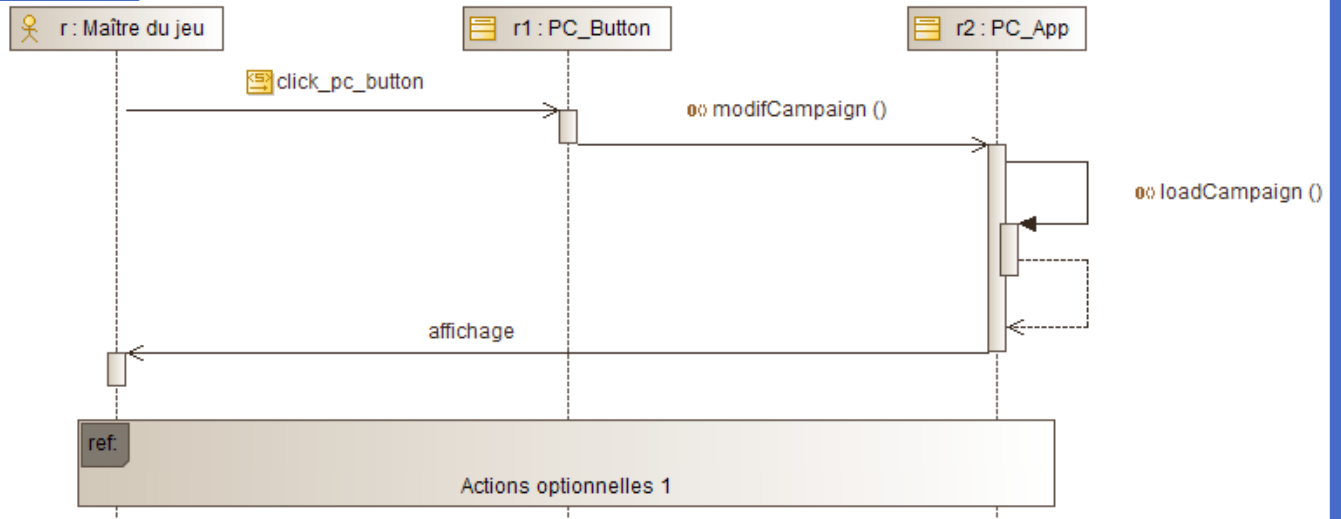
La campagne désormais créée, un panel d'option s'offre alors au MJ (sous forme d'onglets décrits précédemment). Néanmoins toutes ces actions sont considérées comme **optionnelles** : en effet, rien n'oblige le MJ à utiliser tous les outils mis à sa disposition (quel est l'intérêt d'importer un PDF de règles, si l'on possède déjà l'exemplaire physique ?) De plus, la seule condition pour lancer une campagne est qu'elle ait été créée. (Même si le MJ n'a aucun intérêt à lancer une campagne vide)

Lorsqu'on quitte le logiciel, les modifications sont sauvegardées et stockées sur le PC, afin de pouvoir les utiliser ultérieurement.



### 3.3~ Modification campagne sur l'application PC

#### SEQ : modification campagne

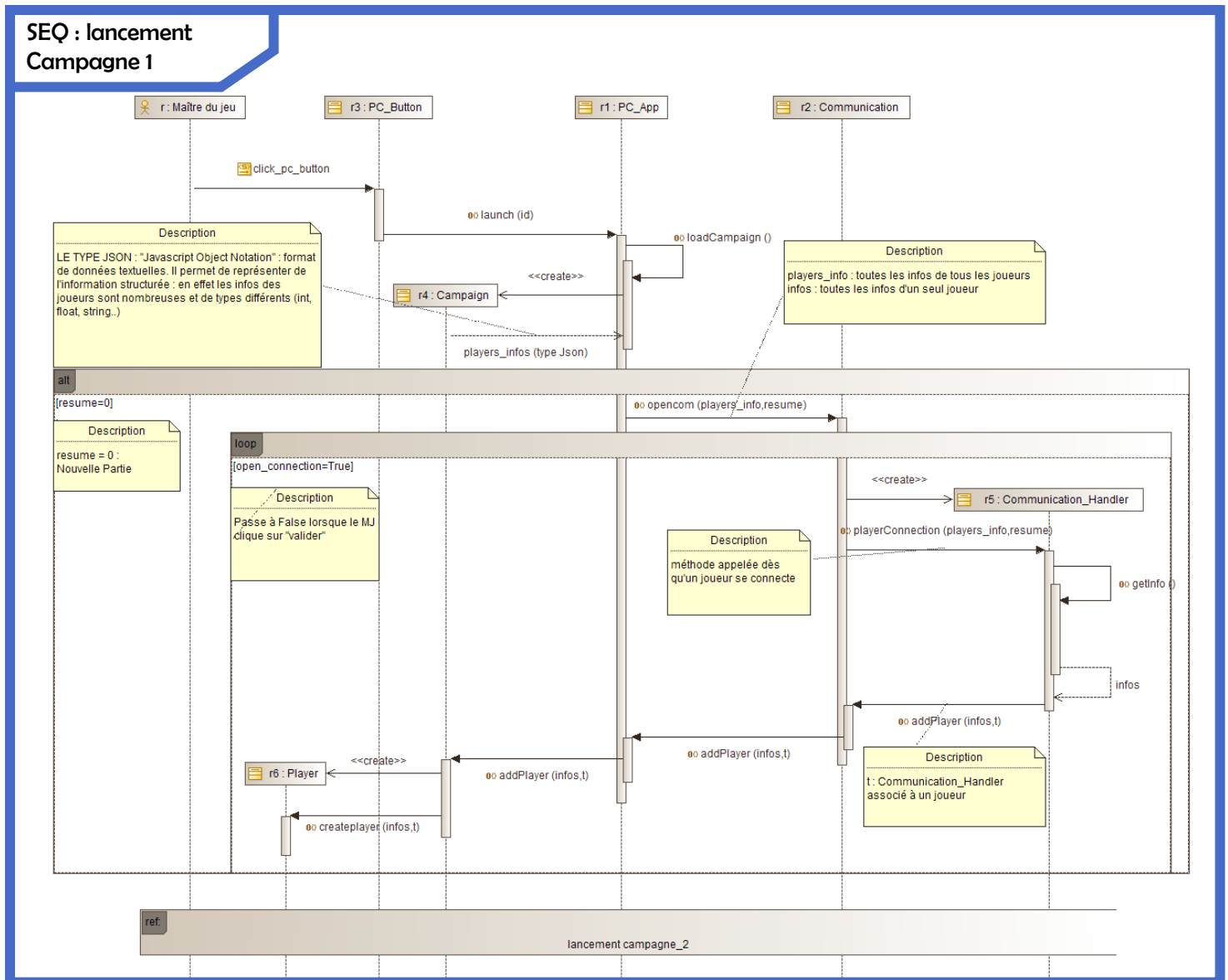


Ce diagramme représente les interactions entre classes lors de la **modification d'une campagne existante**. L'appui d'un bouton dédié chargera la campagne, stockée sur le PC, afin de l'afficher. Comme lors d'une création, les **actions optionnelles** s'offrent au MJ.



### 3.4~ Lancement Campagne sur l'application PC

#### SEQ : lancement Campagne 1



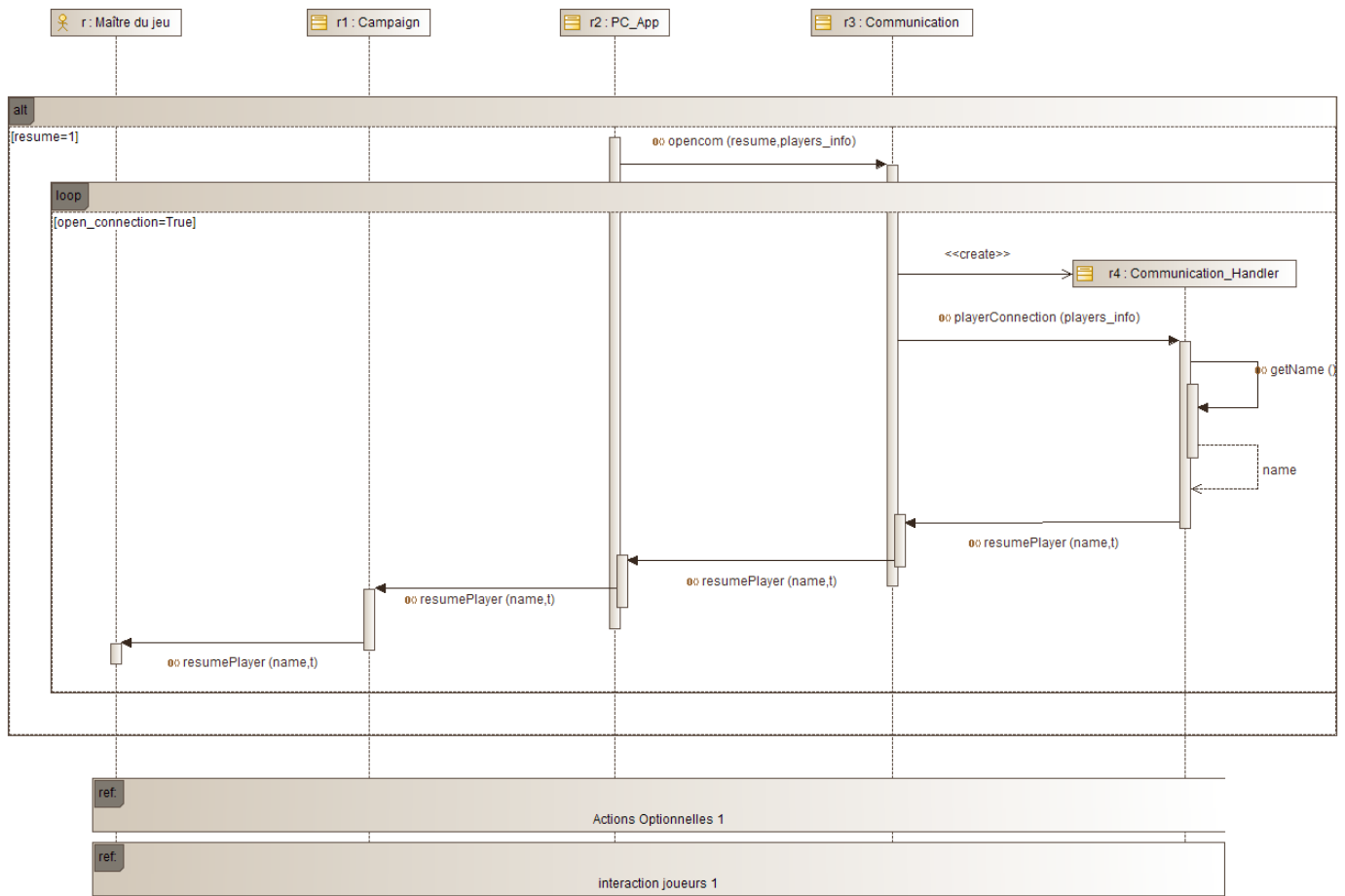
Une fois qu'une campagne est créée, il est possible de la **lancer**. La campagne peut se lancer en cas de partie réelle: lorsque tous les joueurs sont présents autour de la table, prêts à jouer.

Le MJ clique sur le bouton dédié, ce qui a pour effet de charger la campagne stockée sur le PC, et d'afficher un écran d'attente de connexion. Une adresse IP est générée et affichée, ce qui va permettre aux joueurs de se connecter via leur téléphone (cf. « [Connexion campagne depuis l'application mobile](#) »). Le MJ attend alors que les joueurs se connectent. Deux cas de figures sont alors envisageables :

~ Si la campagne n'a jamais été lancée (resume = 0) : aucune information de joueur n'est envoyée (players\_infos = NULL). Dès qu'un joueur se connecte, le logiciel récupère les informations que le joueur a rentrées sur son téléphone (infos) puis crée une instance de classe « Player » complétée. Ce processus se répète pour chaque joueur et se termine dès que le MJ clique sur le bouton « valider ».



## SEQ : lancement Campagne 2



~ Si la campagne a déjà été lancée auparavant (resume = 1) : les infos des personnages sont envoyées au portable (players\_infos), cela va alors permettre à l'application d'afficher les noms des personnages existants et donc de laisser les joueur sélectionner le leur. Le nom est alors retourné au logiciel PC afin de faire le lien entre les informations des joueurs qu'il possède et le joueur en question.

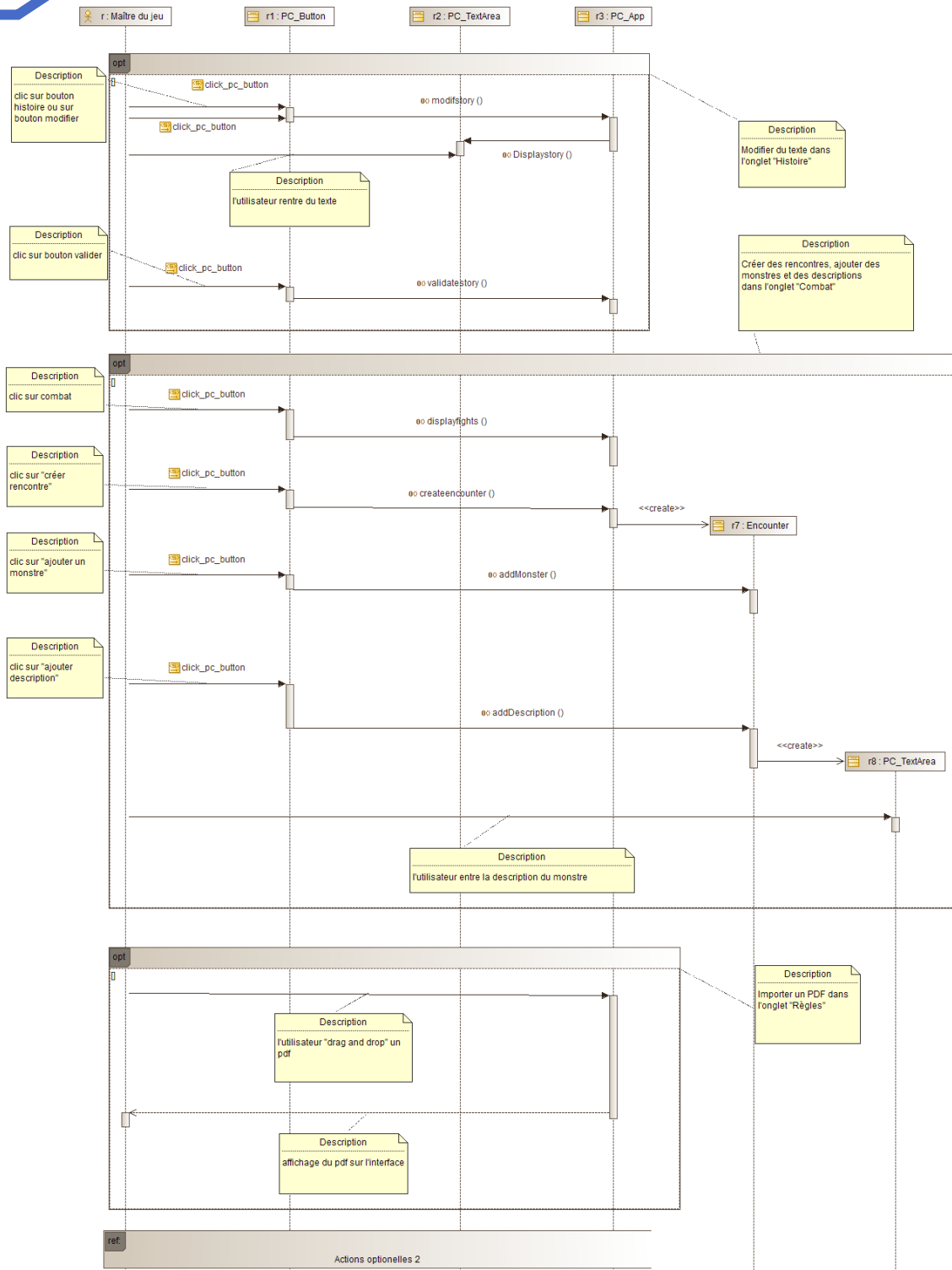
Le lancement d'une campagne permet alors au MJ d'effectuer les **actions optionnelles** mais également **d'interagir avec les joueurs connectés**.



### 3.5~ Actions optionnelles

Ce diagramme est séparé en plusieurs parties étant donné sa grande taille.

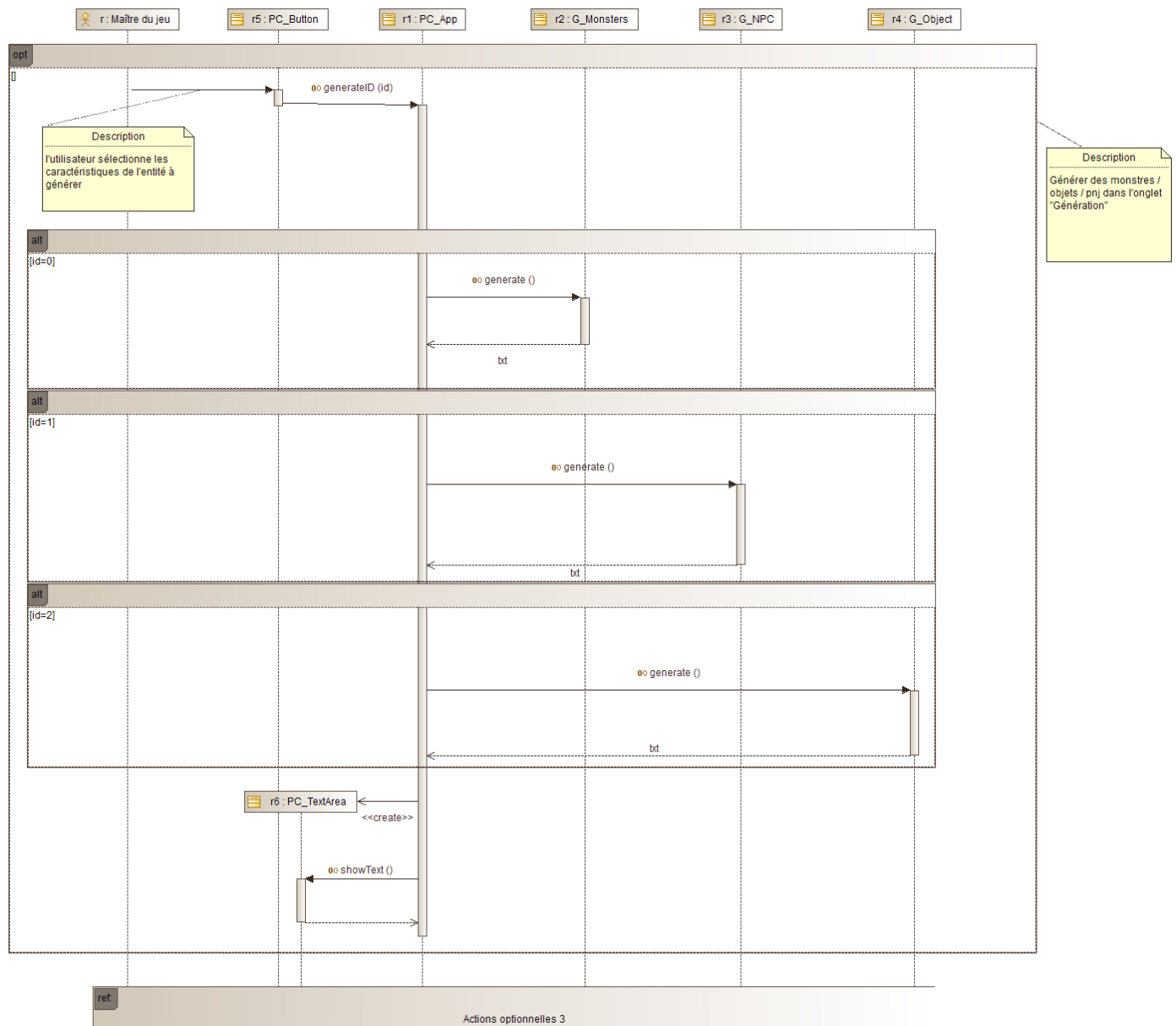
#### SEQ : actions optionnelles 1





SEQ : actions optionnelles 1 représente les interactions entre classes lors de l'utilisation des fonctionnalités possibles dans les onglets **Histoire**, **Combat** et **Règles**. (cf. « [2.1 ~ Onglet Combat](#) », « [2.3 ~ Onglet Règles](#) » & « [2.4 ~ Onglet Histoire](#) », pour une description détaillée de ces fonctionnalités)

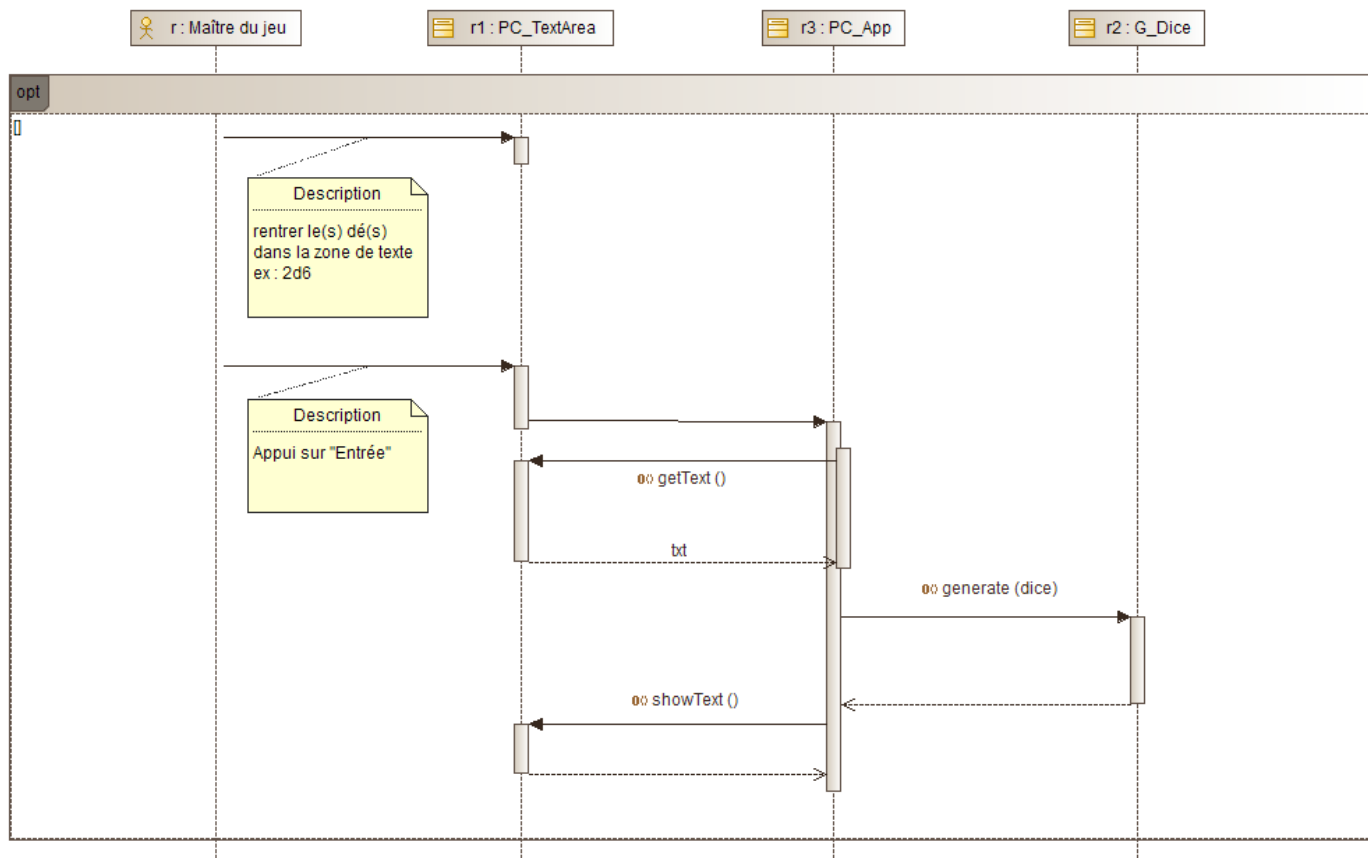
## SEQ : actions optionnelles 2



SEQ : actions optionnelles 2 représente les interactions entre classe lors de l'utilisation des fonctionnalités possibles dans l'onglet **Génération**. (cf. « [2.5 ~ Onglet Génération](#) » pour une description détaillée.)



## SEQ : actions optionnelles 3

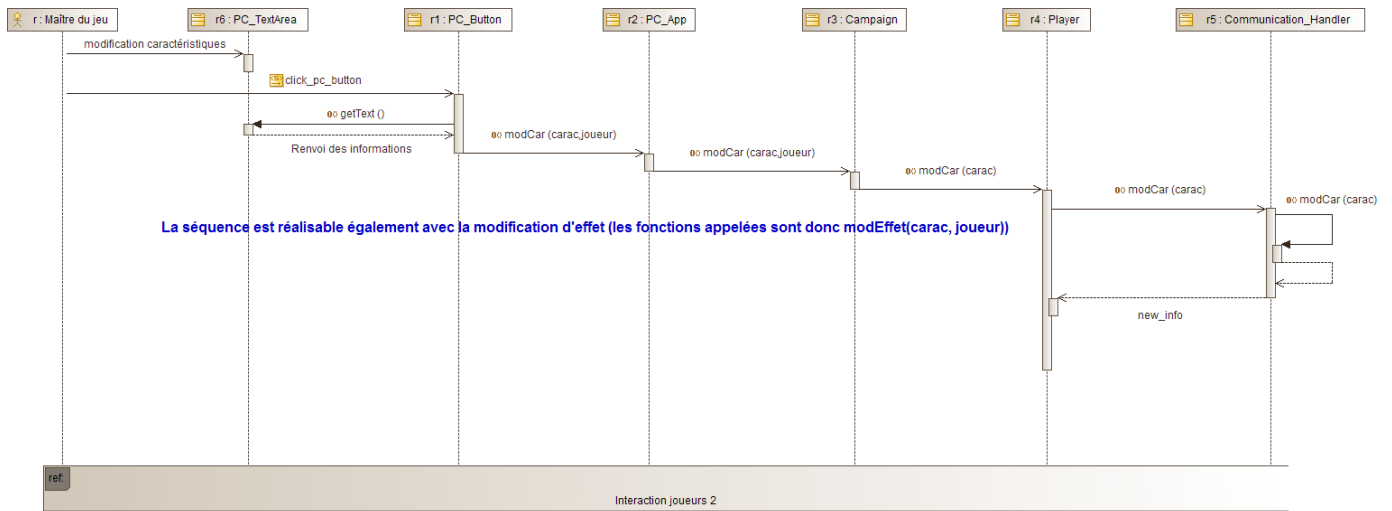


SEQ : actions optionnelles 3 représente l'interaction entre classes lors de la **génération d'un lancer de dé**. (cf. « [2.5 ~ Onglet Combat: 6- Lancer de dé](#) », pour une description de la procédure)



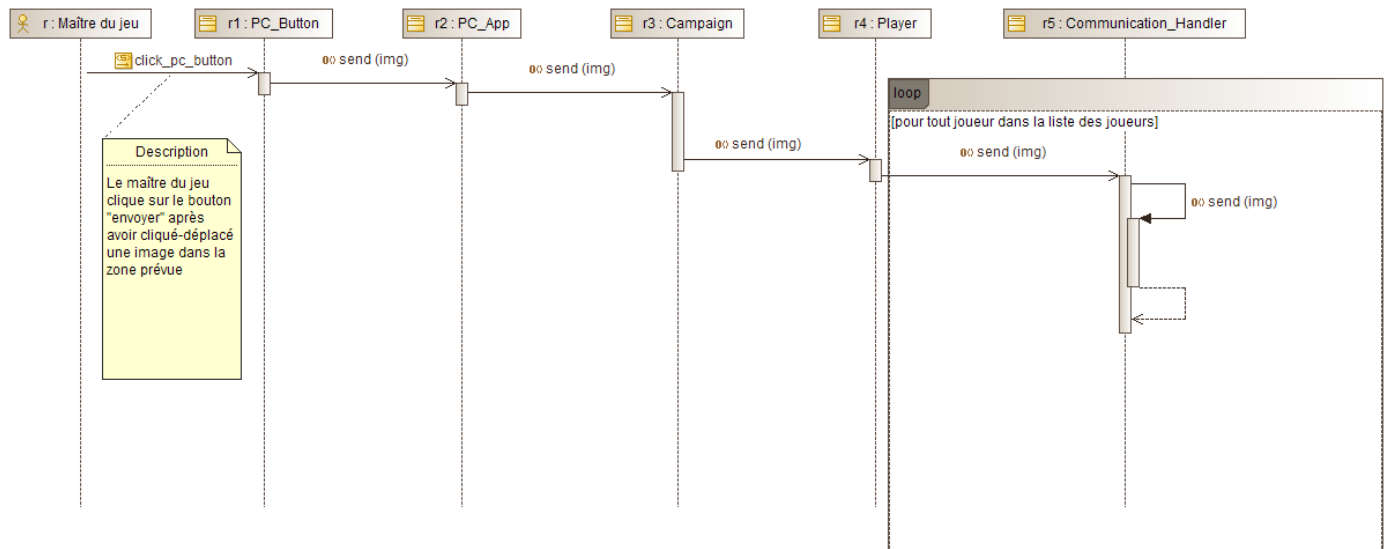
## 3.6~ Interactions Joueurs

### SEQ : interactions joueurs 1



Lors d'une partie réelle, le MJ dispose de la possibilité d'interagir avec les joueurs connecté. Il peut changer leurs statistiques (méthode<sup>12</sup> « `modCar()` ») ou leur ajouter des effets temporaires (méthode « `modEffect()` »)

### SEQ : interactions joueurs 2



Le MJ peut également envoyer une image aux joueurs, en la cliquant-déplaçant dans une zone appropriée et par l'appui d'un bouton dédié. L'image s'affiche alors sur tous les téléphones et peut s'effacer par la pression d'un bouton dédié de la part des joueurs.

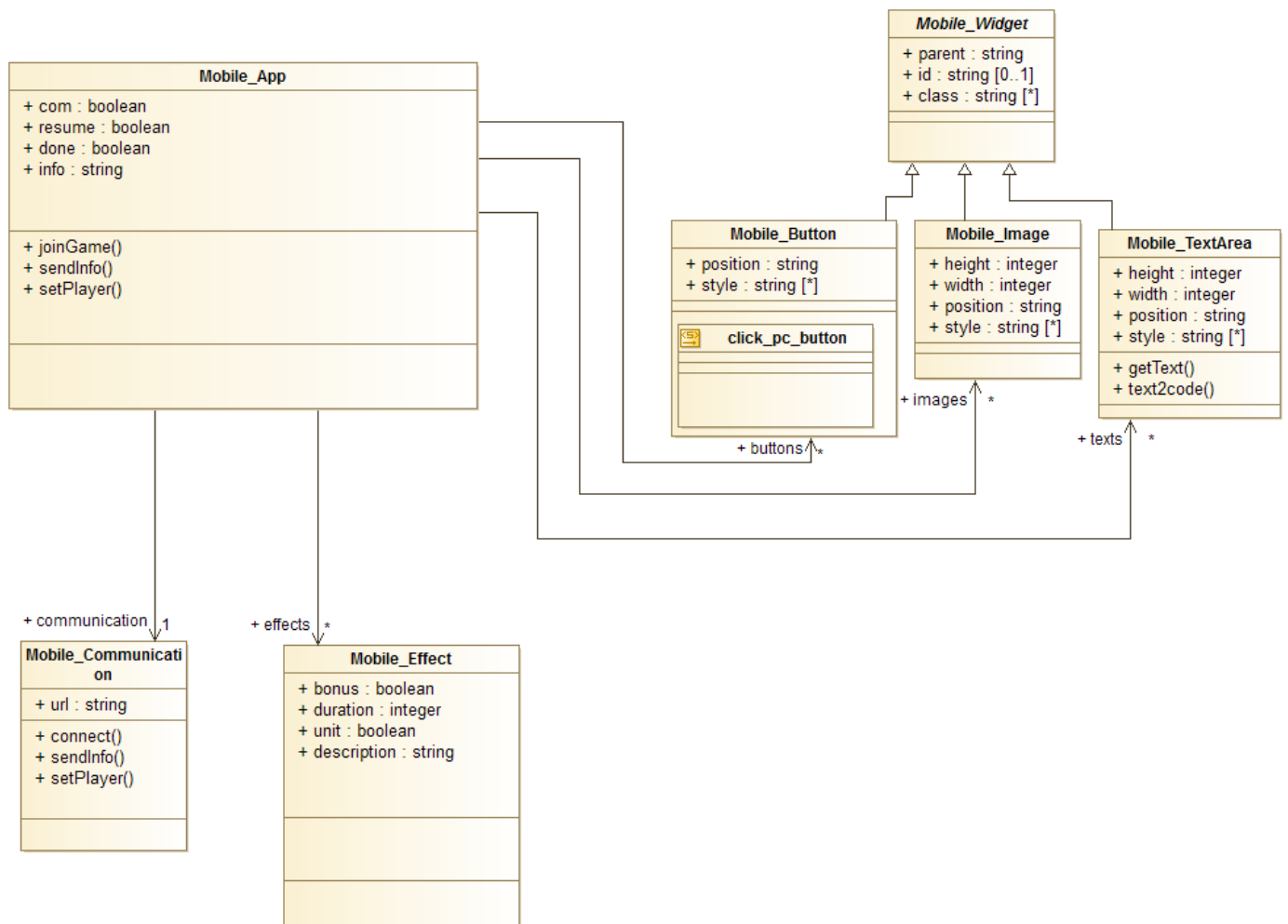
<sup>12</sup> En programmation orientée objet (POO), une méthode est une fonction membre d'une classe.



## 4~ Diagrammes de classe

### 4.1~ Application mobile

#### CLASS : application mobile



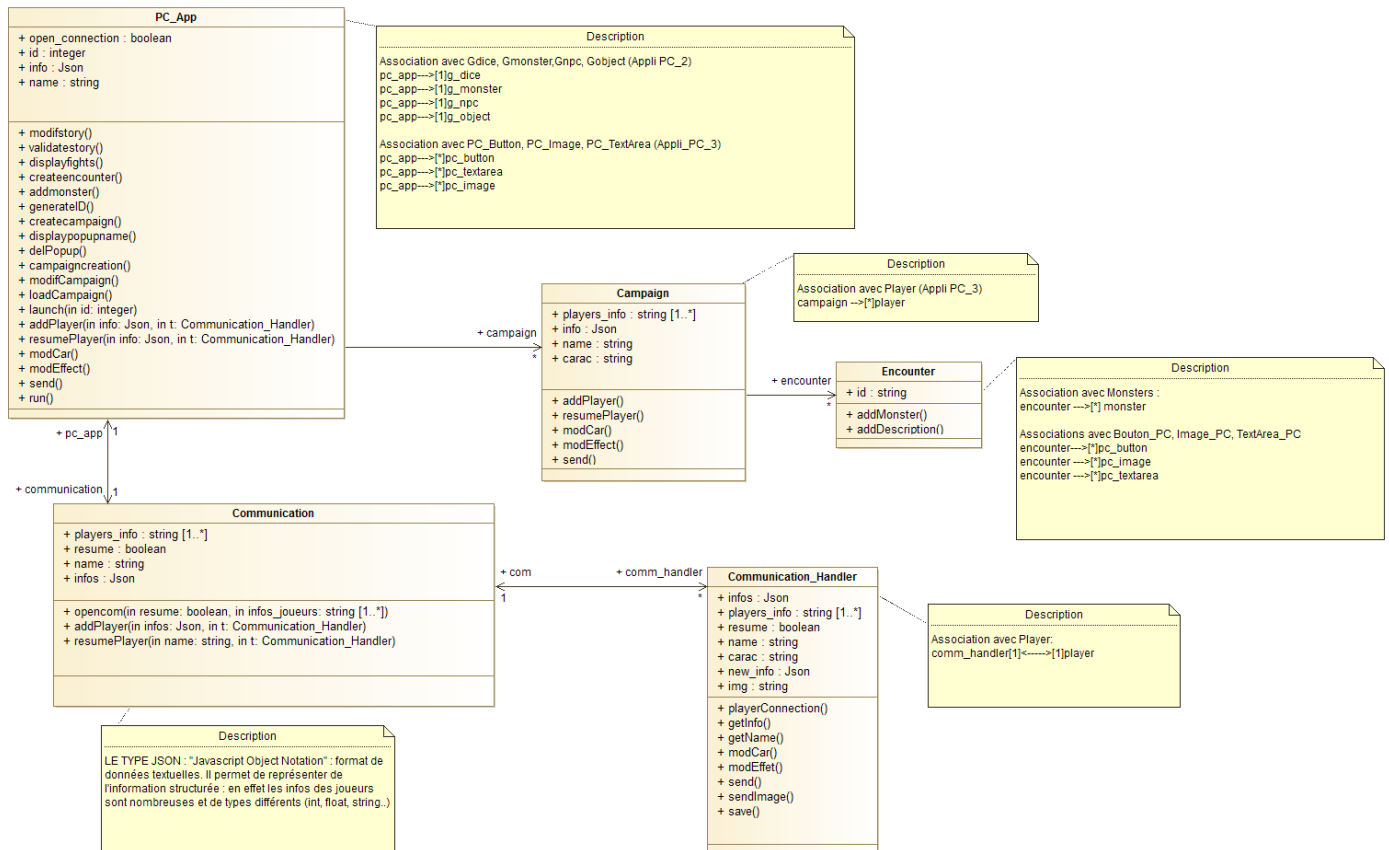
La classe principale du programme est la classe « Application », elle gère la mise en relation des différentes parties de l'application. Cette classe contiendra un objet de la classe « Communication », qui comme son nom l'indique gèrera l'envoi et la réception de données entre l'application et le logiciel. De plus, l'application mobile gèrera une liste d'effets correspondant aux bonus et malus pouvant être envoyés par le MJ via son logiciel. Les différents éléments graphiques appartiennent à une classe « Widget », séparée en différents sous widgets comme les Images, les boutons ou encore les zones de texte. La classe Application contiendra donc ces différents widgets afin de pouvoir les afficher et coordonner les envois de données.



## 4.2~ Logiciel PC

Le diagramme de classe a été séparé en plusieurs vues, pour faciliter la lisibilité. Les liens entre les différents diagrammes sont explicités par les boîtes de descriptions.

### CLASS : Logiciel PC vue 1

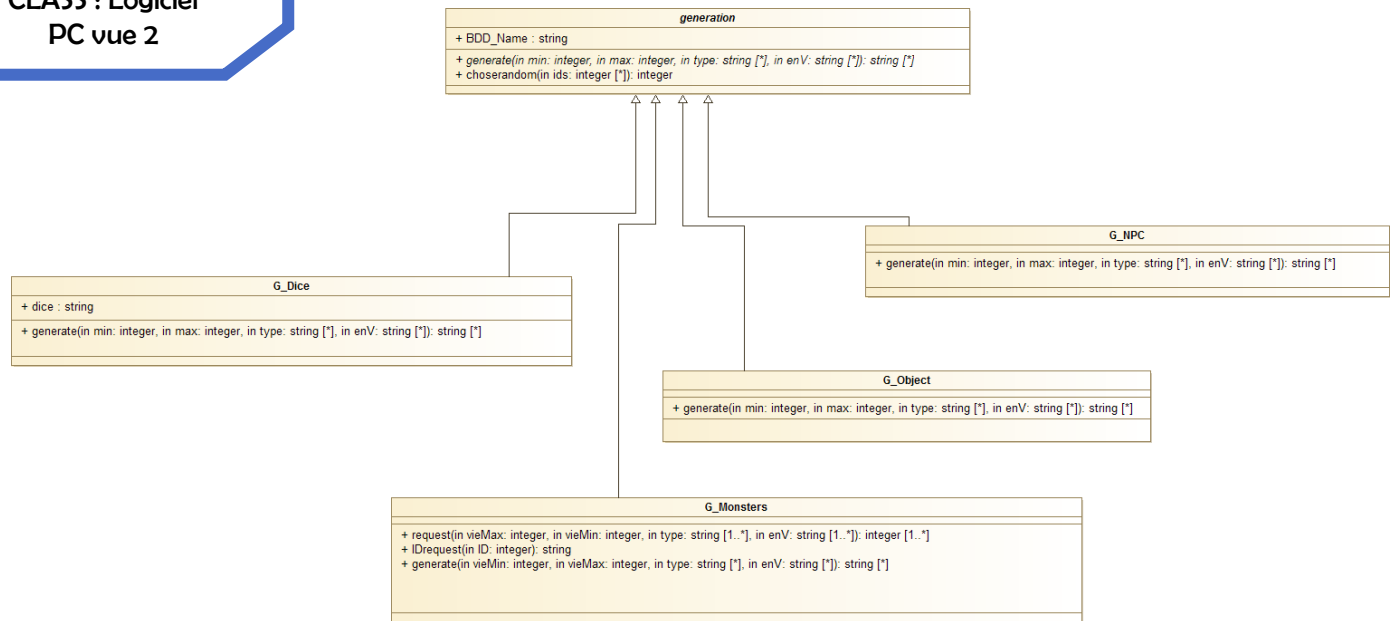


«PC\_App» est la classe centrale de l'application, elle peut être composée de plusieurs campagnes indépendantes qui stockent elle-même toutes les infos importantes (les infos des joueurs, les différentes rencontres..). Elle est également composée d'une instance de «Communication» et plus précisément de «Communication Handler», qui a pour rôle de gérer les interactions entre le logiciel PC et l'application des joueurs.

Cette classe est également composée d'une instance de Génération et de ses dérivés, ainsi que tous les widgets utiles à l'affichage (boutons, zone de texte et images).

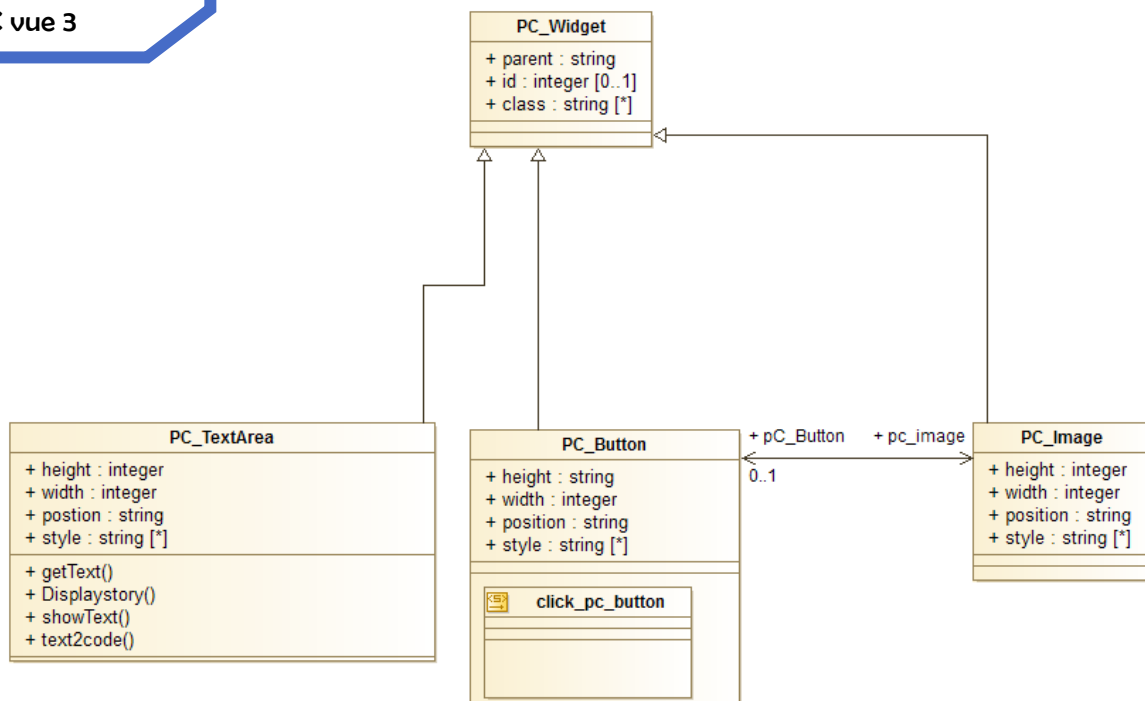


## CLASS : Logiciel PC vue 2



« Generation » est une classe mère qui définit des méthodes abstraites, qui seront redéfinies par ses classes dérivées. En effet, chaque classe dérivée, générera des entités différentes et nécessite une méthode de génération qui lui est propre. Chaque méthode « generate() » utilisera une base de donnée « D&D 3.5 » qui contient toutes les informations contenues dans les livres de règles de Donjon et Dragon version 3.5.

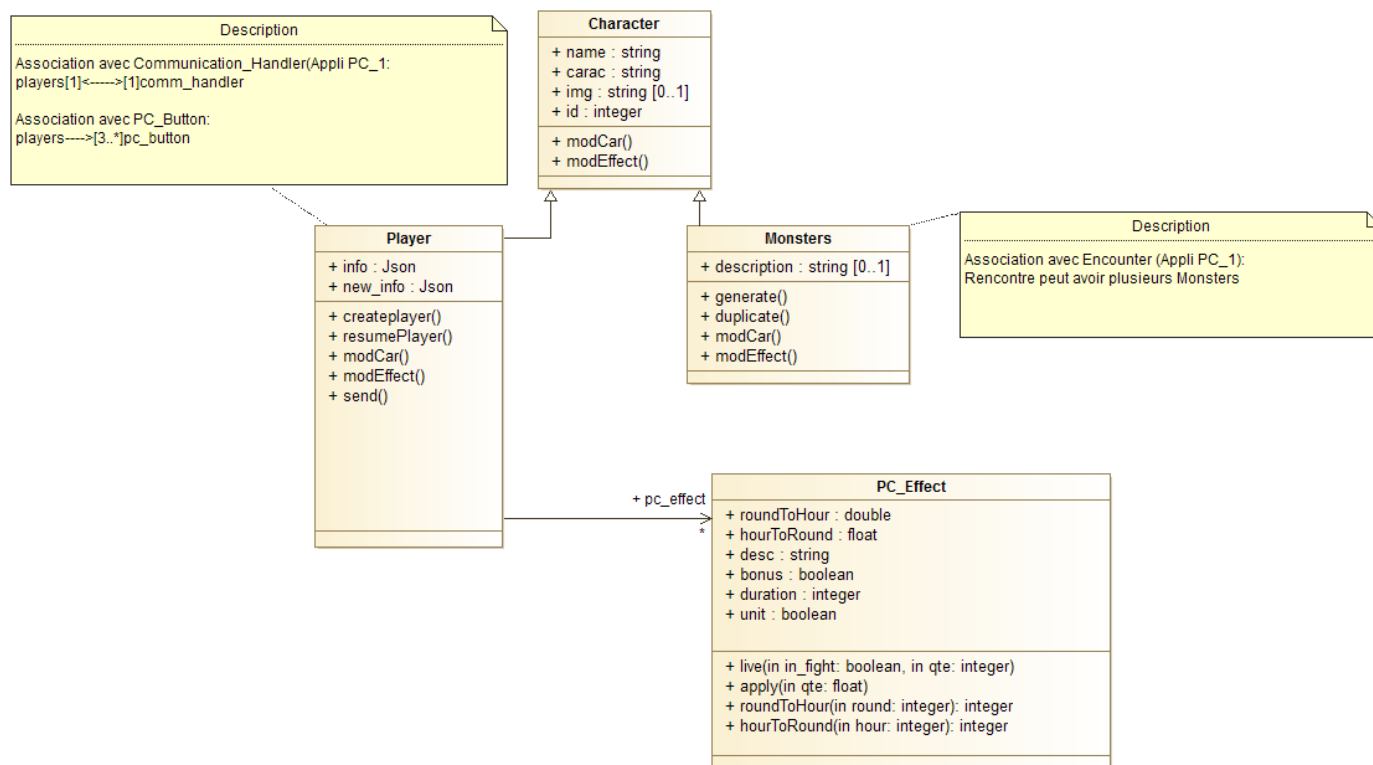
## CLASS : Logiciel PC vue 3



Ce diagramme montre les différents attributs et méthodes des éléments graphiques principaux de l'application PC: les boutons, images et zones de texte.



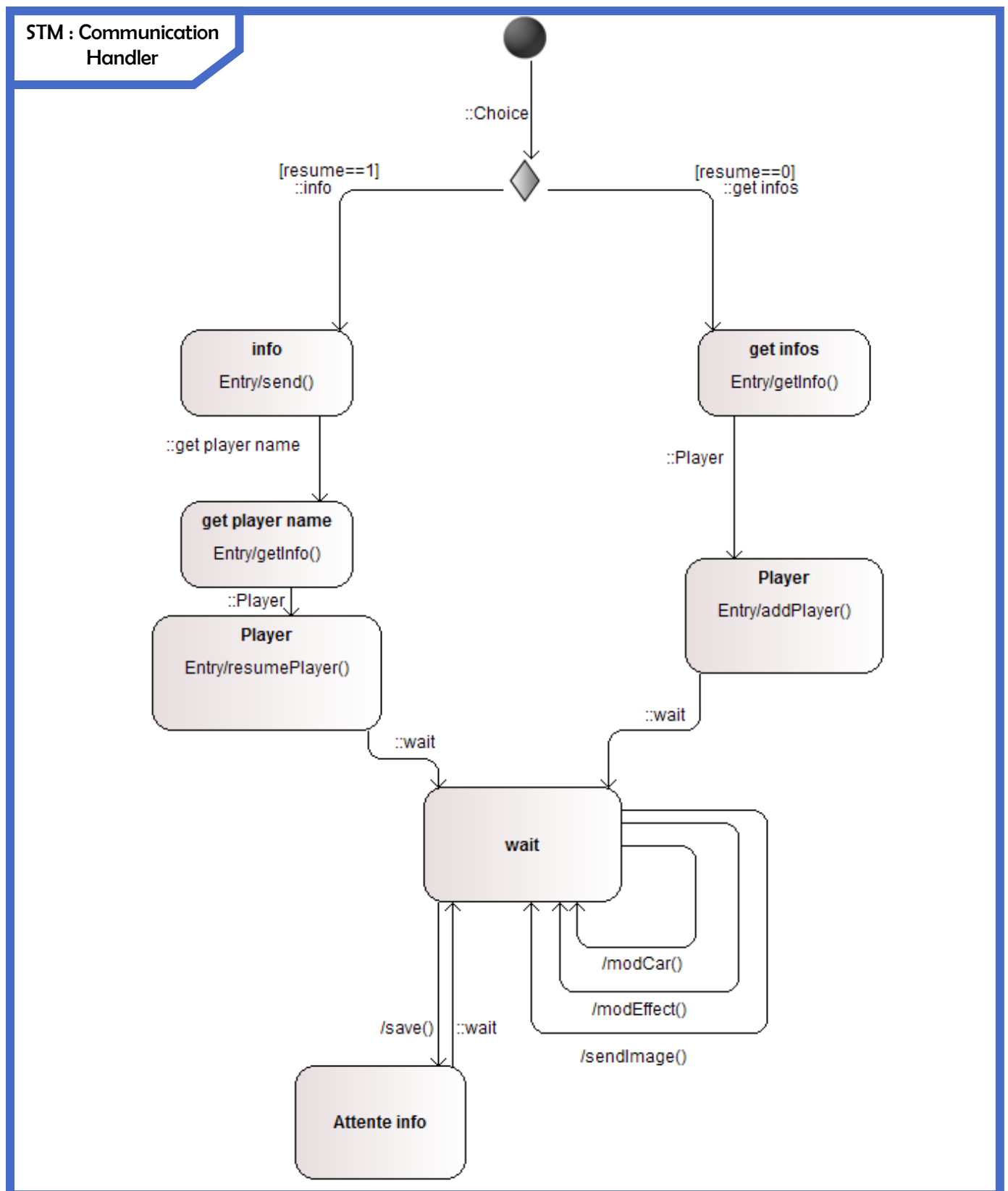
## CLASS : Logiciel PC vue 4



Ce diagramme présente les classes utilisées pour gérer les informations des joueurs et des monstres. Les joueurs et les monstres possèdent des caractéristiques similaires, c'est pourquoi ils dérivent d'une classe « Character ». Chaque joueur peut posséder un certain nombre d'effets différents, sous la forme d'instances de la classe « PC\_effect ».



## 5~ Diagrammes d'état-transition





---

Le diagramme état-transition représente les différents états d'une classe ainsi que les moyens de passer d'un état à un autre. Ici le diagramme représente la classe «Communication\_Handler», une instance de cette classe sera attribuée à chaque joueur, elle permettra au pc de communiquer avec les téléphones. Le point noir est l'entrée, la flèche sortante nous amène à une décision. Est-ce une nouvelle partie (resume = 0) ou la reprise d'une ancienne partie sauvegardée (resume = 1) ?

~ Cas d'une nouvelle partie. Nous sommes renvoyés à l'état «getInfos» cet état se charge de récupérer les infos du joueur présent sur l'application mobile. Puis l'étape suivante instancie un objet de type «Player» avec les informations.

~ Cas d'une reprise. Ici les informations ont été sauvegardées, les noms des joueurs sont donc envoyés aux téléphones pour qu'ils sélectionnent le leur. Puis l'objet «Player» est créé avec les informations correspondant au joueur.

Ainsi les deux branches se réunissent sur l'étape «wait» en attendant une action du MJ. Il peut y en avoir 4 différentes

~ «modCar()». Cette action permet de modifier une caractéristique du joueur sur le téléphone.

~ «modeffect()». Cette action a pour but d'ajouter, supprimer, modifier un effet sur le joueur. Exemple d'effet: «en feu pendant 3 tour».

~ «sendImage()». Comme vu plus haut le MJ a la possibilité d'envoyer des images à tous les joueurs.

~ «Save()». Cette action sera utilisée à la fin de la session de jeu. Elle permettra d'enregistrer toutes les notes que les joueurs ont rentrées sur leurs téléphones ainsi que leurs caractéristiques actuelles.

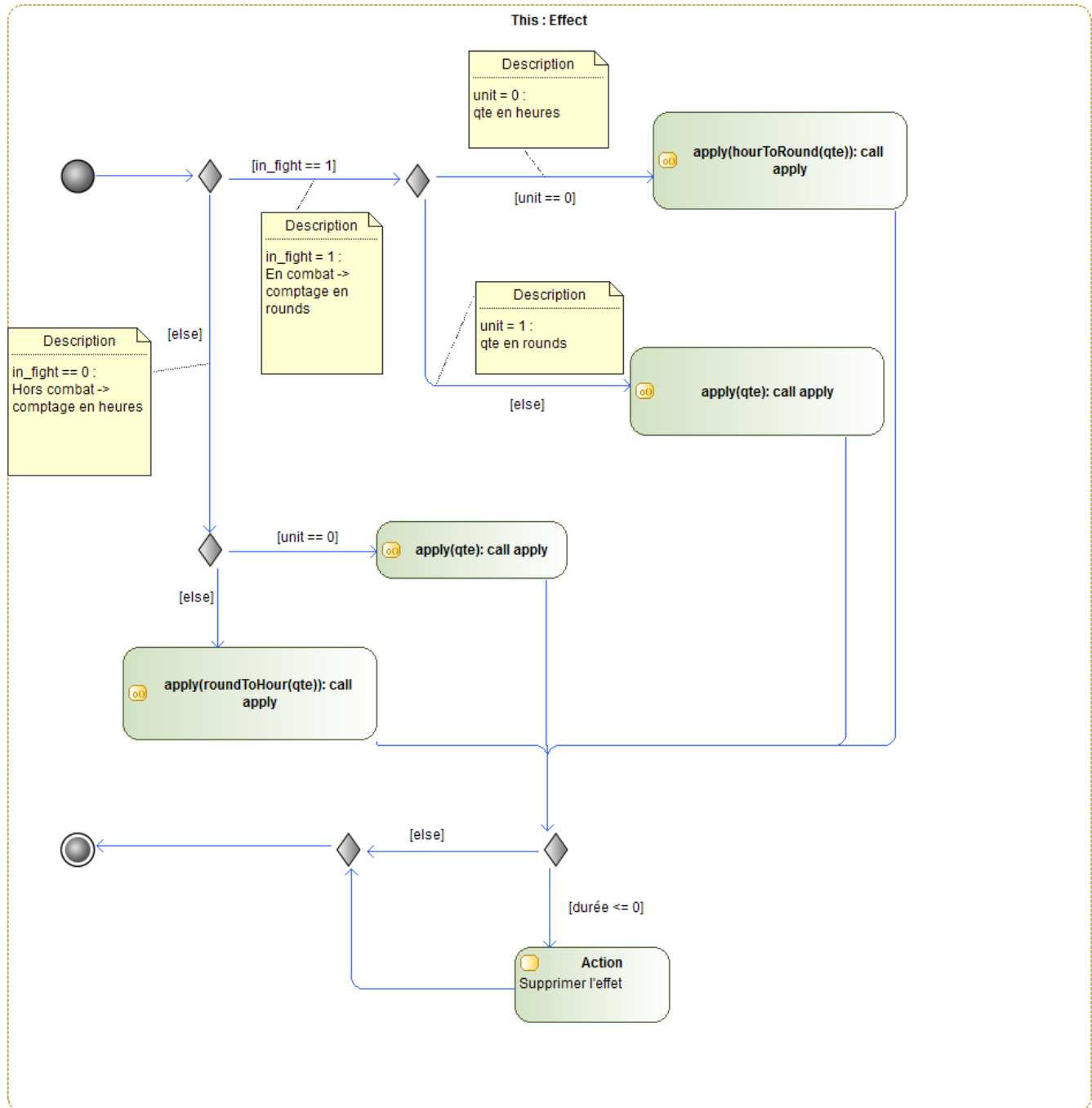
A la fin de toutes les actions, le programme retourne dans l'état d'attente.



## 6~ Diagrammes d'activité

ACT: Live

Diagramme d'activité de la méthode live(in fight, qte) de la classe Effect



---

Ce diagramme a pour but d'expliquer le fonctionnement de la méthode «live()» de la classe «Effect». Cette méthode sera appelée dès qu'un effet est amené à changer: les effets dans Donjons et Dragons (D&D) sont tous temporaires. Leur durée s'exprime en **rounds** ou en **heures**. Il est nécessaire de décrémenter ces effets en fonction de la situation: en effet, dans D&D on peut considérer 2 moyens de compter le temps qui passe.

~ Le temps est compté **en round** quand les joueurs sont **en combat**. En effet, les combats dans D&D se déroulent au tour par tour mais l'on considère que leur résolution est simultanée. Tous les joueurs peuvent agir de manière limitée (Ces limitations sont expliquées dans le livre de règles et leur description ne nous intéressent pas ici. On considère que la plage d'action d'un personnage est égale à **6 secondes** environ) et une fois que tous les joueurs et ennemis ont joué dans un ordre précis déterminé au début du combat, on considère qu'un round est passé, et on recommence pour le round suivant. Les rounds s'enchaînent jusqu'à fin du combat (tous les ennemis sont hors d'état de nuire, ou fuite des joueurs).

~ Le temps est compté **en heure** quand les joueurs sont **hors combat**. En temps normal, les joueurs sont libres de faire ce qu'ils veulent, et l'estimation du temps de réalisation est donné par le MJ ou par les joueurs en fonction des situations (Exemple 1 -> Joueur1: «Je vais m'entraîner à manier mon bâton pendant 2h et passer l'après-midi à faire des recherches à la bibliothèque (4h)» / Exemple 2 -> MJ: «A pied, le trajet de la ville A à la ville B dure 3h»)

Les effets peuvent être causés par des sorts lancés par des joueurs ou des ennemis, ou des raisons extérieures (feu, maladie...). Les durées des différents effets sont détaillées dans le livre des règles.

Il peut alors arriver un moment où un effet qui dure plusieurs heures est appliqué en combat. (Exemple: En début de combat, un joueur lance le sort de «vision dans le noir», qui dure 4h») Il est alors important d'effectuer des conversions en fonction de la situation.

Chaque effet est une instance de la classe «Effect», et comporte alors des attributs tels que sa durée (duration) ainsi que son unité (unit == 0: heures / unit == 1: rounds)

La méthode «live()» comporte deux arguments d'entrée:

- ~ in\_fight: une variable booléenne qui indique la situation actuelle (in\_fight ==1: les joueurs sont en combat).
- ~ qte: la quantité à décrémenter.

La méthode commence donc par vérifier si les joueurs sont en combat, puis va regarder l'unité de l'effet. 4 cas de figure sont possibles:

- 1) Joueurs en combat, unité de l'effet en round: on applique la méthode «apply()», qui décrémente le nombre de rounds (qte) à la durée de l'effet (duration).
- 2) Joueurs en combat, unité de l'effet en heures: il est nécessaire de convertir la durée de l'effet en round, pour pouvoir décrémenter du nombre de rounds (qte). On utilise la méthode «hourToRound()».
- 3) Joueurs hors combat, unité de l'effet en heure: pas de conversion nécessaire, on applique la méthode «apply()».
- 4) Joueurs hors combat, unité de l'effet en rounds: conversion nécessaire avec la méthode «roundToHour()» afin de décrémenter la durée de l'effet avec «apply()».

Une fois un des quatre cas de figure effectué, on regarde la durée restante de l'effet (duration). Si elle est inférieure à 0, on **supprime** l'effet. Dans le cas contraire, la durée restante est enregistrée et on attend le prochain appel de la méthode «live()».



---

## Les outils de développement



# Qui sommes-nous?



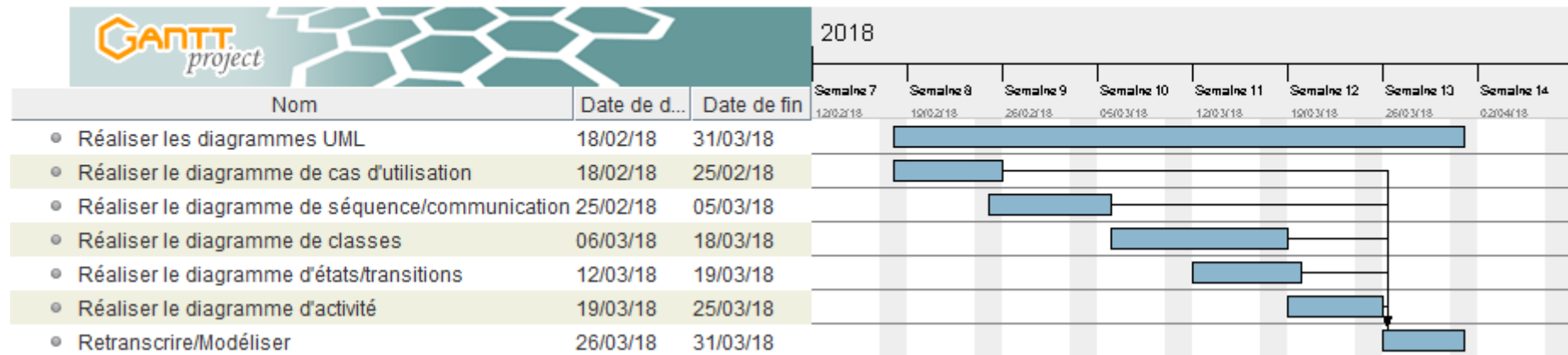
De gauche à droite

Nom	email	CTD	Labo	responsable de projet
Jean-Loup MONNIER	j5monnie@enib.fr	B	3	✓
Loïck LE COZ	l5lecoz@enib.fr	B	3	
Arthur EPAUD	a5epaud@enib.fr	B	2	
Louis OLIVIER	l5olivie@enib.fr	B	3	

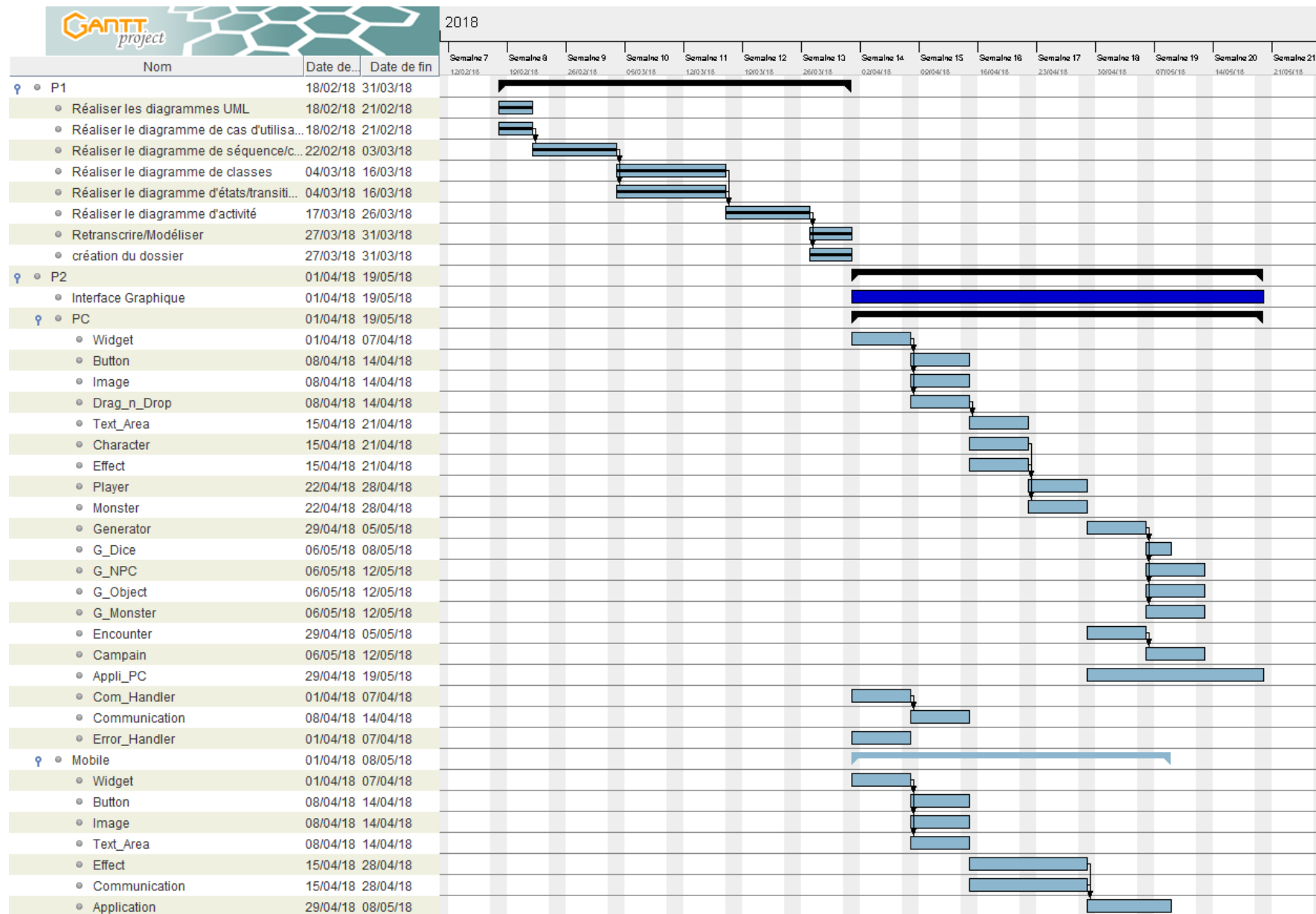


# Gantt

1 ~ Prévisionnel P1



## 2 ~ Définitif P1 & Prévisionnel P2



# Répartition des tâches

	Début P1	Fin P1	P2			Classe
Jean-Loup	Diagramme papier	Diagramme papier Création Dossier	Ensemble des classes de l'Appli Mobile		Campain Appli_PC	
			Interface Graphique			
Louis	Diagramme papier	Diagramme papier Diagramme Modelio	DragndDrop Text_Area	Character Effect	Campaign Appli_PC	
			Interface Graphique			
Loïck	Diagramme papier	Diagramme papier Création dossier	Widget Button Image	Player Monster	Generator G_Dice G_NPC G_Object	
			Interface Graphique			
Arthur	Diagramme papier descriptions textuelles	Diagramme papier Diagramme Modelio	Com_Handler communication	Encounter	Campain Appli_PC	
			Interface Graphique			





---

# Conclusion

RPG Helper c'est donc avant tout un projet utile et pratique pour tous les joueurs de jeux de rôle : Le Maître du Jeu habitué retrouvera rapidement les différents aspects du jeu dont il a l'habitude via des outils faciles à utiliser et centralisés sur un seul et unique logiciel PC, et le joueur néophyte pourra plus aisément trouver ses marques et appréhender le fonctionnement des jeux de rôle via une interface compréhensible sur son téléphone. Dans sa première version, RPG Helper sera centré sur le jeu de rôle Donjons & Dragons 3.5, mais notre but est avant tout de concevoir le fonctionnement du "squelette" de l'application. Celui-ci sera par la suite capable d'héberger différents univers de jeu, afin que de plus en plus de rôlistes de par le monde puissent vivre leurs aventures épiques de manière simple et pratique.

Sur ce, bon jeu, et comme dirait César : "Alea jacta est !"

