

Win-Win

# Compte-Rendu de stage

1 er Juillet 2022 – 31 Décembre 2022

Kenan Blasius

# Table des matières :

I.	Présentation de l'entreprise.....	2
II.	Les différents projets.....	4
1.	Recherche de solutions métavers .....	5
A.	Présentation du projet .....	5
B.	Résultat.....	5
C.	Conclusion .....	5
2.	Apprentissage de Mongoose et Express.js (Bot Discord).....	7
A.	Présentation du projet .....	7
B.	Structure du projet, problèmes et facilités rencontrés.....	7
C.	Résultat.....	10
3.	Entraînement Mongoose (Bot Discord).....	11
A.	Présentation du projet .....	11
B.	Structure du projet, problèmes et facilités rencontrés.....	11
C.	Résultat.....	13
4.	Apprentissage de Socket.io (Système de chat) .....	14
A.	Présentation du projet .....	14
B.	Structure du projet, problèmes et facilités rencontrés.....	14
C.	Résultat.....	14
5.	Apprentissage de GitHub Pages (Génération de documentation).....	16
A.	Présentation du projet .....	16
B.	Structure du projet, problèmes et facilités rencontrés.....	16
C.	Résultat.....	17
6.	Management app (Outil de gestion interne).....	18
A.	Présentation du projet .....	18
B.	Structure du projet, problèmes et facilités rencontrés.....	18
C.	Résultat.....	19
7.	Unreal engine 4.27 (« The Binding of Isaac : Afterbirth + » en Blueprint) .....	22
A.	Présentation du projet .....	22
B.	Structure du projet, problèmes et facilités rencontrés.....	23
C.	Résultat.....	25
III.	Conclusion .....	26

# I. Présentation de l'entreprise

**Win-Win.com** (à ne pas confondre avec winwin.com) **existe depuis mai 2007.**

Elle regroupe une cinquantaine de « **Winners** » répartis sur deux sites : **Paris et Marseille.**

**Win-Win est une agence de communication événementielle qui conçoit et produit des expériences en live, en E-live ou en Link** (Organisation de conventions, de congrès, de salons, de stands...).

**Par exemple, Win-Win est présent dans l'organisation du #BigTour, festival de l'entreprise France et de la fierté du fabriqué en France** (Ou festival des entrepreneur).

**Le pôle de Paris est plutôt axé sur l'organisation des événements alors que le pôle de Marseille travaille principalement sur la production des plateformes digitales.**

**La mascotte de Win-Win est un gros caneton jaune que vous pouvez croiser à l'occasion sur les routes française.**

**J'ai effectué mon stage à Win-Win Marseille en tant qu'apprenti développeur full stack et Unreal Engine.**



## II. Les différents projets

## 1. Recherche de solutions métavers

### A. Présentation du projet

Dès mon arrivé dans l'entreprise, on m'a demandé de rechercher différents métavers fonctionnant sur navigateur internet.

Ma recherche devait servir à trouver une alternative moins onéreuse pour le client que la solution actuellement proposée par Win-Win afin de rendre le service plus accessible.

### B. Résultat

J'ai donc réussi à trouver une multitude de métavers fonctionnant sur navigateur internet, tel que :  
-Spatial.io



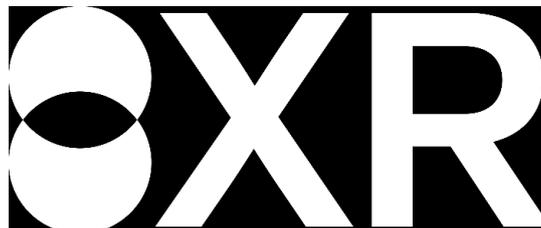
-Framevr.io



-Yabal.io



-8xr



(Liste non exhaustive)

### C. Conclusion

Win-Win étudie actuellement la possibilité d'inclure une offre utilisant Spatial.io qui propose différentes spécificités intéressantes, telles que :

- Fait avec Unity (le système de chat vocal est photon permettant une communication de proximité ou globale)
- Prix : 25€/mois pour un compte pro (recommandé pour les organisateurs)
- Fonctionne sur :
  - Navigateur Web

- Mobile (Android/IOS)
- VR (Oculus Quest/ Oculus Quest 2)
- Pas plus de 50 personnes par instance
- Pas de limitation du nombre de monde
- Système de portail afin de changer de monde
- Les participants doivent avoir un compte pour pouvoir interagir
- Possibilité de faire des instances privées (lien envoyé par mail)
- Tout est déjà paramétré, il suffit d'importer le monde fait en 3D
- Possède un système d'administration (compte pro uniquement)
- Les utilisateurs peuvent mettre des « post-it » afin de poser des questions qui seront répondu plus tard
- Possibilité de partager son écran ou des fichiers en direct

## 2. Apprentissage de Mongoose et Express.js (Bot Discord)

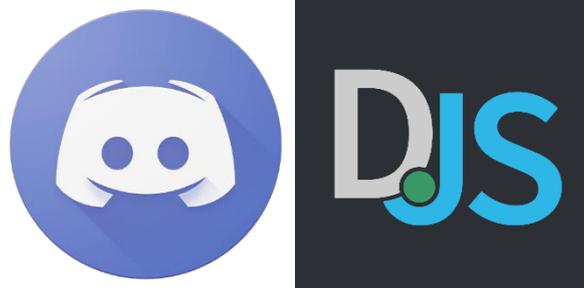
### A. Présentation du projet

**Par la suite, on m'a demandé d'apprendre à utiliser Express.js** (Express.js est un framework permettant de faire des applications web avec Node.js, le langage de programmation utilisé pour ce projet).

**Il m'a aussi été demandé de travailler sur MongoDB** (MongoDB est un système de gestion de base de données) **à l'aide de Mongoose** (Mongoose est une bibliothèque de programmation permettant d'utiliser MongoDB avec Node.js).



**Connaissant déjà l'API** (« Application Programming Interface » ou « interface de programmation d'application ») **de Discord, en accord avec mon chef de projet, j'ai proposé de faire un bot Discord de gestion de serveur en intégrant Express.js et Mongoose, afin d'apprendre à les utiliser.**



**Le but de ce bot discord est donc de pouvoir gérer un serveur discord en permettant d'avertir, d'exclure ou de bannir temporairement ou définitivement un membre**

(L'avertissement/l'exclusion/le bannissement sera stocké dans la base de données à l'aide de Mongoose pour une consultation ultérieure).

**Il a également comme objectif de pouvoir créer des concours, des événements ou des sondages**

(Les données, comme la date de début et de fin du concours/événement/sondage par exemple, seront également stocké dans la base de données).

**Il possède aussi un système d'expérience qui augmente en parlant sur le serveur ou en étant**

**présent dans un salon vocal** (L'expérience gagné par chaque utilisateur sera stockée dans la base de données et un classement sera disponible via un site fait à l'aide de Express.js).

### B. Structure du projet, problèmes et facilités rencontrés

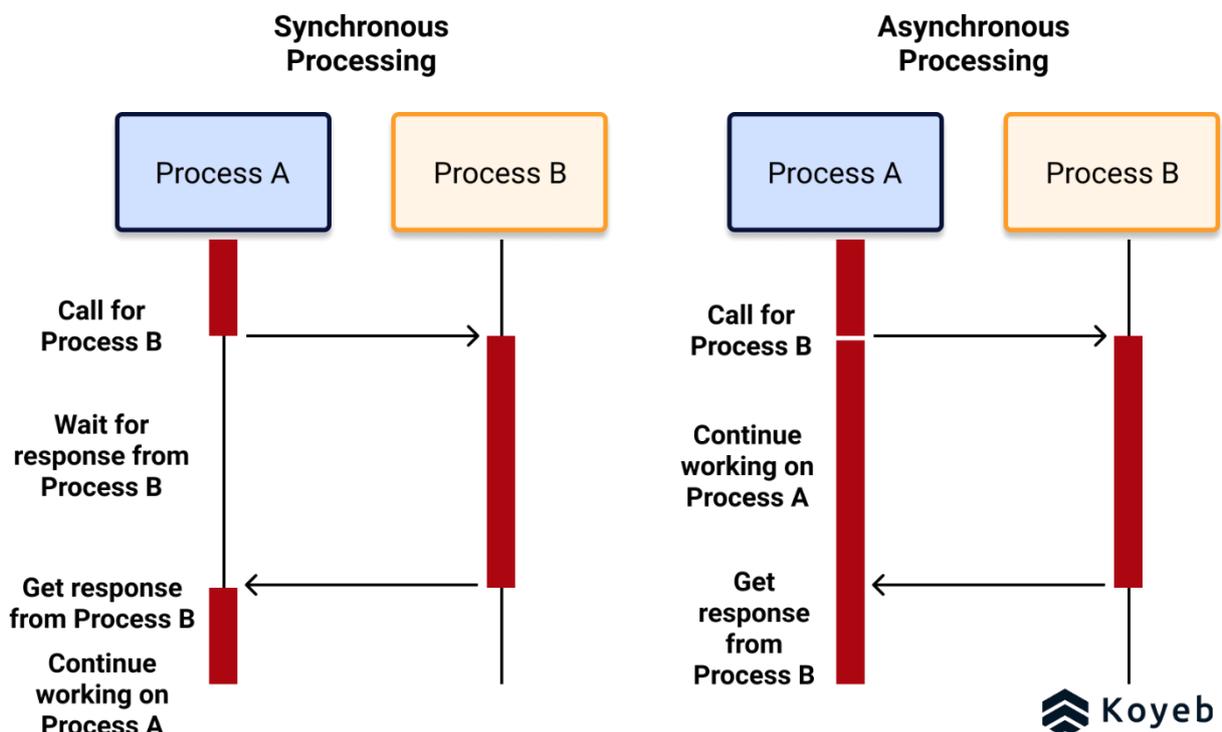
**J'ai donc dans un premier temps interprété les messages des différents utilisateurs afin de savoir ce qu'ils souhaitent faire avec le bot, ensuite j'ai commencé à me renseigner sur le fonctionnement de Mongoose, j'ai principalement appris à enregistrer et à lire les données provenant de ma base de données.**

J'ai rencontré une première difficulté en utilisant Mongoose, les actions sur la base de données se font de manière asynchrone.

Un code asynchrone exécutera plusieurs tâches en simultanée alors qu'un code synchrone (ce que j'avais pour habitude de faire) exécute une tâche seulement après avoir fini la précédente.

J'ai donc dans un premier temps essayé de forcer le code à attendre que les actions de la base de données soient finies avant d'exécuter la suite du code mais cela n'a pas fonctionné.

En recherchant sur internet, j'ai trouvé comment coder avec du code asynchrone il suffit de mettre son code dans une fonction qui sera exécutée par la méthode « `.then()` » (Une méthode, en programmation informatique, est une fonction qui s'applique à une variable, et dans le cas de l'exécution de code asynchrone une méthode « `.then()` » et une méthode « `.catch()` » s'ajoute afin d'exécuter du code une fois la tâche finie en cas de réussite, méthode « `.then()` », ou en cas d'échec ou erreur, méthode « `.catch()` »).



Par la suite j'ai commencé à implémenter mes différentes fonctionnalités afin de m'entraîner sur :

- L'enregistrement de tous les avertissements/exclusions/bannissements ainsi que les concours/événements/sondages

- L'ajout d'expérience à l'utilisateur à chacun de ses messages dépendant de la taille du message

- L'ajout d'expérience à l'utilisateur par minute resté dans un salon vocal bien que ce dernier ajout fût plus compliqué puisqu'il m'a fallu me servir des timestamp (« horodatage » en français) afin de calculer le temps passé dans le salon vocal.

Puis j'ai commencé à apprendre à me servir de Express.js.

Je n'ai pas eu de difficulté à utiliser Express.js cependant il fallait que j'apprenne à faire un site web et donc apprendre le HTML (« HyperText Markup Language » est un langage de balisage utilisé pour la création de page web).

## Site avec HTML :

Voici le leaderboard du serveur VRCFR:



#1	CCORSSAIRE#2000	Niveau: 57	402997 XP
#2	Pomelos#5473	Niveau: 48	250541 XP
#3	rx#3285	Niveau: 47	226016 XP
#4	ememem_#9799	Niveau: 44	186720 XP
#5	leobreuil#3302	Niveau: 42	167658 XP
#6	Elite ♀#1117	Niveau: 41	159361 XP
#7	TrOis ♀ #0003	Niveau: 40	149372 XP
#8	Pak#1995	Niveau: 37	124381 XP
#9	Kika#9338	Niveau: 38	123886 XP
#10	Nastonium#3508	Niveau: 38	122108 XP

Puis il m'a également fallu apprendre le CSS (« Cascading Style Sheet ») afin de mettre en forme la page que je viens de générer.

Site avec CSS :

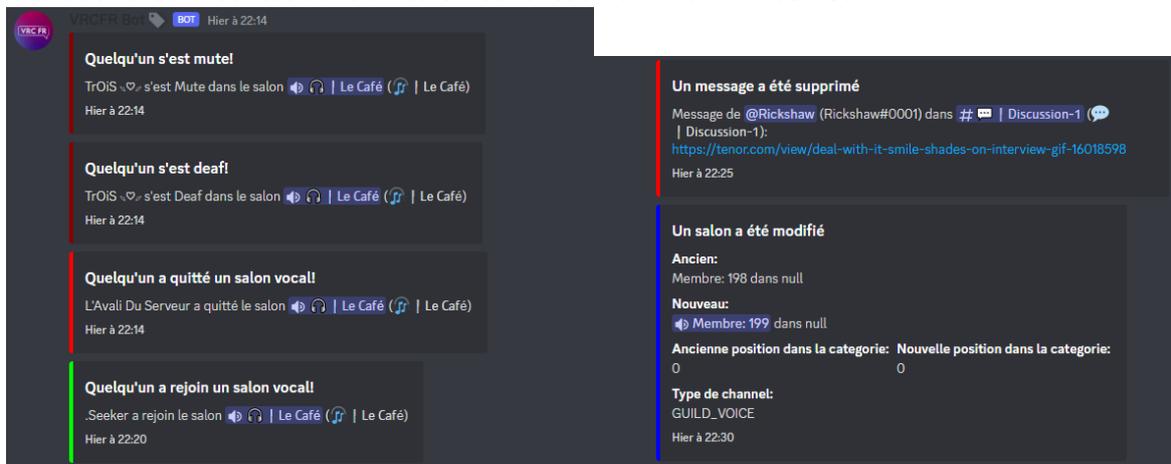


Voici le leaderboard du serveur VRCFR:			
#1	CCORSSAIRE#2000	Niveau: 57	402997 XP
#2	Pomelos#5473	Niveau: 48	250541 XP
#3	rx#3285	Niveau: 47	226016 XP
#4	ememem_#9799	Niveau: 44	186720 XP
#5	leobreuil#3302	Niveau: 42	167658 XP
#6	Elite ♀#1117	Niveau: 41	159361 XP
#7	TrOis ♀ #0003	Niveau: 40	149372 XP
#8	Pak#1995	Niveau: 37	124381 XP
#9	Kika#9338	Niveau: 38	123886 XP
#10	Nastonium#3508	Niveau: 38	122108 XP
#11	-Taci-#9613	Niveau: 35	107454 XP
#12	RamZu#9259	Niveau: 35	107064 XP
#13	wynycya#0002	Niveau: 35	106621 XP
#14	Starquad#8556	Niveau: 34	94930 XP
#15	RemsLeVras#0001	Niveau: 34	92954 XP
#16	Romi#1210	Niveau: 32	85358 XP

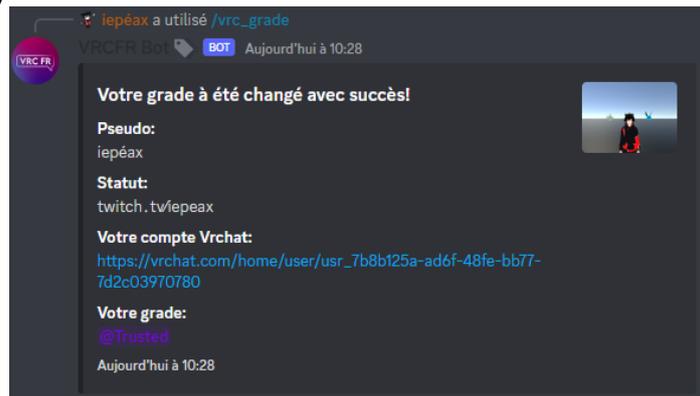
Suite à la demande des utilisateurs du bot, j'ai ajouté un système de création de salon vocal et écrit temporaire, lorsqu'un utilisateur rejoint un salon vocal particulier, un nouveau salon vocal ainsi qu'un nouveau salon écrit sont créés et seront supprimés lorsque le salon vocal créé ne contiendra plus aucun utilisateur.

La personne à l'origine de ces deux salons peut utiliser des commandes avec le bot afin d'apporter des modifications au salon vocal.

J'ai également ajouté un système de log avancé permettant de voir toutes les actions faites sur le serveur, notamment les actions non enregistrées par Discord par défaut comme la suppression des messages ou les utilisateurs qui se connectent ou déconnectent des salons vocaux, toutes les actions sont retranscrites dans un salon écrit réservé au staff du serveur.



Ce bot étant sur un serveur d'une communauté VRChat, j'ai également créé une commande permettant d'obtenir son grade VRChat sur le discord en utilisant le lien menant au profil de l'utilisateur sur le site officiel en utilisant une API faite par la communauté me permettant de récupérer les informations d'un joueur à partir de son ID (L'ID est récupéré grâce au lien menant au profil de l'utilisateur).



### C. Résultat

Au cours de ce projet, j'ai réalisé un bot Discord permettant de gérer un serveur et j'ai également appris à utiliser Mongoose ainsi que Express.js.

### 3. Entraînement Mongoose (Bot Discord)

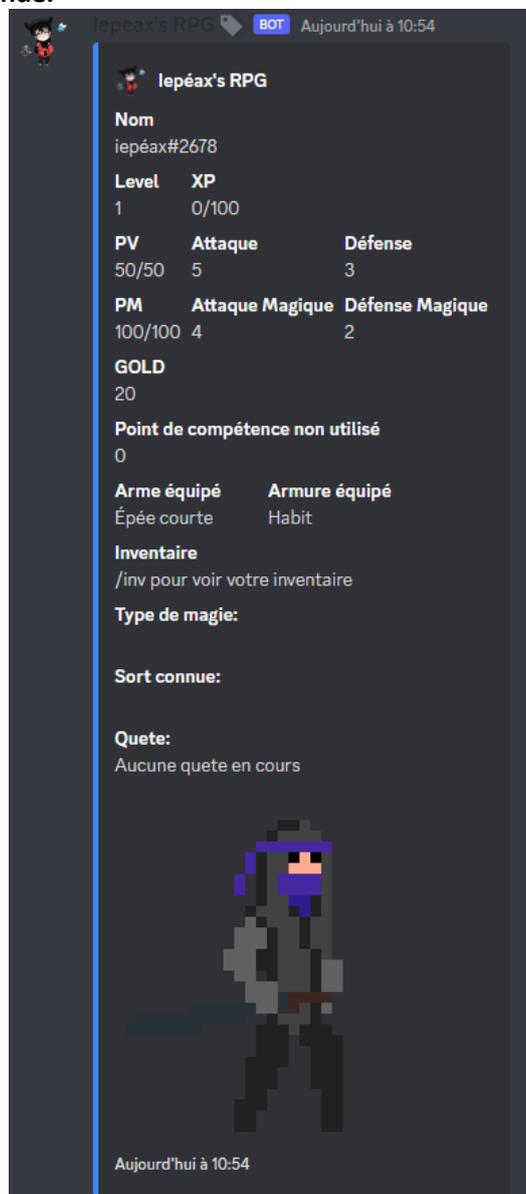
#### A. Présentation du projet

Mon maître de stage, n'ayant pas de projets sur lesquelles je puisse intervenir, a décidé que je continuerais à me former avec Mongoose en créant un nouveau bot discord de RPG (« Role-Playing Game » ou « Jeu de Rôle ») en Node.js avec discord.js.

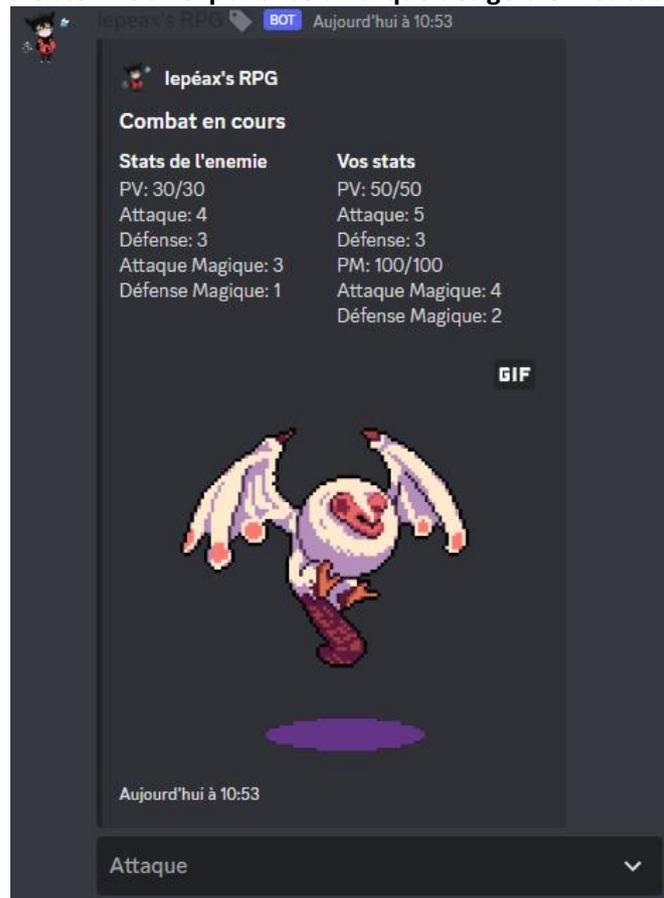
#### B. Structure du projet, problèmes et facilités rencontrés

Comme dans le précédent projet, je commence par interpréter chaque message des utilisateurs, afin de savoir ce que ces derniers souhaitent faire avec le bot.

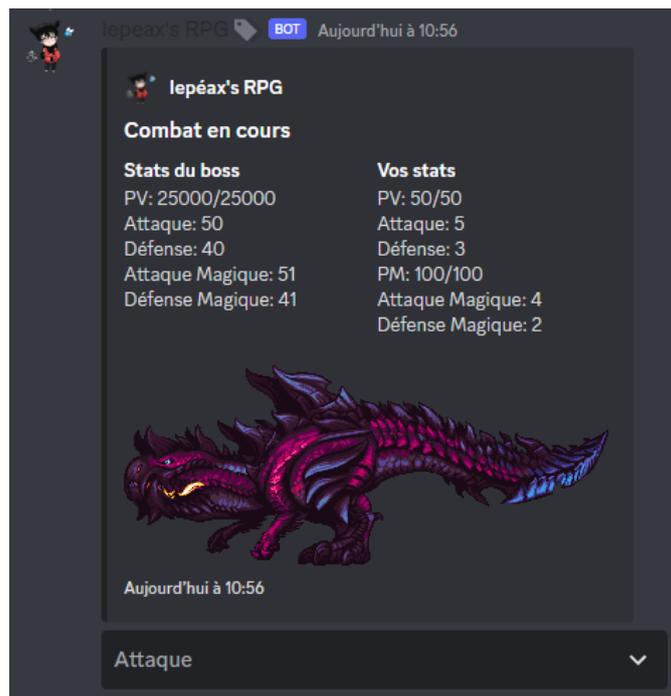
Puis je commence par créer la page du joueur avec de l'expérience, des niveaux, de la vie, de l'attaque, de la défense, des gold (La monnaie du RPG), de l'attaque magique, de la défense magique, des points de magie, un type de magie, une liste des sorts connus, des points de compétence, une arme, une armure, un inventaire et la quête en cours si le joueur en possède une. J'ai également ajouté un gif à la page du joueur afin de permettre au joueur de personnaliser son profil à l'aide d'une commande.



J'ajoute ensuite des ennemis, qui possède une force proportionnelle au niveau du joueur, permettant au joueur d'obtenir de l'expérience ainsi que des gold en les tuants.



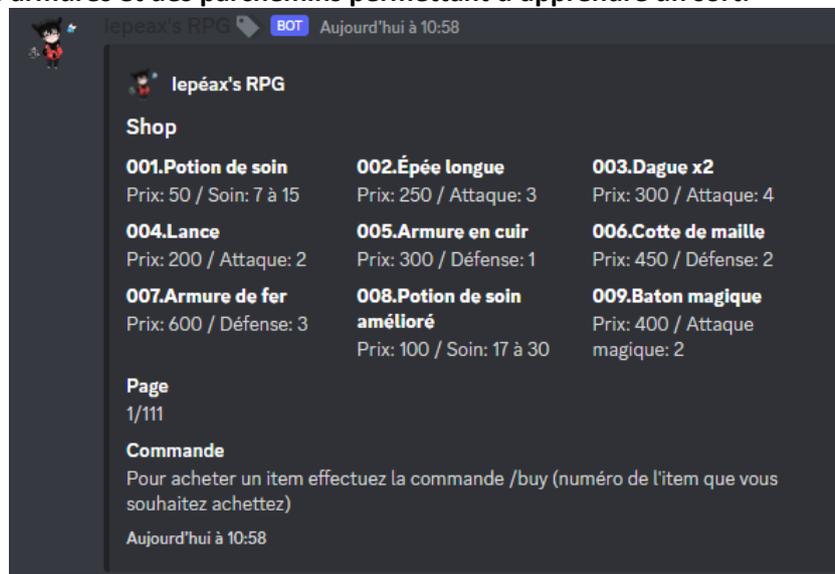
Chaque ennemie ne pouvant être attaqué que par le joueur auquel ils sont liés, j'ai créé un boss qui peut être attaqué par tous les joueurs mais qui commence avec l'équivalent d'un ennemi de niveau dix en force avec énormément de vie afin que tout le monde puisse l'attaquer et obtenir une partie de ses récompenses à sa mort, lorsque le boss meurt un nouveau boss plus puissant prends sa place.



J'ai également créé des commandes utilisables toutes les 4 ou 6 heures, en fonction de la commande, afin de permettre au joueur d'obtenir de l'expérience sans avoir à tuer des ennemis.



J'ai ensuite fait des objets achetables dans le magasin, comme des potions de soins ou de magie, des armes, des armures et des parchemins permettant d'apprendre un sort.



J'ai créé une commande permettant de choisir son type de magie parmi le feu, le vent, l'eau, la terre, la glace, la foudre, la lumière et les ténèbres.

Ensuite, j'ai créé les sorts avec leur type de magie, les sorts ne permettent que d'infliger plus de dégâts.

Puis j'ai commencé à créer des quêtes telles qu'éliminer des monstres ou infliger des dégâts au boss pour recevoir de l'expérience et des gold.

Toutes les données telles que les caractéristiques des joueurs, des ennemis, du boss, les sorts, les objets et les quêtes sont stockées dans la base de données à l'aide de Mongoose.

### C. Résultat

J'ai donc créé un RPG textuel sur Discord où il est possible de se battre contre des ennemis et des boss afin de gagner de l'expérience et de se battre contre des ennemis encore plus forts.

Ce projet m'a permis d'être autonome dans l'utilisation Mongoose.

#### 4. Apprentissage de Socket.io (Système de chat)

##### A. Présentation du projet

**On m'a également demandé d'apprendre à me servir de Socket.io** (Socket.io est une bibliothèque de programmation événementielle permettant une communication bidirectionnelle en temps réel entre le serveur et le navigateur internet).



**J'ai codé un système de conversation entre utilisateur en temps réel en Node.js avec Mongoose, que j'ai appris au cours de l'un de mes précédents projets, afin de stocker les messages et les conserver même après s'être déconnecté puis reconnecté.**

##### B. Structure du projet, problèmes et facilités rencontrés

**Je commence d'abord par suivre l'exemple fournit sur la page officiel de Socket.io et y ajoute quelques fonctionnalités telles que la possibilité de choisir un nom d'utilisateur en le tapant dans la barre de recherche** (Exemple : « localhost:3000/utilisateur1 » est le lien utilisé pour être connecter en tant que « utilisateur1 » alors que « localhost:3000/utilisateur2 » est le lien utilisé pour être connecter en tant que « utilisateur2 ») **ainsi que le stockage des messages pour ne pas perdre l'historique des messages, mais j'ai un problème, lorsqu'un utilisateur se connecte, tous les autres utilisateurs reçoivent à nouveau tous les anciens messages envoyé dans la conversation depuis sa création.**

**Afin de corriger ce bug, lorsqu'un utilisateur se connecte je récupère son nom d'utilisateur et lors de l'envoi des messages j'indique que les messages sont destinés à cet utilisateur uniquement. De cette manière seul l'utilisateur qui vient de se connecter traite les messages reçus.**

**À la suite de ce bug, j'ai eu l'idée de faire un système de « messages privés », c'est-à-dire que si trois utilisateurs sont présents dans la conversation mais que l'utilisateur 1 souhaite parler seulement à l'utilisateur 2 sans que l'utilisateur 3 ne voit les messages, c'est possible.**

**Pour cela il lui suffit d'écrire le nom de l'utilisateur suivi de son message afin que seul l'utilisateur ciblé le reçoive.**

**Puis j'ai ajouté la date d'envoi du message afin d'avoir une temporalité ainsi que la liste des utilisateurs actuellement connectés.**

##### C. Résultat

**Voilà donc à quoi ressemble mon système de conversation entre utilisateur, fait à l'aide de Socket.io et Mongoose.**

**Vu de la liste des utilisateurs connectés :**



### Vu de la conversation par utilisateur1 :

**utilisateur1** le 9/12/2022 à 13:29:53  
bonjour!

**utilisateur2** le 9/12/2022 à 13:30:03  
bonjour! ça va?

**utilisateur3** le 9/12/2022 à 13:30:12  
ça va et vous?

**Message privée envoyé à utilisateur2** le 9/12/2022 à 13:30:39  
tu connais utilisateur3?

**Message privée reçu de utilisateur2** le 9/12/2022 à 13:30:52  
non

### Vu de la conversation par utilisateur2 :

**utilisateur1** le 9/12/2022 à 13:29:53  
bonjour!

**utilisateur2** le 9/12/2022 à 13:30:03  
bonjour! ça va?

**utilisateur3** le 9/12/2022 à 13:30:12  
ça va et vous?

**Message privée reçu de utilisateur1** le 9/12/2022 à 13:30:39  
tu connais utilisateur3?

**Message privée envoyé à utilisateur1** le 9/12/2022 à 13:30:52  
non

### Vu de la conversation par utilisateur3 :

**utilisateur1** le 9/12/2022 à 13:29:53  
bonjour!

**utilisateur2** le 9/12/2022 à 13:30:03  
bonjour! ça va?

**utilisateur3** le 9/12/2022 à 13:30:12  
ça va et vous?

### Vu globale de la page :

<b>utilisateur1</b> le 9/12/2022 à 13:29:53 bonjour!	<a href="#">Utilisateurs connectés</a> utilisateur1 utilisateur2 utilisateur3 <a href="#">Connecté en tant que</a> utilisateur1
<b>utilisateur2</b> le 9/12/2022 à 13:30:03 bonjour! ça va?	
<b>utilisateur3</b> le 9/12/2022 à 13:30:12 ça va et vous?	
<b>Message privée envoyé à utilisateur2</b> le 9/12/2022 à 13:30:39 tu connais utilisateur3?	
<b>Message privée reçu de utilisateur2</b> le 9/12/2022 à 13:30:52 non	
<input type="text"/>	<input type="button" value="Envoyer"/>

## 5. Apprentissage de GitHub Pages (Génération de documentation)

### A. Présentation du projet

Ensuite on m'a demandé d'apprendre à utiliser GitHub Pages afin de générer la documentation des différents projets plutôt que d'avoir à recréer un site à chaque projet juste pour la documentation. GitHub Pages est un service d'hébergement de site internet qui récupère les fichiers HTML, CSS et JS présent dans le dépôt GitHub ou qui peut générer ces fichiers à l'aide de GitHub Action (GitHub Action est un système permettant d'exécuter du code en fonction des actions faite sur le dépôt GitHub).



### B. Structure du projet, problèmes et facilités rencontrés

**J'ai donc commencé par apprendre à configurer un dépôt GitHub** (Un dépôt GitHub contient tous les fichiers d'un projet ainsi que l'historique de chaque fichier. Un dépôt peut être public, limitée à une organisation ou privée).

**Afin de pouvoir utiliser GitHub Pages, j'apprends que pour se servir de GitHub Pages il faut avoir un dépôt public ou appartenant à une organisation, ne pouvant donc pas faire de dépôt privé, je fais un dépôt public afin d'apprendre, je le configure en suivant la documentation et le site est maintenant disponible, cependant ce site a été fait par moi-même et non pas généré par GitHub comme je le souhaite.**

**Je continue donc de suivre la documentation de GitHub où il est écrit qu'avec Jekyll on peut générer notre site à partir de Markdown** (Markdown est un langage de balisage plus simple à écrire que le HTML) **cependant même en suivant la documentation, je ne parviens pas à générer de site avec Jekyll.**

**Je cherche des solutions et n'en trouve aucune qui règle mon problème mais j'ai appris qu'il était possible d'utiliser d'autre générateur que Jekyll.**

**J'ai trouvé MkDocs qui est aussi un générateur de site qui de plus est pensé pour créer des documentations.**

**Je suis donc la documentation et je parviens à générer mon site du premier coup, par la suite il me suffisait juste de modifier le fichier de configuration et d'écrire mes différentes pages en Markdown pour pouvoir générer une documentation complète sans avoir à utiliser de HTML ou de CSS.**

**Et pour que ce soit réutilisable plus facilement j'ai créé une page regroupant toutes les balises du Markdown utilisable avec MkDocs afin de ne pas avoir à les rechercher sur internet en cas d'oubli.**

## C. Résultat

Voici donc à quoi ressemble la page regroupant toutes les balises Markdown utilisable avec MkDocs avant la génération (en haut, le fichier source en Markdown) et après la génération (en bas, le site internet généré par MkDocs).

```
# Titre (h1)

## Titre (h2)

### Titre (h3)

#### Titre (h4)

##### Titre (h5)

<!-- Commentaire en markdown -->

## Text formatting

Du texte

==Du texte surligné==

^^Du texte souligné^^

~~Du texte barré~~

**Du texte en gras**

*Du texte en italique*

Du texte ~subscripted~

Du texte ^superscripted^

Touche de clavier: ++ctrl++ ou ++ctrl+shift+esc++

Du texte peut être {--suprimé--} et {++ajouté++}. Cela peut aussi être combiné en {~un~} une seule opération. {==Surligné==} est aussi possible {>} et des commentaire peuvent être ajouté<<}.

{=}

Cela peut aussi s'appliquer au bloc en mettant la balise d'ouverture et de fermeture sur d'autre ligne.

==}
```

The screenshot shows a web browser displaying a MkDocs site. The header is blue and contains the text 'REPO\_NAME\_OR\_SITE\_NAME' and a search bar with the placeholder 'Rechercher'. Below the header, there are navigation links: 'Home', 'Simple File', 'Markdown Demo', and 'Simple Folder'. The main content area is dark blue and displays the rendered output of the Markdown source code. It includes a table of contents on the right side with links to various sections: 'Titre (h2)', 'Titre (h3)', 'Titre (h4)', 'Titre (h5)', 'Text formatting', 'Note blocks', 'Different type de note', 'Code blocks', 'Content tab', 'Data Tables', 'Listes', 'Diagram', and 'Footnotes'. The main content area shows the rendered HTML for the source code, including headings, text formatting, and a table of contents on the right side.

Présentation faite à mon chef de projet pour une éventuelle utilisation ultérieure par Win-Win.

## 6. Management app (Outil de gestion interne)

### A. Présentation du projet

**Pour ce projet, on m'a demandé de refaire l'UX/UI** (« User eXperience/User Interface » ou « expérience utilisateur/interface utilisateur ») **du site de gestion interne des employés, des appareils, des licences et des clés cryptographiques de l'entreprise possédant aussi un système de ticketing intégré** (Si un employé à un problème, il peut ouvrir un ticket ce qui créera une discussion avec un sujet, un statut, pour savoir où en est la résolution du problème, une importance, ainsi que la date du problème) **en suivant un exemple fait proposé par Win-Win.**

### B. Structure du projet, problèmes et facilités rencontrés

**Tout d'abord le site ayant été fait avec Nuxt.js** (Nuxt.js est un framework basé sur Node.js et Vue.js) (Vue.js est un framework fait pour construire des pages internet et permettant d'étendre le HTML) **pour le front end** (Le front end est la partie visible par l'utilisateur) **et PHP** (« PHP : Hypertext Preprocessor » est un langage de programmation) **avec MariaDB** (MariaDB est un système de gestion de base de données) **pour le back end** (Le back end est la partie qui gère le fonctionnement du site), **il m'a fallu apprendre à me servir de Nuxt.js et donc de Vue.js, mais bien que Vue.js étende le HTML, la syntaxe n'est pas exactement la même notamment l'ajout de certains attributs** (Un attribut en HTML permet de contrôler le comportement d'un élément).

**De plus pour ce site Tailwind CSS a été utilisé il a donc fallu que j'acquière des connaissances sur ce Framework.**

**Tailwind CSS est un framework CSS permettant de mettre facilement une page en forme en utilisant des classes** (Une classe en HTML permet d'apporter des modifications à tous les éléments comportant une même classe).



J'ai appris comment utiliser Nuxt.JS et Tailwind CSS ainsi que refaire la structure de la page afin de la faire correspondre à l'exemple fourni.

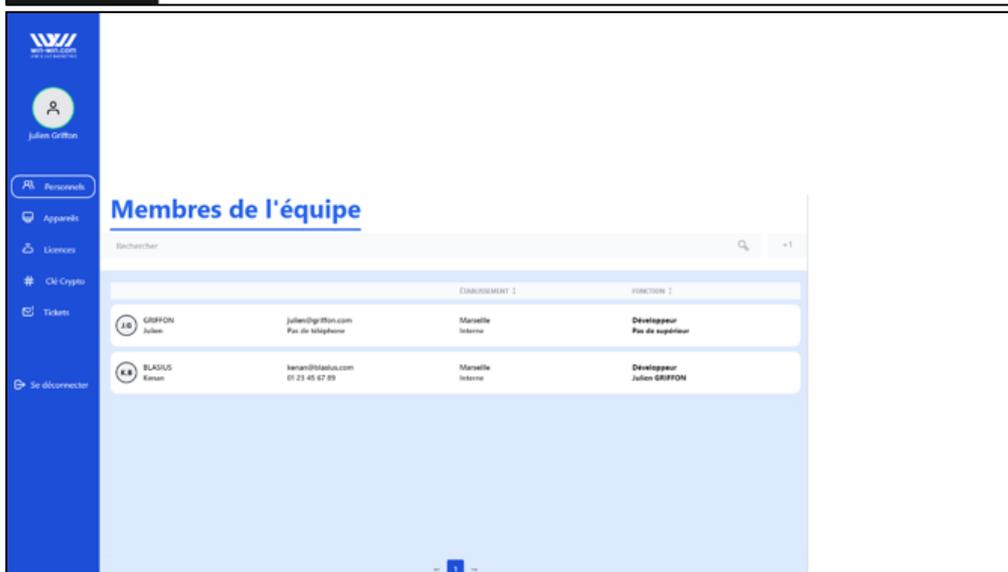
Puis on m'a demandé d'ajouter des informations pour les membres et les appareils.

J'ai remarqué qu'il me suffisait de modifier le fichier de création de la base de données en y ajoutant les nouvelles données que je souhaite stocker ou lire et le code PHP fera le lien entre les données envoyées et les données stockées.

Par exemple si je souhaite ajouter le numéro de téléphone dans ma base de données, il me suffit de créer « phone\_number » dans ma base de données et quand je souhaite lire ou stocker un membre de l'entreprise depuis le site, le front end, il me suffit de demander au back end les données concernant ce membre et je recevrai toutes les données y compris « phone\_number » sans avoir à modifier le code PHP.

### C. Résultat

Voici donc à quoi ressemble le site avant que je travaille dessus et après que j'ai travaillé dessus (Toutes les pages avec un thème noir et blanc sont les anciennes pages, toutes les pages avec un thème bleu et blanc sont les nouvelles pages) :



Dans la version précédente, afin d'avoir plus d'informations ou modifier une information d'un membre/appareil/etc... il fallait changer de page tant dis que maintenant tout s'effectue sur la même page.

This screenshot shows a form titled 'Informations personnelles' (Personal Information) with several input fields. The form is organized into sections: 'Informations personnelles', 'Informations du contrat' (Contract Information), and 'Informations d'accès' (Access Information). The user's name is 'Julien Griffon'.

Informations personnelles	
NOM*	kanan
PRENOM*	kanan
EMAIL*	kanan@blabla.com
NOUVEAU MOT DE PASSE	
CONFIRMATION DU NOUVEAU MOT DE PASSE	
Informations du contrat	
STATUT*	Interne
TYPE DE CONTRAT*	Stage
DATE D'ARRIVÉE*	01/07/2022
DATE DE DÉPART*	31/12/2022
POSTE*	Développeur
EMBAUSSEMENT*	Marseille
SURVEILLÉ	[Sujet Object]
ROLE*	Utilisateur
Informations d'accès	
VIRUS*	NON
SERVEUR*	

This screenshot shows the 'Membres de l'équipe' (Team Members) page. It features a search bar and a table listing team members. The table has columns for 'EMBAUSSEMENT 1' and 'FONCTION 1'. A 'Modifier' button is visible next to the entry for Julien Griffon.

	EMBAUSSEMENT 1	FONCTION 1	
J.G. GRIFFON Julien	Julien@griffon.com Pas de téléphone	Marseille Interne	Développeur Pas de supérieur
K.B. BLABLA Blabla	blabla@blabla.com Téléphone: 01 23 45 67 89	Marseille Interne	Profil complet Développeur Supérieur Julien GRIFFON

This screenshot shows the 'Membres de l'équipe' page with a detailed edit form for the selected member, Julien Griffon. The form includes fields for personal information, contract details, and access information. The 'Modifier' button has been replaced by 'Annuler' and 'Confirmer' buttons.

Informations personnelles	
NOM*	kanan
PRENOM*	kanan
EMAIL*	kanan@blabla.com
NOUVEAU MOT DE PASSE	01 23 45 67 89
CONFIRMATION DU MDP*	
Informations du contrat	
STATUT*	Interne
TYPE DE CONTRAT*	Stage
DATE D'ARRIVÉE*	01/07/2022
DATE DE DÉPART*	31/12/2022
POSTE*	Développeur
EMBAUSSEMENT*	Marseille
ROLE*	Utilisateur
Informations d'accès	
VIRUS*	NON
SERVEUR*	<input type="checkbox"/> AD <input type="checkbox"/> AZUR <input type="checkbox"/> BMS
COMPTES OFFICE	NON

Idem pour la création d'un nouveau membre/appareil/etc... il suffit de cliquer sur le « +1 » à droite de la barre de recherche afin de d'ouvrir le menu de création alors qu'il fallait ouvrir une nouvelle page.

This screenshot shows a form titled "Informations personnel" (Personal Information) with the following fields:

- PRENOM\***: Julien
- NOM\***: Dior
- EMAIL\***: julien@diior.com
- MÉTIER\***: Développeur
- MOT DE PASSE\***: [masked]
- CONFIRMATION DU MOT DE PASSE\***: [masked]
- INFORMATIONS DU CONTRAT**:
  - STATUT\***: [dropdown]
  - TYPE DE CONTRAT\***: [dropdown]
  - DATE D'EMBAUVE\***: [calendar icon]
  - POSTE\***: Développeur
  - ÉTABLISSEMENT\***: [dropdown]
  - ADRESSE\***: Rechercheur
  - ROLE\***: [dropdown]
- INFORMATIONS D'ACCÈS**:
  - VPN\***: [dropdown]
  - SERVEUR\***: [dropdown]
  - STRUCTURE\***:
    - AD
    - AZUR
    - JANS
    - OTHI Prod
    - Segur Prod
    - Segur Dev
    - Administratif
    - Administratif Global
    - Sécurité
    - Client

This screenshot shows the "Membres de l'équipe" (Team Members) page. At the top, there is a search bar with a "+1" button. Below it is the "Création d'un nouveau membre de l'équipe" (Create new team member) form with the following fields:

- NOM\***: Dior
- EMAIL\***: julien@diior.com
- ÉTABLISSEMENT\***: [dropdown]
- POSTE\***: Développeur
- PRENOM\***: Julien
- MÉTIER\***: Rechercheur
- MOT DE PASSE\***: [masked]
- CONFIRMATION DU MOT DE PASSE\***: [masked]
- INFORMATION CONTRAT**:
  - TYPE DE CONTRAT\***: [dropdown]
  - DATE D'EMBAUVE\***: [calendar icon]
  - ROLE\***: [dropdown]
- INFORMATIONS D'ACCÈS**:
  - VPN\***: [dropdown]
  - SERVEUR\***:
    - AD
    - AZUR
  - Compte Office ACTIF\***: [dropdown]

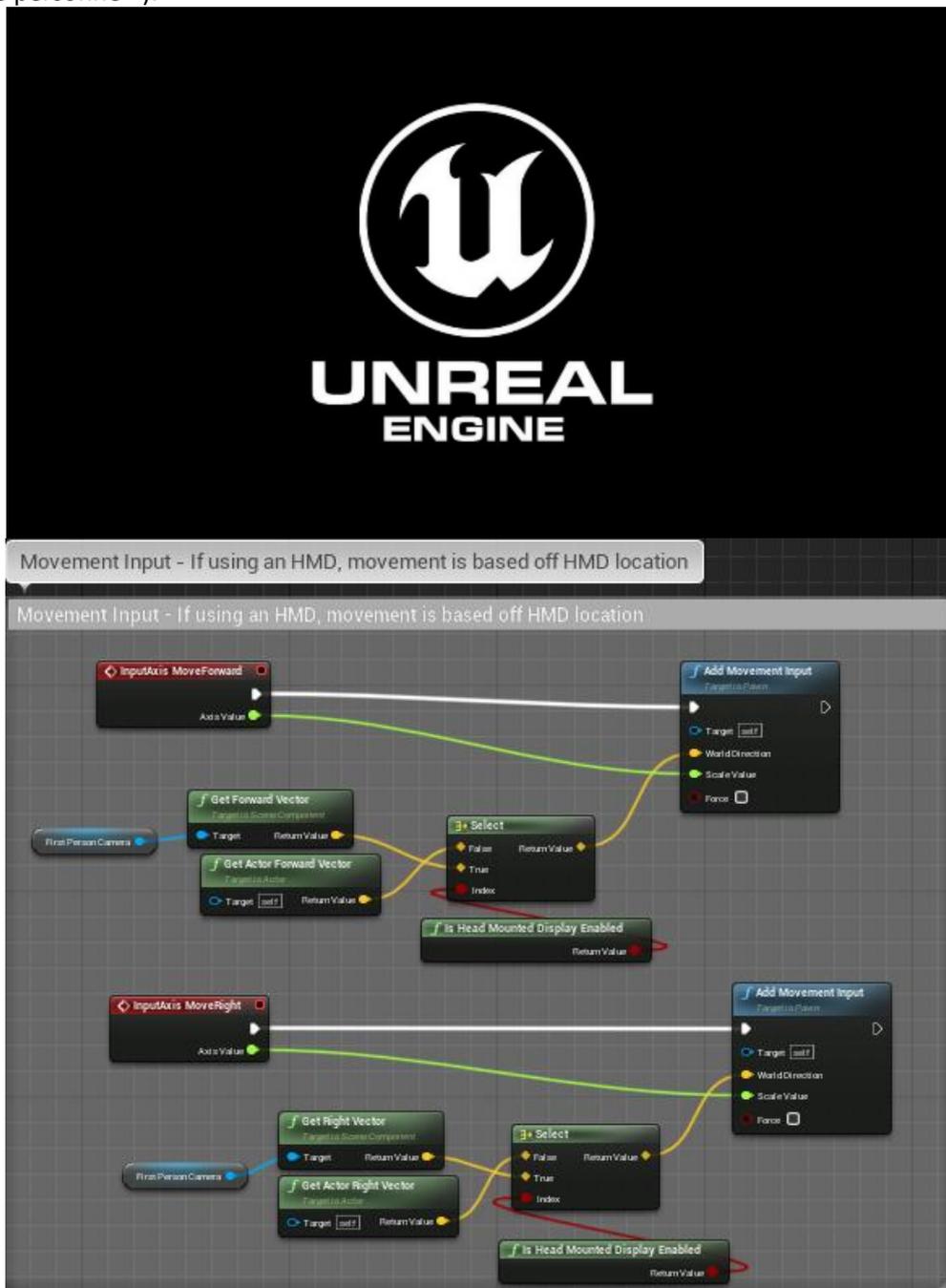
A "Confirmer" button is located at the bottom right of the form.

7. Unreal engine 4.27 (« The Binding of Isaac : Afterbirth + » en Blueprint)  
A. Présentation du projet

Win-Win a quelques projets basé sur Unreal Engine mais n'ayant pas la possibilité de me faire intégrer ce projet mon maître de stage m'a demandé de lui présenter un projet personnel utilisant ce moteur de jeu.

Dans un premier temps, j'ai suivi un cours sur OpenClassrooms afin d'apprendre les bases sur Unreal Engine 4.27 et le Blueprint (Blueprint est un système de programmation appelé « Visual Scripting » ou « script visuel », où, au lieu d'écrire des lignes de code, on relie différents éléments et on les fait interagir entre eux).

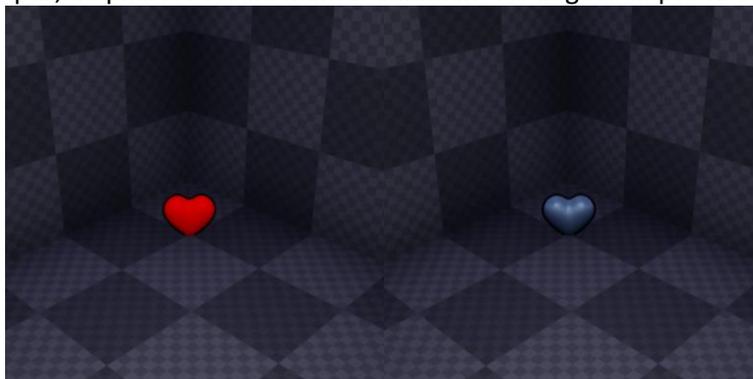
Suite à ce cours, j'ai commencé mon projet personnel où mon but est de refaire le jeu « The Binding of Isaac : Afterbirth + » en 3D et en FPS (« First Person Shooter » ou « jeu de tir à la première personne »).



## B. Structure du projet, problèmes et facilités rencontrés

**Dans un premier temps j'ai commencé par recréer le système de vie par cœur du jeux** (En prenant un coup le joueur perd une moitié de cœur qu'il peut récupérer en ramassant un cœur, le joueur ne peut pas avoir plus de 10 cœurs, si le cœur est vide le réceptacle de cœur reste).

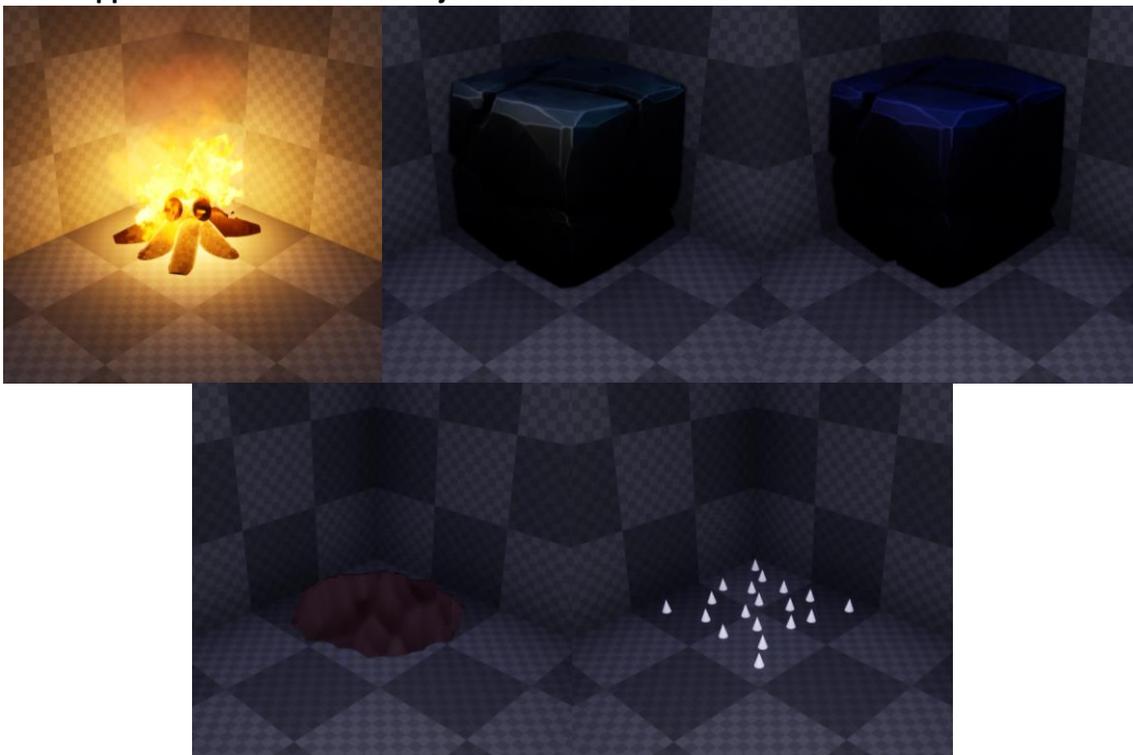
**Une fois le système de vie fini il m'a fallu créer les cœurs, il m'a aussi fallu ajouter les cœurs d'âmes** (Les cœurs d'âmes sont des cœurs non permanents ce qui signifie que si le cœur est vide le réceptacle ne reste pas, de plus les cœurs d'âmes subissent les dégâts en priorité au cœur classique).



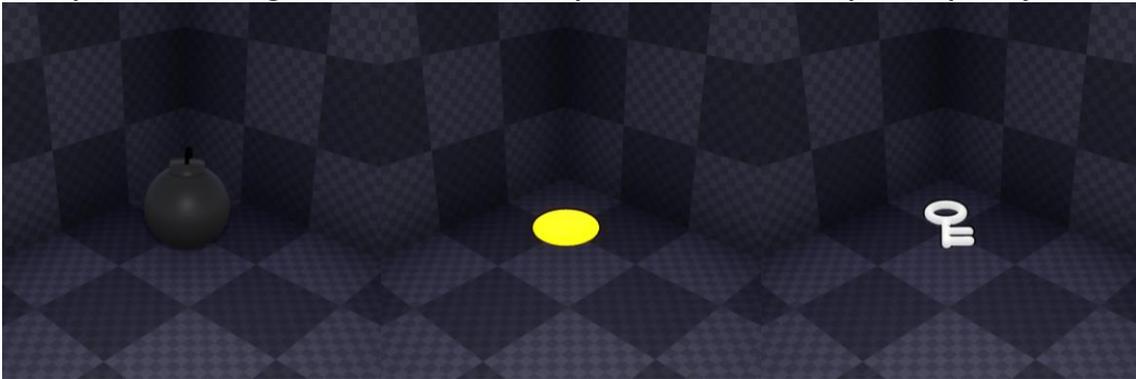
Puis j'ai recréé les différents obstacles du jeu tels que les feux de camps, les rochers, les rochers marqués, les trous et les piques.

Les feux de camps, les rochers ainsi que les trous sont des obstacles infranchissables par le joueur. Les feux de camps peuvent devenir franchissable en les éteignant en tirant dessus, les rochers peuvent être détruit avec des bombes et les trous peuvent être rebouché par un rocher si ce dernier a été détruit par une bombe se trouvant de l'autre côté du rocher par rapport au trou. Les feux de camps allumés et les piques infligent des dégâts au joueur lorsqu'il entre en contact avec ces derniers.

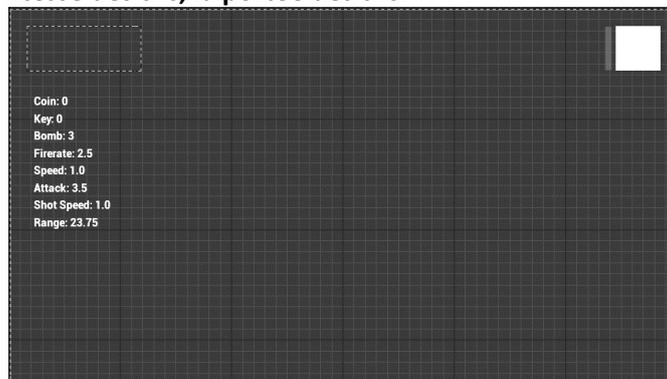
Les rochers marqués fonctionnent comme les rochers classiques cependant ils n'ont que 2,5% de chance d'apparaître et donnent des objets aléatoires lors de leur destruction.



J'ai recréé les bombes permettant de détruire les obstacles, d'infliger des dégâts aux ennemis et au joueur à proximité. J'ai également recréé les clés, pièces et bombes récupérable par le joueur.



Ensuite j'ai créé l'HUD (« Heads-Up Display » ou « affichage tête haute » est l'ensemble des informations renseignant le joueur sur les caractéristiques de son personnage) où l'on peut voir la vie du personnage, son objet actif, s'il en possède un, ainsi que ses caractéristiques tels que le nombre de pièces, de clés, de bombes, de tirs par seconde, la vitesse du joueur, les dégâts infligés par le personnage, la vitesse des tirs, la portée des tirs.



Ensuite, j'ai commencé la génération procédurale des salles.

Dans un premier temps, j'avais comme idée de générer un labyrinthe en texte avec des « \* » pour représenter les salles et des « X » pour les murs, puis de générer, à partir de ce labyrinthe, la disposition du niveau.

Cela posait plusieurs problèmes, la génération de mon labyrinthe commençait toujours en haut à gauche et se terminait toujours en bas à droite donc bien que le chemin soit aléatoire, le joueur savait déjà où se trouvait la sortie, ou le fait que le labyrinthe ne sois qu'en deux dimensions au lieu de trois.

J'ai donc décidé de faire une génération en temps réel c'est-à-dire que les salles adjacentes ne sont générées que lorsque j'y rentre.

Ne maîtrisant pas encore totalement Unreal Engine, je suis allé voir la partie de l'équipe qui l'utilise afin de leur demander un peu d'aide.

Ils m'ont d'abord présenté le métavers qu'ils avaient fait pour un événement, puis l'un d'entre eux m'a dirigé vers un tutoriel pouvant m'aider.

Dans ce dernier la personne supprime les salles une fois les avoir quittées ce que je ne voulais pas puisque le joueur doit pouvoir revenir dans les salles précédentes si il n'a pas emprunté le bon chemin ou afin d'y récupérer des objets qu'il aurait laisser pour plus tard.

J'ai donc enlever la partie correspondant à la suppression de la salle.

Un nouveau problème apparu, des salles peuvent apparaître sur une autre salle déjà existante, il m'a donc fallu vérifier à chaque génération si une salle ce trouve à l'endroit où je souhaite en générer une nouvelle afin de savoir si je peux générer la salle ou non.

Ensuite, j'ai commencé à créer différents modèles de salles pour simplement les faire apparaître aléatoirement plutôt que de les générer de A à Z.



Par la suite j'ai commencé à recréer les objets du jeu, certains d'entre eux sont simple à faire et ne m'ont posé aucun problème puisqu'il ne s'agit que de simple augmentation/diminution des caractéristiques du joueur alors que d'autre m'ont posé certains problèmes comme tirer plusieurs tirs en simultanée à des position différentes, faire des tirs à têtes chercheuse, faire des tirs revenant vers le joueur comme un boomerang, donner des familiers au joueur, donc créer des familiers, etc...

J'ai modélisé tous les objets 3D présent dans le projet à l'aide de Blender à l'exception des rochers et des ennemis que j'ai trouvé sur internet ainsi que les modèles fournis par le moteur de jeu.

### C. Résultat

Pour le moment ce projet contient donc une génération procédurale de la disposition des salles, un HUD, les principaux obstacles et mécanisme du jeu ainsi qu'une partie des objets.

# III. Conclusion

**Durant mon stage chez Win-Win, j'ai pu finaliser la refonte de l'UX/UI de l'outil de gestion interne de l'entreprise et celui-ci est maintenant utilisé de manière pérenne.**

**La recherche de solution métavers sur navigateur internet est en cours de réflexion.**

**Les différents projets effectués au cours de mon stage m'ont permis d'apprendre à me servir de Mongoose, Socket.io, Express.js, GitHub Pages, Nuxt.js, ainsi qu'Unreal Engine avec le Blueprint. Ce stage a confirmé mon envie de travailler dans le domaine du développement logiciel mais a aussi confirmé que je ne raffole pas du développement web.**

**Je tiens à remercier toute l'équipe de Win-Win Marseille de m'avoir accueilli dans le cadre de mon premier stage Epitech, ce qui n'était pas évident puisque la première année d'Epitech nous ne travaillons que sur le C or c'est un langage très peu utilisé au sein de cette entreprise.**

**Un merci plus spécifique à mon maître de stage, Julien Griffon, pour son implication et son idée de me faire travailler sur des technologies nouvelles pour moi en se basant sur mes projets personnels.**

**Une pensée spéciale pour mes deux soutiens inconditionnels, surtout à l'heure des repas, Patate et Napoléon dit Napo, les deux mascottes spécifiques de Marseille.**

