

ETML

# Rapport de Projet SpicyInvaders

CID2B

Mathis Botteau  
03/11/2023

# Rapport Projet SpicyInvaders

## Table des matières

UX – Experience utilisateur .....	1
A. Introduction.....	1
B. Analyse .....	1
C. Conception .....	5
D. Evaluations .....	8
DB – Data Base .....	9
A. Importer les données et le schéma de base de données.....	9
B. Gestions des utilisateurs .....	11
C. Requêtes de sélection .....	13
D. Création des index.....	18
E. Backup / Restore .....	19
F. Liaison C# Base de données .....	20
POO – Programmation orientée objet.....	21
A. Introduction.....	21
B. Analyse fonctionnelle .....	21
C. Analyse technique .....	27
D. Tests Unitaire.....	28
E. Utilisation ChatGPT .....	30
F. Conclusion .....	30

## UX – Experience utilisateur

### A. Introduction

Dans ce rapport, nous allons voir notre projet UX visant à élaborer un prototype interactif avec un menu et des options de jeu cliquables. Ces options, bien qu'elles ne soient pas implémentées dans notre jeu fonctionnel, doivent comprendre :

- Un choix entre le jeu solo ou multijoueur
- La possibilité de sélectionner des palettes graphiques supplémentaires
- La personnalisation des designs pour les ennemis
- Une page de "highscore" pour rivaliser
- Des réglages de gameplay pour adapter la jouabilité, l'accessibilité et l'empreinte écologique.

L'UX est au cœur de ce projet, car elle joue un rôle important dans l'expérience des joueurs. Notre objectif est donc de concevoir une interface conviviale et intuitive qui permettra aux joueurs de vivre une expérience de jeu unique et adapté.

Ce rapport explorera en détail le processus de développement de l'UX en suivant une approche centrée sur l'utilisateur. Cela implique la création de persona, le choix de la palette graphique, l'éco-conception et l'accessibilité.

Dans la section de conception, nous définirons les écrans et les choix à la création de notre prototype interactif. Enfin nous conclurons notre rapport par une évaluation de notre prototype à grâce à des test d'utilisabilité. Cela nous fournira des informations pour améliorer notre projet UX.

### B. Analyse

Conception centrée utilisateur :

La conception centrée sur l'utilisateur est le point central dans notre approche du projet. Son essence même est que la création du produit réponde aux attentes, aux besoins et aux préférences de nos utilisateurs. Pour ce faire, nous avons créé deux personas avec chacun des besoins et des frustrations spécifiques, ce qui nous a permis de définir des options en fonction de nos utilisateurs potentiels et de savoir à quoi nous devons faire attention.


Nous avons aussi choisi une palette graphique par défaut et une autre palette graphique, afin que les utilisateurs aient le choix de la changer si nécessaire. Enfin, nous avons réfléchi à comment rendre notre jeu accessible à tout le monde. Pour cela nous avons créé une option spécifique à l'accessibilité des utilisateurs où ils peuvent modifier la taille des textes, les contrastes, activer ou désactiver les sous-titres, etc.

Personas :

Les Personas sont des personnages fictifs que nous créons pour représenter des utilisateurs cibles. Ces Personas ont différents caractéristiques, besoins et préférences, cela nous permet ainsi de mieux comprendre qui sont nos utilisateurs potentiels et ce qu'ils pourraient attendre de notre produit. Dans notre contexte on les utilise pour savoir quels options, mode de jeu, interface-nous allons mettre en place et à quels problèmes potentiels nous devons faire attention.

Le premier persona a été créé dans le but de concevoir une expérience de jeu vidéo qui prendrais en compte la simplicité et l'accessibilité pour les joueurs plus âgés en minimisant les frustrations liées à la complexité technique et à la lisibilité des éléments dans le jeu.

- Le voici ci-dessous :



**Marie Dubois**  
Retraîtée

**A propos**

72  
Lyon  
Femme  
Retraîtée  
Mariée

**Compétences**

Internet  
Réseaux sociaux  
Shopping en ligne

**Histoire**

Marie Dubois est une retraitée qui a passé la majeure partie de sa vie à enseigner en tant qu'institutrice. Elle est originaire de Lyon et est une grand-mère aimante de quatre petits-enfants. Marie a découvert les jeux vidéo plus tard dans sa vie grâce à ses petits-enfants et les a adoptés comme une nouvelle source de divertissement et de lien avec sa famille.

**Besoins**


- Simplicité d'utilisation : Une interface simple et des contrôles intuitifs pour faciliter le jeu.
- Grande taille de texte et d'icônes : Une option pour augmenter la taille du texte et des icônes pour une lisibilité améliorée.
- Options de sous-titrage : Des sous-titres clairs pour suivre l'histoire.

**Frustrations**

- Complexité excessive : Frustration face à des jeux trop complexes avec des commandes compliquées, ce qui rend difficile la compréhension et la jouabilité.
- Faible lisibilité : Frustration en raison de la petite taille des textes et des icônes, ce qui rend difficile la lecture des informations à l'écran.
- Inaccessibilité : Frustration lorsque les jeux ne proposent pas d'options de sous-titrage ou que les dialogues ne sont pas clairement audibles, rendant difficile la compréhension de l'histoire.

Le deuxième persona a été créé dans le but d'aider à développer une interface de jeu vidéo qui répond aux besoins spécifiques d'un joueur passionné, en mettant l'accent sur l'accessibilité, les options de personnalisation et la jouabilité pour une expérience de jeu de qualité.

- Le voici ci-dessous :



**Alexandre Dupont**  
Graphiste indépendant

**A propos**

30  
Paris  
Homme  
Graphiste  
Célibataire

**Compétences**

Internet  
Réseaux sociaux  
Shopping en ligne

**Histoire**

Alexandre Dupont est un graphiste talentueux basé à Paris. Il a toujours été passionné par les jeux vidéo depuis son enfance, et il a décidé de faire de sa passion son métier. Il a étudié les arts graphiques à l'École Supérieure des Arts de Paris et a travaillé sur divers projets de conception graphique, notamment des jeux mobiles et des sites web. Alexandre est également un fervent défenseur de l'inclusion et de l'accessibilité, en partie parce qu'un de ses proches amis a une déficience visuelle.

**Besoins**

- Accessibilité : L'interface doit offrir des options d'accessibilité, notamment la possibilité d'ajuster la taille des textes, les contrastes et les paramètres sonores.
- Modes de jeu : L'interface doit permettre de choisir entre plusieurs modes de jeu, notamment le mode multijoueur et le mode solo. Les options de jeu doivent être clairement présentées et faciles à sélectionner.
- Options personnalisables : Un sous-menu "Options" doit être inclus, offrant la possibilité d'ajuster les paramètres du jeu selon les préférences d'Alexandre.

**Frustrations**

- Manque d'informations sur les mises à jour : Frustration en raison du manque d'informations sur les mises à jour du jeu, ce qui l'empêche de rester à jour avec les améliorations et les ajouts récents.
- Absence de classement : Frustration liée à l'absence de classements pour suivre sa progression et sa performance par rapport aux autres joueurs.

Choix de la palette graphique :

La palette graphique joue un rôle important dans de l'expérience visuelle de notre jeu. Les joueurs ont la possibilité de choisir entre deux palettes graphiques, chacune avec une ambiance visuelle différente dans l'univers du jeu.

Le thème de notre jeu est l'espace et les aliens c'est pour cela que sur toutes les pages du jeu nous avons mis un fond qui représente l'environnement d'une planète alien animée. Cette image a été exprès pensée loin du réalisme pour que cela montre l'aspect amusant du jeu. Cette approche a été choisie exprès pour que le jeu soit perçu comme amusant.

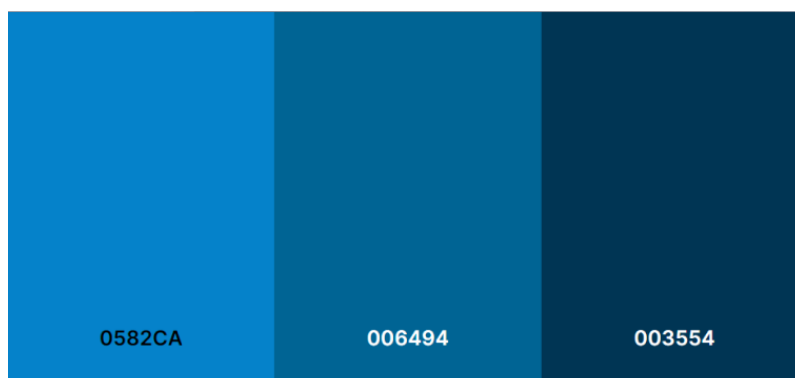
- Voici ci-dessous l'image qui a été choisi pour représenter ce thème :



En ce qui concerne les palettes de couleurs, nous avons décidé d'en choisir deux, une qui sera la palette par défaut et une autre, afin que les utilisateurs puissent les changer si besoin.

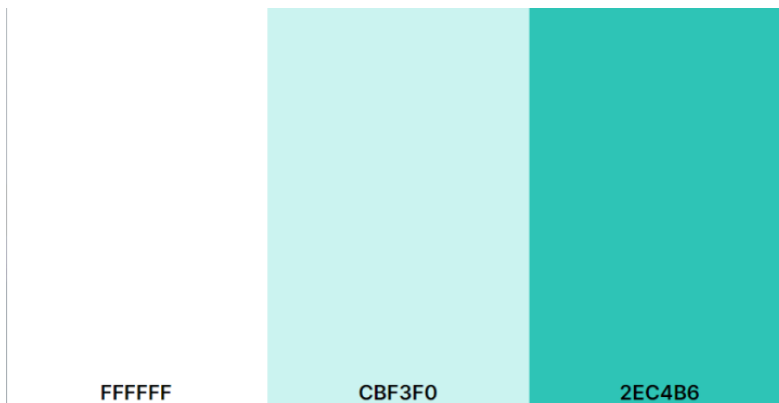
La palette graphique par défaut a pour but de donner l'impression d'exploration d'un univers où l'espace et les extraterrestres se rencontrent. Ces couleurs bleu foncé et turquoise créent une sensation immersive et intrigante, ce qui plonge les joueurs au cœur de découvertes et au centre même de la thématique de notre jeu.

- Voici la palette par défaut sélectionnée :



La palette claire a pour but de donner de la lumière, une impression d'infinité. Ces couleurs blanches et bleu clair créent une ambiance lumineuse. Elle offre une sensation visuelle pleine de bonne humeur, ce qui est parfait avec le thème de notre jeu.

- Voici la palette claire sélectionnées :



#### Eco-conception :

L'éco-conception vise à minimiser l'impact environnemental et écologique d'un produit de sa création à sa fin. Cette approche prend en compte certains critères comme :

- La réduction de la consommation d'énergie
- La durabilité des matériaux
- La réduction des déchets

Dans notre projet cela semblait important d'amener une touche d'éco-conception, c'est pour cela que nous avons essayé de prendre en compte le critère de « La réduction de la consommation d'énergie ».

En effet notre interface est de couleur sombre, cela nécessite donc moins d'énergie pour être afficher. De plus cela contribue notamment à la prolongation de l'autonomie de la batterie des appareils mobiles, ce qui réduit les déchets potentiel et aide à la durabilité de ces matériaux. Enfin nos images sont en format PNG, grâce à cela nous permettons lors de la compression de l'image d'évité la perte de qualité et nous réduisons aussi la charge énergétique que l'image prend sur les serveurs.

#### Accessibilité :

L'accessibilité en UX consiste à concevoir un produit de manière qu'il soit utilisable par tout type de personne, même celles qui sont atteint d'un handicap ou de besoins spécifiques. Dans notre jeu, nous avons mis en place une option spécifique à l'accessibilité, en effet nos utilisateurs peuvent s'ils le souhaitent activer ou désactiver certaines options.

Les options disponibles sont :

- Option contraste (elle permet de gérer le taux de contraste des interfaces)
- Option dyslexique (elle permet d'afficher les textes dans une police plus lisible)
- Option contrôle vocal (elle permet de naviguer vocalement dans les interfaces)
- Option synthèse vocal (elle permet de lire vocalement toutes les informations qui son présent sur les pages)
- Option sous-titres (elle permet d'afficher les sous titres)



## C. Conception

Définition des écrans :

La conception de nos maquettes son sur des bases de définition 1440 x 1024, nous avons choisi cette définition car cette résolution est souvent utilisée sur les écrans des ordinateurs et donc cela garantit que les maquettes seront adaptées à une grande variété d'appareils.

Choix effectué :

Dans notre étape de conception nous avons créé des de maquettes pour offrir aux utilisateurs une expérience fluide, personnalisable et transparente.

Voici les choix importants effectué :

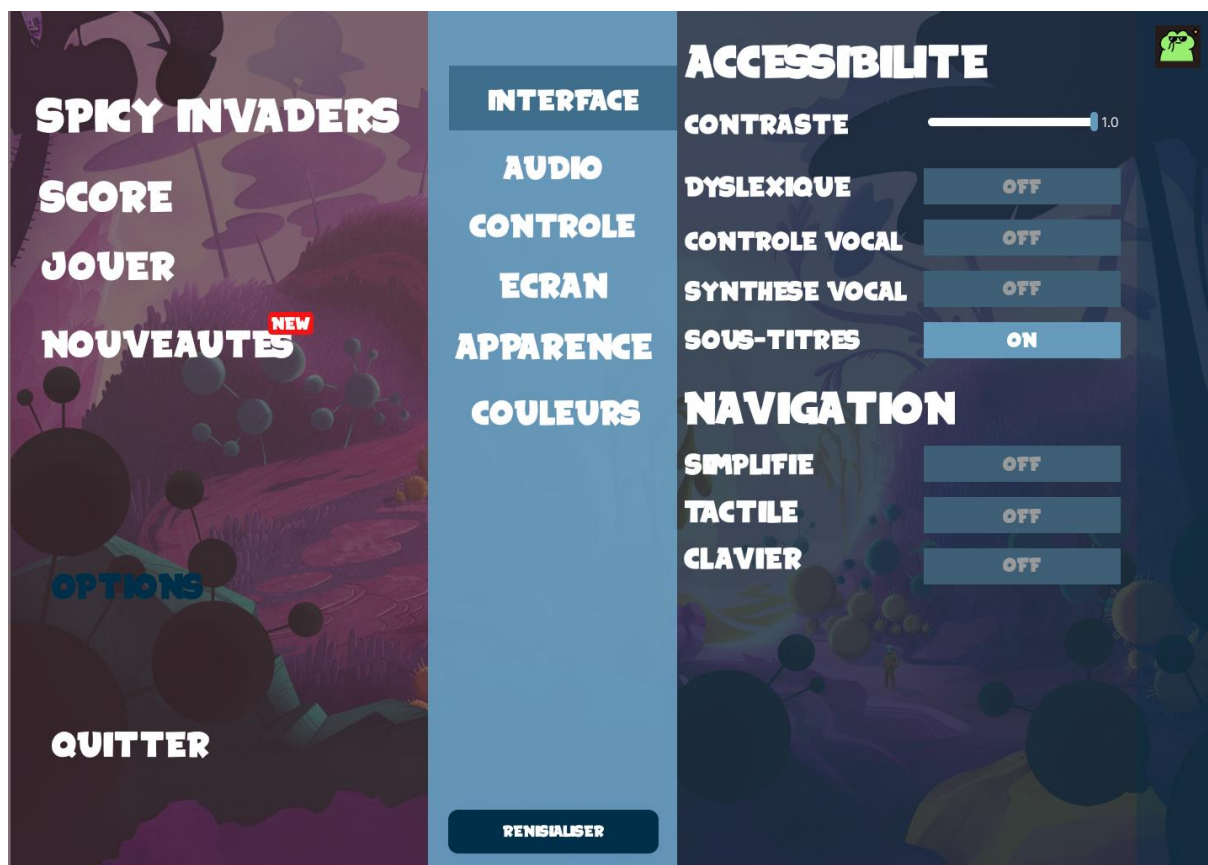
**Menu Principal de Navigation** : Nous avons veillé à ce que le menu principal de navigation soit visible sur toutes les pages. Cette approche permet aux utilisateurs de se déplacer librement entre les différentes parties du jeu depuis n'importe où, offrant une accessibilité optimale.

**Menu d'Accueil** : Nous avons créé un menu d'accueil convivial qui accueille les joueurs sur une page centrale, leur permettant de commencer leur aventure depuis n'importe quel point du jeu.

**Photo de Profil** : Sur toutes les pages, nous avons placé la photo de profil du joueur du côté gauche. Cela permet aux utilisateurs de toujours savoir sur quel compte ils sont connectés.



**Sous-Menu Options :** Nous avons créé un sous-menu "Options" pour permettre aux utilisateurs de personnaliser leur expérience de jeu et leur interface. Ils peuvent ajuster des paramètres tels que l'interface, l'audio, les contrôles, l'écran, les couleurs et l'apparence pour obtenir la meilleure expérience possible.



**Sous-Menu Scores :** Un sous-menu "Scores" offre aux joueurs la possibilité de consulter leur classement en mode multijoueur et en mode solo, ajoutant de la compétitivité au jeu.





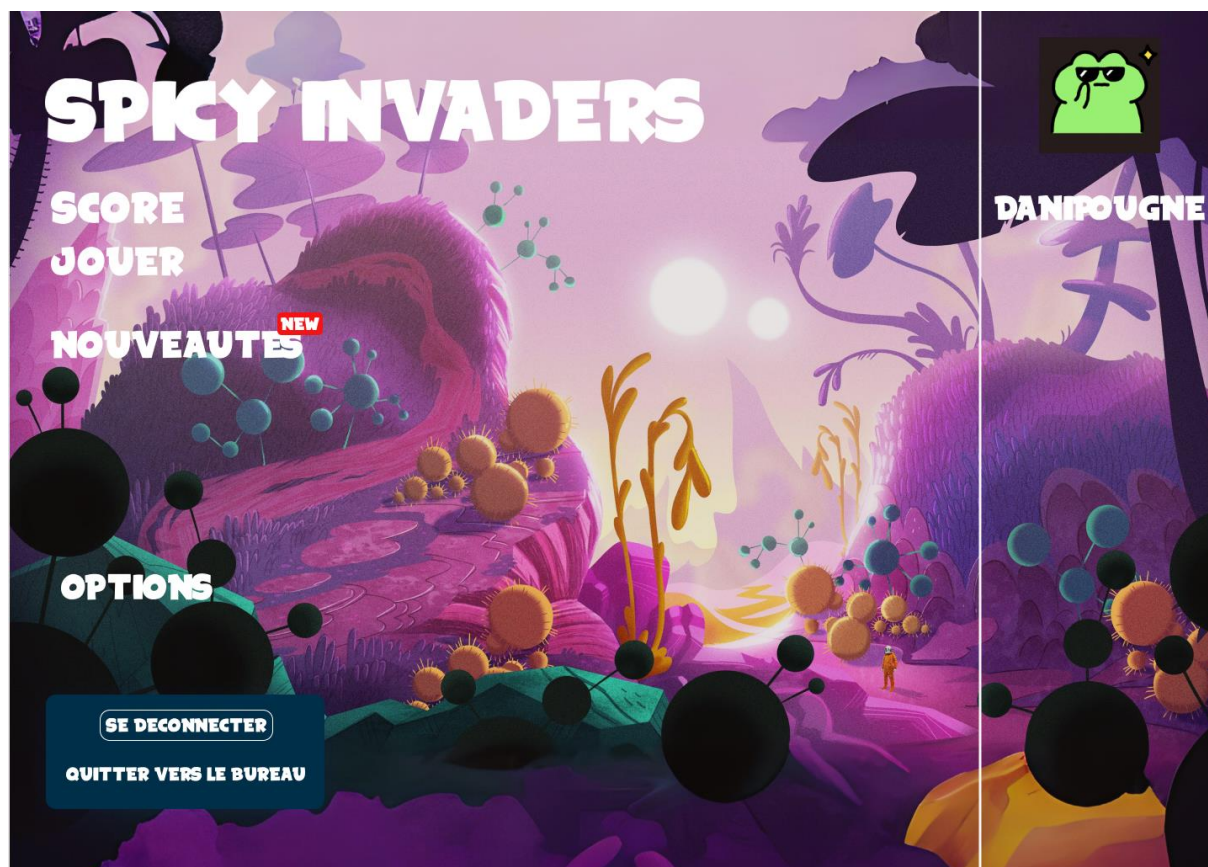
**Sous-Menu Jouer :** Le sous-menu "Jouer" permet aux utilisateurs de choisir entre le mode multijoueur et solo. Dans le multijoueur ils ont accès à une liste de serveurs qu'ils peuvent rejoindre ou créer. Dans le mode solo il y a des niveaux de difficulté progressifs et même une option de partie d'entraînement pour varier les modes de jeu.



**Sous-Menu Nouveautés :** Un sous-menu "Nouveautés" tient les utilisateurs informés de toutes les mises à jour et des ajouts au jeu, ce qui permet d'établir une communication transparente pour renforcer la confiance.



**Bouton Quitter** : Sur le menu de navigation, nous avons inclus un bouton "Quitter" qui permet aux utilisateurs de revenir au menu principal en un clic, de retourner au bureau en deux clics ou de fermer le jeu, cela permet d'offrir une multiplié de choix à nos utilisateurs.



En conclusion nos choix de conception visent à offrir aux utilisateurs la plus grande liberté et la meilleure expérience possible, tout cela en prenant en compte des besoins spécifiques tels que l'accessibilité et la personnalisation.

#### D. Evaluations

Nom du Menu	Description	Condition pré-requise	Etape de test	Donnée de test	Résultat attendu	Résultat actuelle
Menu Principal	Menu d'accueil du jeu, avec des sous menu cliquable	Application installée et lancé	1. Cliquer sur un des sous menus présents		L'utilisateur apparait dans la page du sous menu qu'il à sélectionner	On apparait dans la page du sous menu qu'il à sélectionner
Menu Score (Multijoueur)	Menu score, affiche le classement des joueurs en fonction du score pour le mode	Application installée et lancé et avoir cliquer sur le menu « Score »	1. Clique sur le sous menu « Multi-joueur »		Surlignement du sous menu « multi-joueur » et affichage du classement des joueur	Le sous menu « multi-joueur » se surligne et le classement des joueurs par apport à

	de jeux multijoueur				par apport à leur score	leur score s'affiche
Menu Jouer (Multijoueur)	Menu jouer (multi), permet de rejoindre / créer une partie parmi une liste de serveurs afin d'y jouer	Application installée et lancé et avoir cliquer sur le menu « Jouer »	1. Clique sur le sous menu « Multijoueur » 2. Clique sur le bouton « créer partie »		Une nouvelle fenêtre apparait avec l'interface et les options pour créer une partie	Le sous menu « multi - joueur » se surligne et la liste des serveurs disponible s'affiche
Menu Option (Apparence)	Menu option (apparence), permet de changer l'apparence des aliens et du vaisseau du joueurs	Application installée et lancé et avoir cliquer sur le menu « Option »	1. Clique sur le sous menu « Apparence » 2. Clique sur la flèche « > » pour le joueur		Changement de l'apparence du vaisseau du joueur	L'apparence du vaisseau du joueur change
Menu Nouveautés	Menu nouveauté, affiche toutes les nouveautés et les mise à jours qui ont été ajouté	Application installée et	1. Clique sur le sous menu « Nouveautés »		L'utilisateur apparait dans la page sous menu « nouveautés » et cela affiche les nouveauté actuel	On apparait dans la page sous menu « nouveautés » et cela affiche les nouveauté actuel

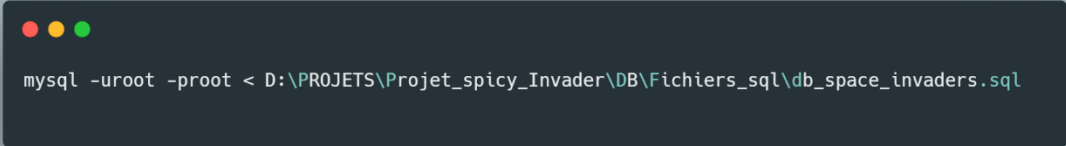
## DB – Data Base

### A. Importer les données et le schéma de base de données

Pour pouvoir charger le fichier qui contient le dump de notre base de données, il faut utilisée les commandes suivantes :

1. La première commande à utiliser est « **mysql -u root -p** », cette commande permet de se connecter au serveur MySQL avec notre utilisateur root et notre mot de passe, grâce à cela nous allons pouvoir interagir avec notre serveur de bases de données.  
« **mysql** » Cette partie sert à indiquer que nous voulons utiliser le client MySQL, ce qui va nous permettre de pouvoir utiliser et exécuter des commandes Mysql directement dans notre cmd.  
« **-u root** » Cette partie sert à savoir et à spécifier avec quel utilisateur nous voulons-nous connecter. Dans notre cas c'est root car c'est le super utilisateur et aussi l'utilisateur par défaut.  
« **-p** » Cette partie permet d'avertir le client MySQL qui est nécessaire d'entrer un mot de passe, après cela il nous demandera d'insérer notre mot de passe pour qu'on puisse se connecter au serveur.

2. La deuxième commande à insérer dans notre cmd est la suivante :  
Cette commande va nous permettre de charger le dump de notre base de données directement dans notre serveur Mysql.  
Comme dit un peu plus haut « **mysql** », « **-uroot** », « **-proot** » servent donc à utiliser le client MySQL, puis ce même client va se connecter avec l'utilisateur et le mot de passe spécifier.

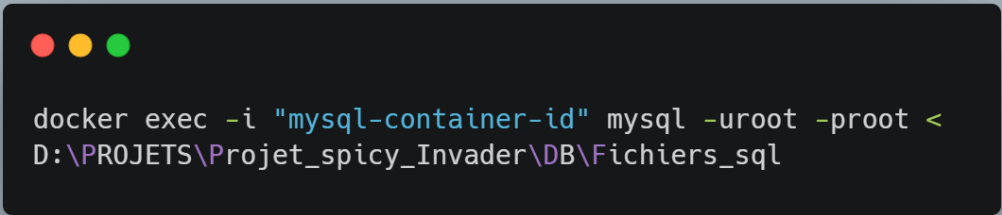


```
mysql -uroot -proot < D:\PROJETS\Projet_spicy_Invader\DB\Fichiers_sql\db_space_invaders.sql
```

La dernière partie,

« **<D:\PROJETS\Projet\_spicy\_Invader\DB\Fichiers\_sql\db\_space\_invaders.sql** » sert à indiquer le chemin où se trouve notre fichier .sql. Quant à l'indicateur « **<** » il spécifie que l'on veut transmettre des informations à partir d'un fichier extérieur à la base de données, dans le cas contraire l'indicateur « **>** » spécifie que l'on veut transmettre des informations de la base de données à un fichier extérieur.

3. Dans le cas, où notre serveur MySQL est situé dans un conteneur docker nous devons utiliser cette commande dans un cmd extérieur.



```
docker exec -i "mysql-container-id" mysql -uroot -proot < D:\PROJETS\Projet_spicy_Invader\DB\Fichiers_sql
```

C'est la même commande que précédemment, mais nous avons ajouté juste devant « **docker exec -i** » « **mysql-container-id** ». Cette commande nous permet de pouvoir accéder au conteneur où se trouve le serveur MySQL et d'y exécuter des commandes. Voici les détails de la commande :

« **docker exec** » Cette partie sert à accéder dans un conteneur et de pouvoir utiliser et exécuter des commandes comme si on se trouvait directement dans l'environnement MySQL.

« **-i** » Cette partie sert à ce que nous puissions interagir dans le conteneur en temps réel. Par exemple si nous voulons pouvoir interagir ou questionner avec notre base de données il est important que cela soit fait en temps réel pour que l'on puisse voir les données.



« **mysql-container-id** » Cette partie sert tout simplement à spécifier l'identifiant du conteneur avec le quelle nous voulons interagir, dans notre cas c'est celui du conteneur db là où se trouve le serveur MySQL.

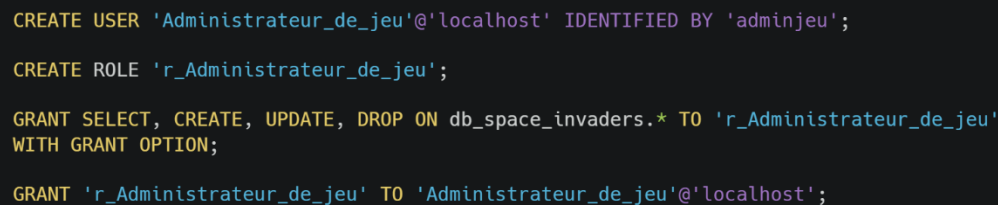
## B. Gestions des utilisateurs

Administrateur :

L'utilisateur demandé doit pouvoir effectuer les missions suivantes :

- Peut créer, lire, mettre à jour et supprimer (CRUD) n'importe quelle table.
- Gérer les utilisateurs et leurs privilèges.

Voici les commandes à faire :



```
CREATE USER 'Administrateur_de_jeu'@'localhost' IDENTIFIED BY 'adminjeu';  
CREATE ROLE 'r_Administrateur_de_jeu';  
GRANT SELECT, CREATE, UPDATE, DROP ON db_space_invaders.* TO 'r_Administrateur_de_jeu'  
WITH GRANT OPTION;  
GRANT 'r_Administrateur_de_jeu' TO 'Administrateur_de_jeu'@'localhost';
```

La première commande « `CREATE USER 'Administrateur_de_jeu'@'localhost' IDENTIFIED BY 'adminjeu';` » sert à créer un nouvel utilisateur nommé 'Administrateur\_de\_jeu' avec comme mot de passe 'adminjeu'.

- La partie « `CREATE USER` » sert à indiquer que nous allons créer un nouvel utilisateur dans notre base de données.
- La partie « `'Administrateur_de_jeu'@'localhost'` » sert à spécifier le nom que nous voulons donner à notre utilisateur et l'host auquel il sera autorisé à se connecter.
- La partie « `IDENTIFIED BY 'adminjeu'` » sert à définir le mot de passe qui sera associé à notre utilisateur.

La deuxième commande « `CREATE ROLE 'r_Administrateur_de_jeu';` » sert tout simplement à créer un rôle du nom 'r\_Administrateur\_de\_jeu'.

- La partie « `CREATE ROLE` » sert à indiquer que nous allons créer un nouveau rôle dans notre base de données.
- La partie « `'r_Administrateur_de_jeu'` » sert à spécifier le nom du rôle que nous allons créer.

La troisième commande « `GRANT SELECT, CREATE, UPDATE, DROP ON db_space_invaders.* TO 'r_Administrateur_de_jeu' WITH GRANT OPTION;` » sert à donner des privilèges au rôle que nous avons créé précédemment.

- La partie « GRANT SELECT, CREATE, UPDATE, DROP » sert à indiquer quels privilèges nous allons donner à notre rôles, dans notre cas on accorde le droit de lecture de données ( SELECT), le droit de créations d'objets (CREATE), le droit de mise à jour de données (UPDATE) et le droit de suppression d'objets (DROP).
- La partie « ON db\_space\_invaders.\* » indique sur quelles base de données et sur quelles tables ces privilèges pourront être utiliser.
- La partie « TO 'r\_Administrateur\_de\_jeu' » sert à indiquer à quel rôle nous allons donner ces privilèges.
- La partie « WITH GRANT OPTION » permet au rôle indiquer d'accorder ces mêmes privilèges à d'autres utilisateurs.

Pour finir la dernière commande « GRANT 'r\_Administrateur\_de\_jeu' TO 'Administrateur\_de\_jeu'@'localhost'; » sert à accorder le rôle 'r\_Administrateur\_de\_jeu' à l'utilisateur 'Administrateur\_de\_jeu', grâce à cela l'utilisateur bénéficiera des privilèges que nous avons à accorder au rôle.

- La partie « GRANT » sert à indiquer que nous allons accorder un rôle.
- La partie « 'r\_Administrateur\_de\_jeu' » spécifie quel rôle va et être accordé.
- La partie « TO 'Administrateur\_de\_jeu'@'localhost' » indique à quel utilisateur le rôle est attribué et que l'utilisateur est autorisé à se connecter depuis l'host spécifié.

Joueur :

L'utilisateur demandé doit pouvoir effectuer les missions suivantes :

- Lire les informations des armes (pour voir quelles armes il peut acheter).
- Créer une commande.
- Lire toutes les commandes.
- Voici les commandes à faire :

```
CREATE USER 'Joueur'@'localhost' IDENTIFIED BY 'joueur';  
CREATE ROLE 'r_Joueur';  
GRANT SELECT ON db_space_invaders.t_arme TO 'r_Joueur';  
GRANT SELECT, CREATE ON db_space_invaders.t_commande TO 'r_Joueur';  
GRANT 'r_Joueur' TO 'Joueur'@'localhost';
```




Les commandes ci-dessus sont les mêmes que nous avons utilisé pour la gestion de l'utilisateur administrateur, mais nous donnons à notre joueur des privilèges différents sur des tables différentes.

Gestionnaire de la boutique :

L'utilisateur demandé doit pouvoir effectuer les missions suivantes :

- Lire les informations sur tous les joueurs.
- Mettre à jour, lire et supprimer des armes.
- Lire toutes les commandes.
- Voici les commandes à faire :



```
CREATE USER 'Joueur'@'localhost' IDENTIFIED BY 'joueur';  
CREATE ROLE 'r_Joueur';  
GRANT SELECT ON db_space_invaders.t_arme TO 'r_Joueur';  
GRANT SELECT, CREATE ON db_space_invaders.t_commande TO 'r_Joueur';  
GRANT 'r_Joueur' TO 'Joueur'@'localhost';
```

### C. Requêtes de sélection

Requête n°1 :

La requête demandée est de sélectionner les 5 joueurs qui ont le meilleur score et les joueurs doivent être classés dans l'ordre décroissant.

- Voici la commande pour le faire :



```
SELECT * FROM t_joueur ORDER BY jouNombrePoint DESC LIMIT 5;
```

- La clause "SELECT" est utilisée pour spécifier les colonnes que l'on souhaite extraire de la table. L'utilisation de l'astérisque "\*" signifie que l'on extrait toutes les colonnes disponibles dans la table "t\_joueur".
- La clause "FROM" indique la table à partir de laquelle on va extraire les données, en l'occurrence c'est la table "t\_joueur".

- La clause « ORDER BY » sert à ordonner le résultat de la requête en fonction d'une colonne. Dans notre cas on l'ordonne en fonction de « jouNombrePoint » dans un ordre décroissant « DESC », il est aussi possible de l'ordonner par croissant « ASC ».
- La clause « LIMIT » sert à limiter le nombre de résultat envoyer des colonnes. Dans notre cas on le limite à 5 pour avoir les 5 meilleur joueurs.

Requête n°2 :

La requête demandée est de trouver le prix maximum, minimum et moyen des armes. Les colonnes doivent avoir comme nom « PrixMaximum », « PrixMinimum » et « PrixMoyen ».

- Voici la commande pour le faire :

```
SELECT MAX(armPrix) AS PrixMaximum, MIN(armPrix) AS PrixMinimum, AVG(armPrix) AS PrixMoyen
FROM t_arme;
```

- La fonction « MAX() » sert à prendre la valeur la plus haute de la colonne qui se trouve dans les parenthèses. En l'occurrence c'est « armPrix » car on veut le prix le plus haut.
- La fonction « MIN() » sert à prendre la valeur la plus basse de la colonne qui se trouve dans les parenthèses. En l'occurrence c'est « armPrix » car on veut le prix le plus bas.
- La fonction « AVG() » sert à prendre la valeur moyenne de la colonne qui se trouve dans les parenthèses. En l'occurrence c'est « armPrix » car on veut le prix moyen.
- L'alias « AS » sert à renommer le résultat de la fonction qui la précède, dans notre cas on les renomme en « PrixMaximum », « PrixMinimum » et « PrixMoyen ».

Requête n°3 :

La requête demandée est de trouver le nombre total de commande par joueur, de l'ordonner du plus grand au plus petit et les colonnes doivent avoir le nom "IdJoueur", "NombreCommandes".

- Voici la commande pour le faire :

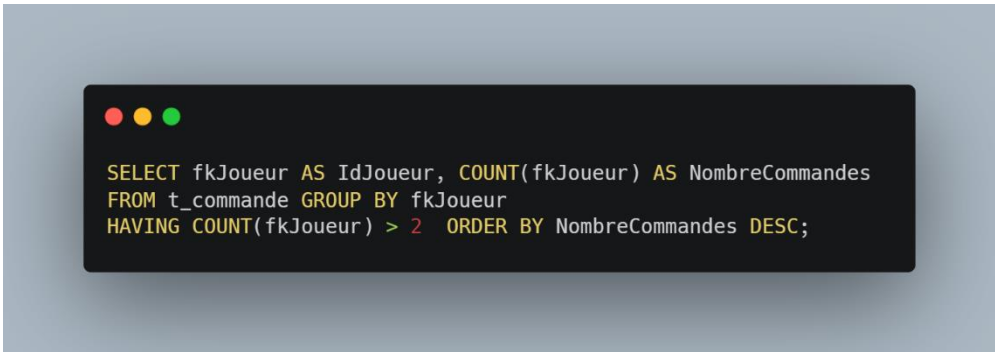
```
SELECT fkJoueur AS IdJoueur, COUNT(fkJoueur) AS NombreCommandes
FROM t_commande GROUP BY fkJoueur
ORDER BY NombreCommandes DESC;
```

- La fonction « COUNT() » sert à compter le nombre de ligne de la colonne dans les parenthèses, dans notre cas c'est « fkJoueur », car on veut le nombre total de commande.
- La clause « GROUP BY » sert à regrouper les données en fonction d'une colonne, dans notre cas on les regroupe par « fkJoueur » car on veut trouver le nombre de commande pour chaque joueur. Cela permet donc de compter le nombre de commande par joueur.

Requête n°4 :

La requête demandée est de sélectionner les joueurs qui ont passé plus de 2 commandes, et les colonnes doivent avoir le nom "IdJoueur", "NombreCommandes".

- Voici la commande pour le faire :



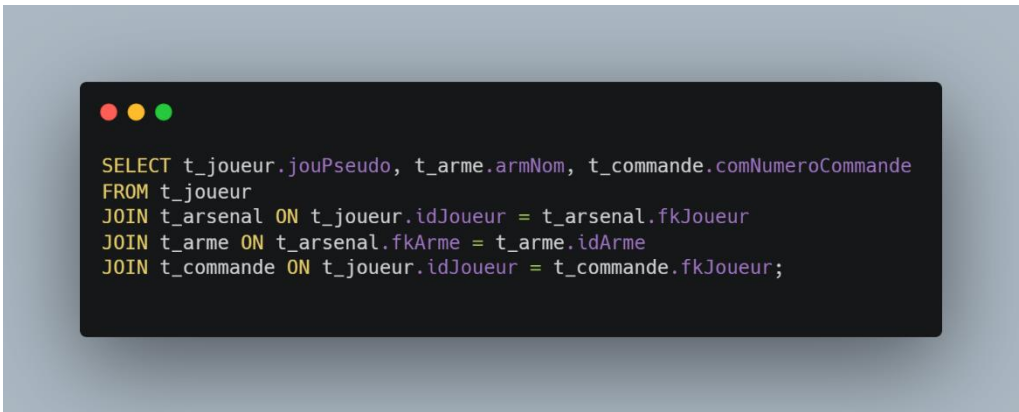
```
SELECT fkJoueur AS IdJoueur, COUNT(fkJoueur) AS NombreCommandes
FROM t_commande GROUP BY fkJoueur
HAVING COUNT(fkJoueur) > 2 ORDER BY NombreCommandes DESC;
```

- La clause « HAVING » permet de filtrer les résultats de « GROUP BY ». Dans notre cas on sélectionne seulement les groupes « fkJoueur » qui ont passé plus de 2 commandes.

Requête n°5 :

La requête demandée est de trouver le pseudo du joueur et le nom de l'arme pour chaque commande.

- Voici la commande pour le faire :



```
SELECT t_joueur.jouPseudo, t_arme.armNom, t_commande.comNumeroCommande
FROM t_joueur
JOIN t_arsenal ON t_joueur.idJoueur = t_arsenal.fkJoueur
JOIN t_arme ON t_arsenal.fkArme = t_arme.idArme
JOIN t_commande ON t_joueur.idJoueur = t_commande.fkJoueur;
```

- La clause « JOIN » sert à effectuer une jointure entre une première table et une deuxième table en utilisant le « ON » pour spécifier sur quelle table on la joint. Une jointure permet de combiner des données de plusieurs tables différentes, dans notre cas on a besoin des données de la table « t\_joueur », « t\_arsenal » et « t\_commande ».

## Requête n°6 :

La requête demandée est de trouver le total dépensé par chaque joueur en ordonnant par le montant le plus élevé en premier, et limiter aux 10 premiers joueurs. La colonne doit avoir pour nom "IdJoueur" et "TotalDepense"

- Voici la commande pour le faire :

```
SELECT t_joueur.idJoueur AS IdJoueur, SUM(t_arme.armPrix * t_detail_commande.detQuantiteCommande)
AS TotalDepense
FROM t_commande
JOIN t_joueur ON t_commande.fkJoueur = t_joueur.idJoueur
JOIN t_detail_commande ON t_commande.idCommande = t_detail_commande.fkCommande
JOIN t_arme ON t_detail_commande.fkArme = t_arme.idArme
GROUP BY IdJoueur
ORDER BY TotalDepense DESC
LIMIT 10;
```

- La fonction « SUM() » calcul la somme des données de la colonnes entre parenthèses, dans notre cas on calcul la somme des dépense total de chaque joueur et ducoup on multiplie le prix de chaque armes par la quantité commandée.

## Requête n°7 :

La requête demandée est de récupérer tous les joueurs et leurs commandes, même s'ils n'ont pas passé de commandes.

- Voici la commande pour ce faire :

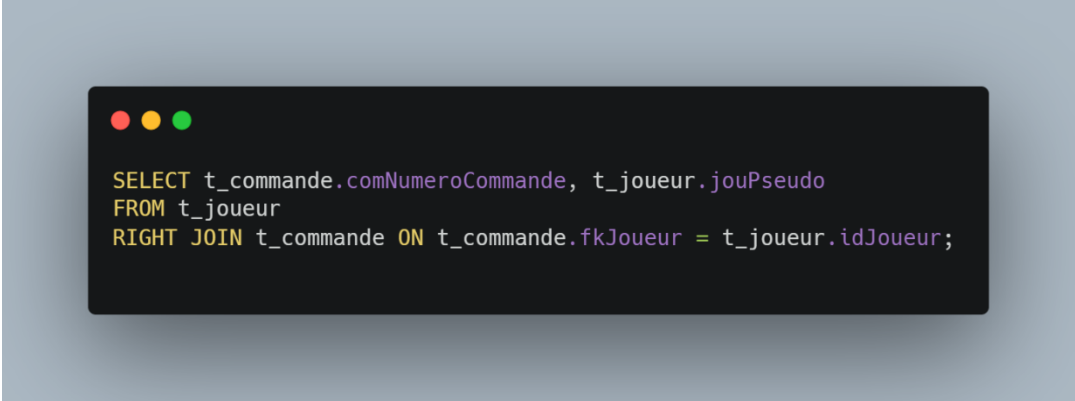
```
SELECT t_joueur.jouPseudo, t_commande.comNumeroCommande
FROM t_joueur
LEFT JOIN t_commande ON t_commande.fkJoueur = t_joueur.idJoueur;
```

- La clause "LEFT JOIN" est le même principe que le "JOIN", sauf que dans un "LEFT JOIN" toutes les lignes de la table située à gauche de la jointure (dans ce cas c'est la table "t\_joueur") sont incluses dans les résultats.

Requête n°8 :

La requête demandée est de récupérer toutes les commandes et afficher le pseudo du joueur si elle existe, sinon montrer `NULL` pour le pseudo.

- Voici la commande pour le faire :



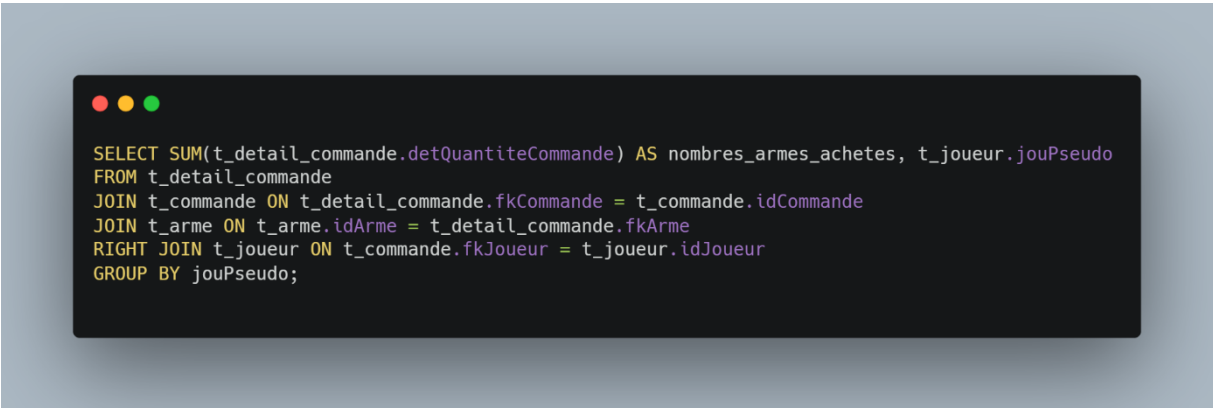
```
SELECT t_commande.comNumeroCommande, t_joueur.jouPseudo
FROM t_joueur
RIGHT JOIN t_commande ON t_commande.fkJoueur = t_joueur.idJoueur;
```

- La clause "RIGHT JOIN" est le même principe que le "JOIN", sauf que dans un "RIGHT JOIN", toutes les lignes de la table située à droite de la jointure (dans ce cas c'est la table "t\_commande") sont incluses dans les résultats.

Requête n°9 :

La requête demandée est de trouver le nombre total d'armes achetées par chaque joueur (même si ce joueur n'a acheté aucune Arme).

- Voici la commande pour le faire :

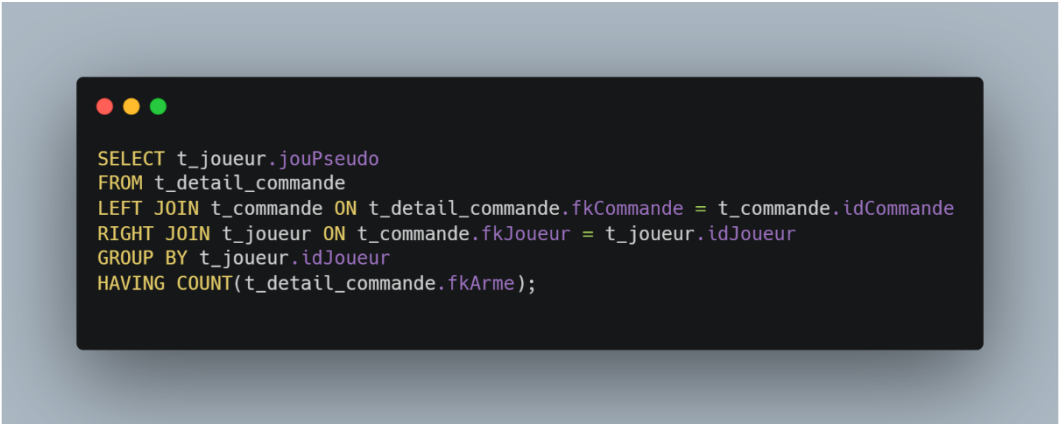


```
SELECT SUM(t_detail_commande.detQuantiteCommande) AS nombres_armes_achetes, t_joueur.jouPseudo
FROM t_detail_commande
JOIN t_commande ON t_detail_commande.fkJoueur = t_commande.fkJoueur
JOIN t_arme ON t_arme.idArme = t_detail_commande.fkJoueur
RIGHT JOIN t_joueur ON t_commande.fkJoueur = t_joueur.idJoueur
GROUP BY jouPseudo;
```

Requête n°10 :

La requête demandée est de trouver les joueurs qui ont acheté plus de 3 types d'armes différentes.

- Voici la commande pour le faire :



```
SELECT t_joueur.jouPseudo
FROM t_detail_commande
LEFT JOIN t_commande ON t_detail_commande.fkCommande = t_commande.idCommande
RIGHT JOIN t_joueur ON t_commande.fkJoueur = t_joueur.idJoueur
GROUP BY t_joueur.idJoueur
HAVING COUNT(t_detail_commande.fkArme);
```

#### D. Création des index

Pourtant certains index existent déjà. Pourquoi ? :

En examinant notre dump MySQL "db\_space\_invaders.sql", on peut constater que le mot-clé "INDEX" n'est pas explicitement utilisé pour créer des index. Cependant certains index existent déjà dans la base de données.

Ces index sont créés automatiquement par MySQL lorsque des colonnes ont déclarées comme clés primaires. Lorsqu'une colonne est indiquée comme une clé primaire, un index unique est automatiquement créé et attribuer à cette colonne. Cela permet que les valeurs de cette colonne soient uniques et optimise les opérations de recherche dans la table. Cela permet aussi d'indexer nos ID qui sont définit en tant que clés primaires.

Quels sont les avantages et les inconvénients des index ?

Les index sont des structures de données permettant d'optimiser les opérations de recherche, ainsi lorsqu'une requête est effectuée, l'accès aux informations est plus rapide. Donc les index permettent d'améliorer les performances des requêtes.

Cependant, les index présentent également des inconvénients. Ils augmentent la taille de la base de données, ce qui peut sembler négligeable dans le cas de bases de données plus petites, mais cela peut devenir problématique dans des bases de données plus grande. En effet dans ce cas l'augmentation de taille peut entraîner une plus grosse consommation de l'espace de stockage et donc ralentir les opérations.

Sur quel champ cela pourrait être pertinent d'ajouter un index ?

Dans notre base de données l'ajout d'index doit être une décision réfléchie étant donné que notre volume de données actuel est petit. Il est donc important de peser attentivement les avantages et les inconvénients de chaque index, car un excès d'index peut surcharger la base de données. Cependant si nous devons envisager d'ajouter un d'index, il serait bénéfique d'en ajouter un à la colonne "jouNombrePoints" de la table "t\_joueur". En effet cette colonne va être fréquemment utilisée dans



le cadre de notre jeu Space Invaders, notamment pour les opérations de recherche des scores des joueurs.

### E. Backup / Restore

Backup :

Pour cette commande il nous est demandé de faire un backup de la base de données « db\_space\_invaders ».

- Voici la commande pour le faire :

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The command `mysqldump -uroot -proot db_space_invaders > sauvegarde.sql` is entered in a light-colored monospace font.

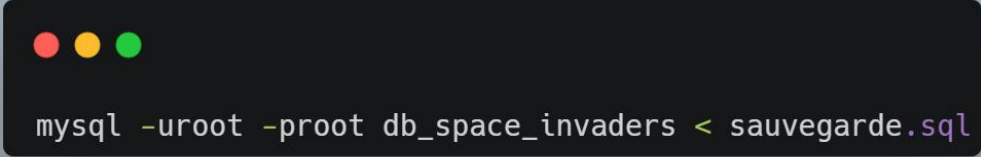
```
mysqldump -uroot -proot db_space_invaders > sauvegarde.sql
```

- La partie "mysqldump" sert à ce qu'on puisse interagir avec le serveur MySQL pour créer une sauvegarde des données de la base de données spécifiée.
- La partie « -uroot » spécifie le nom d'utilisateur que nous voulons utiliser pour nous connecter au serveur.
- La partie « -proot » indique le mot de passe associé à notre utilisateur pour que l'on puisse se connecter au serveur.
- La partie « db\_space\_invaders » sert à indiquer le nom de la base de données que nous voulons sauvegarder.
- La partie « > sauvegarde.sql » sert à indiquer que nous allons rediriger la base de données indiquée plus tôt dans la commande, dans un nouveau fichier nommé « sauvegarde.sql »

Restore :

Pour cette commande il nous est demandé de faire un restore de la base de données « db\_space\_invaders ».

- Voici la commande pour le faire :

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The command `mysql -uroot -proot db_space_invaders < sauvegarde.sql` is displayed in a light-colored monospace font.

```
mysql -uroot -proot db_space_invaders < sauvegarde.sql
```

- La partie « mysql » sert à ce que l'on puisse interagir avec le serveur MySQL.
- La partie « -uroot » spécifie le nom d'utilisateur que nous voulons utiliser pour nous connecter au serveur.
- La partie « -proot » indique le mot de passe associé à notre utilisateur pour que l'on puisse se connecter au serveur.
- La partie « db\_space\_invaders » sert à indiquer le nom de la base de données dans laquelle nous voulons restaurer les données.
- La partie « < sauvegarde.sql » sert à indiquer que nous allons restaurer la base de données indiquée plus tôt dans la commande, à partir du fichier nommé « sauvegarde.sql »

#### F. Liaison C# Base de données

Dans notre projet spicy invader, il est demandé de pouvoir afficher les 5 joueurs ayant le plus haut score et que le joueur puisse en fin de partie ajouter son score avec son pseudo. Pour cela nous avons besoin de connecter notre code C# à notre base de données MySQL, afin d'y faire des requêtes et de récupérer ou insérer les données que nous avons besoin.

Pour pouvoir intégrer des requêtes et des commandes MySQL dans notre code, il faut installer une extension spécifique qui répond à nos besoins. Pour cela suivez ces étapes :

1. Cliquer avec le bouton droit sur votre projet, puis sélectionnez "Gérer les packages NuGet pour la solution...".
2. Dans la fenêtre, recherchez l'extension "MySQL.Data".
3. Après l'avoir trouvée, installez cette même extension.

Maintenant que cette extension est installée, nous pouvons simplement ajouter le using ci-dessous pour utiliser les fonctionnalités MySQL dans notre code : « `using MySql.Data.MySqlClient ;` »

Pour finir, nous devons déclarer une variable qui stockera toutes les informations afin d'établir une connexion avec la base de données.

Ces informations sont les suivantes : « L'host du serveur », « le Port du serveur », « le nom de la base de données », « le nom de l'utilisateur » et sont « mot de passe »

Cette variable est souvent nommé "connexionstring". Cette étape est très importante, car elle permettra à notre code de se connecter avec notre base de données MySQL et d'y exécuter des requêtes avec succès.

Voici comment nous l'avons déclaré dans notre code :

```
// Permet de se connecter au serveur
public static string connexionDb = "Server=localhost;Port=6033;Database=db_space_invaders;User=root;Password=root;";
```

## POO – Programmation orientée objet

### A. Introduction

Dans ce rapport, nous allons voir en détail le projet de recreation du jeu Space Invaders. L'approche retenue repose sur la programmation orientée objet (POO), l'utilisation de classes, l'héritage et la création d'une structure de moteur de jeu. Ces éléments sont essentiels pour le bon développement du jeu.

La POO est la base du projet, elle permet de représenter chaque élément du jeu comme un objet, avec des propriétés et des comportements spécifiques. L'utilisation de classes nous permet de modéliser des entités telles que les vaisseaux spatiaux, les aliens et les projectiles. L'héritage est employé pour créer des hiérarchies de classes, cela favorise la réutilisation du code et simplifie la gestion des entités.

Le moteur de jeu a aussi une importance capitale pour la gestion du programme. Il prend en charge la gestion des ressources, la boucle du jeu, les événements et les interactions de l'utilisateur. Cette structure nous permet de nous concentrer sur la logique du jeu.

Ce rapport examinera donc en détail ces concepts et expliquera comment ils ont été mis en œuvre dans la version du Space Invaders. Pour ce faire nous allons utiliser les thèmes les suivants : une analyse fonctionnelle, une analyse technique avec des diagrammes de classe, des tests unitaires, l'utilisation de ChatGPT durant le projet, et enfin une conclusion.

### B. Analyse fonctionnelle

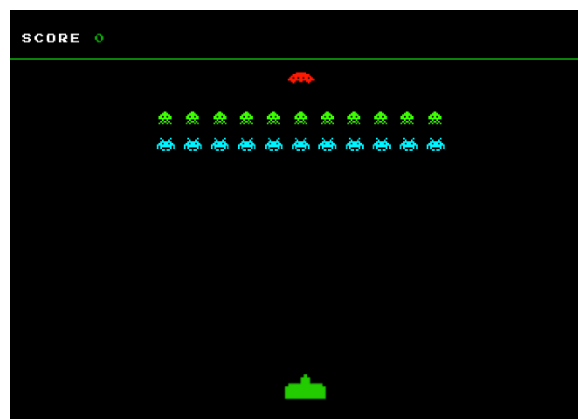
#### Déplacement Joueur

En tant que utilisateur Je veux me déplacer de gauche à droite et de haut en bas Afin d'esquiver les missiles des ennemis

##### Tests d'acceptance:

Déplacement droite	En cours de jeu On appuie sur "d" Le joueur se déplace à droite
Déplacement gauche	En cours de jeu On appuie sur "a" Le joueur se déplace à gauche
Déplacement vers le haut	En cours de jeu On appuie sur "w" Le joueur se déplace vers le haut

Déplacement vers le bas	En cours de jeu On appuie sur "s" Le joueur se déplace vers le bas
Déplacement hors écran droite, gauche	Lorsque que je suis au bord de la console Quand je me déplace à droite ou a gauche encore vers le bord Le joueur est directement téléporté vers l'autre coté de la console
Déplacement hors écran en haut, en bas	Lorsque que je suis au bord de la console Quand je me déplace en haut ou en bas encore vers le bord Le joueur ne fait rien
Affichage Joueur	Dans ma fenêtre visual studio Quand je lance mon programme La fenêtre doit afficher le Playground avec le joueur(voir maquette)



### Ennemi Visible et déplacement

En tant qu'utilisateur Je veux voir mes ennemis bouger	
Tests d'acceptance:	
Ennemi Visible	En cours de partie si je ne fais rien Il y a un ennemi en haut à gauche de l'écran
Ennemi Mouvement	En cours de partie Si je ne fais rien Il y a un ennemi qui bouge jusqu'au coins droite de l'écran
Ennemi Mouvement	En cours de partie quand l'ennemi est au bord de l'écran Si je ne fais rien Il y a un ennemi qui bouge une fois vers le bas
Ennemi Mouvement	En cours de partie quand l'ennemi est tout à droite de l'écran Si je ne fais rien Il y a un ennemi qui bouge jusqu'au coin gauche de l'écran

## Missiles Joueur

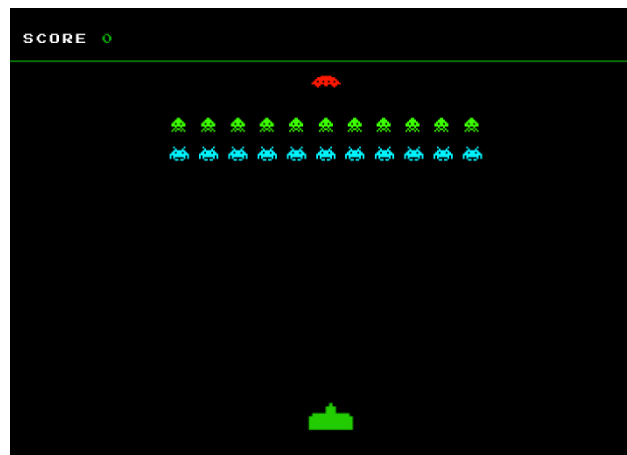
En tant qu'utilisateur Je veux tirer des missiles Afin des détruire les Invaders		
Tests d'acceptance:		
Tire missile joueur	En cours de partie	Quand j'appuie sur la barre d'espace Un missile est créé, il s'affiche et se déplace vers le haut en verticale
Missile joueur	En cours de partie	Quand mon missile atteint la bordure du haut Le missile se désaffiche et se supprime
Missile joueur	En cours de partie	Quand mon missile touche un Invader Le missile se désaffiche, se supprime et l'Invader change d'apparence
Missile joueur	En cours de partie	Quand j'appuie plusieurs fois de suite sur la barre d'espace Un missile est tiré à chaque fois
Tir continu	Dans le jeu,	Quand je maintiens la barre d'espace pressée Un missile est tiré à chaque cycle

## Missiles Invaders

En tant qu'utilisateur Je veux que les invaders tires des missiles		
Tests d'acceptance:		
Tire Missile invader	En cours de partie avec un seul invader restant	Si je ne fais rien L'invader tire des missiles a des moments aléatoires
Missile invader	En cours de partie	Quand un missile d'invaders atteint la bordure du bas Le missile se désaffiche et se supprime
Missile invader	En cours de partie	Quand le missile d'un l'invader touche le joueur Le missile se désaffiche, se supprime et le joueur change d'apparence
Tir des arrières	En cours de partie	Quand un alien "arrière" tire un missile Ce dernier traverse les autres aliens sans les toucher

## Affichage score

En tant qu'utilisateur Je veux qu'un score s'affiche Afin de savoir si j'ai battu mon meilleur score	
Tests d'acceptance:	
Affichage score	Dans ma fenêtre visual studio Quand je lance mon programme La fenêtre affiche le Playground avec le score(voir maquette)
Update score	En cours de partie Quand je tue un invader Le score s'actualise et affiche le nombre de point actuel
Reset score	Au début de partie Si je ne fais rien Le score est à 0



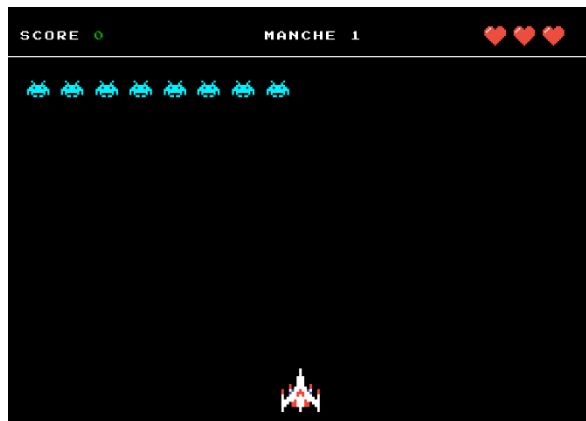
## Affichage nb manche

En tant qu'utilisateur Je veux qu'un nombre de manche s'affiche Afin de savoir si j'ai battu le nombre de manche effectué précédemment	
Tests d'acceptance:	
Update nb manche	En cours de partie Quand je tue tous les invaders La manche augmente de 1
Reset nb manche	Au début de partie Si je ne fais rien Le nb de manche est à 0



## Spawn ennemis

En tant qu'utilisateur Je veux que les ennemis spawn en continue Afin de pouvoir les abattre		
Tests d'acceptance:		
Spawn des ennemis	Lors d'une partie	Quand une manche est finie 8 nouveaux ennemis apparaisse ( voir maquette )



## Menu principale du jeu

En tant qu'utilisateur Je veux avoir un menu Afin de pouvoir me déplacer dans les différentes sous menu		
Tests d'acceptance:		
Affichage menu	Dans ma fenêtre visual studio	Quand je lance mon programme La fenêtre affiche le menu principale (voir maquette)
Changement d'apparence sélection	Dans le menu principale	Quand je sélectionne un sous menu Le sous menu sélectionner change de couleur



## Replay ou Quitter

En tant qu'utilisateur Je veux pouvoir relancer une partie Afin de pouvoir rejouer après une fin partie		
Tests d'acceptance:		
Recommencer	Dans le menu de fin	Quand je lance l'option replay Le jeux recommence
Quitter	Dans le menu de fin	Quand je lance l'option quitter Le jeux se termine

## Sauvegarder la partie

En tant qu'utilisateur Je veux sauvegarder le score de ma partie Afin de pouvoir connaitre mes anciens score		
Tests d'acceptance:		
Affichage ssave	Dans ma fenêtre du jeu	Quand je rentre dans le sous menu save La fenêtre affiche son interface(voir maquette)*
Score sauvegarder	Dans le jeu	Quand je vais dans l'onglet score Cela m'affiche la liste des meilleurs score effectué



## Menu fin de partie

En tant qu'utilisateur Je veux avoir un menu de fin Afin de pouvoir rejouer, quitter et sauvegarder la partie

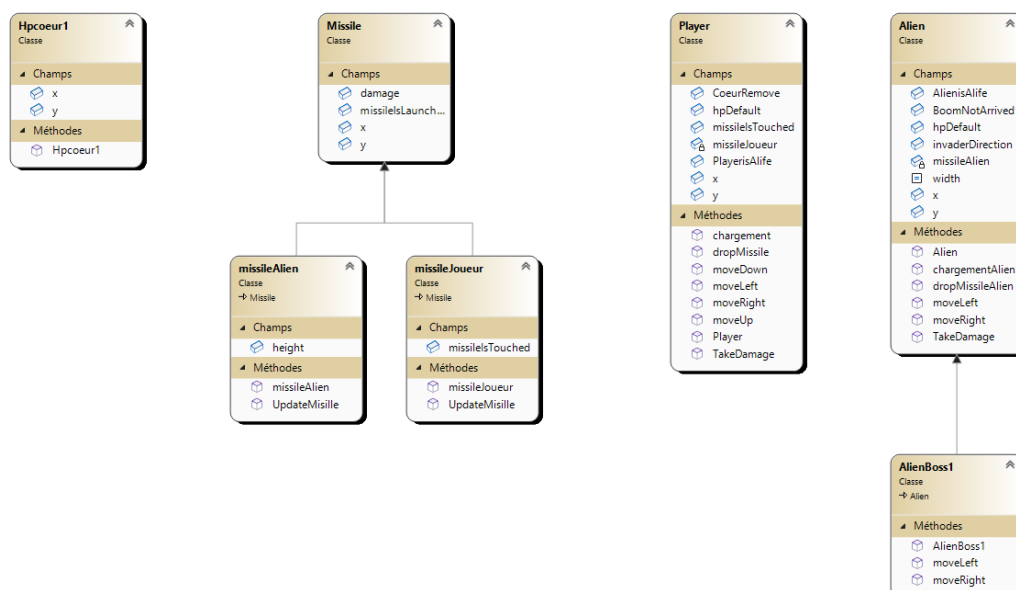
## Tests d'acceptance:

Affichage menu game over	Dans ma fenêtre de jeu Quand je perd une partie La fenêtre affiche le menu game over (voir maquette)
Changement d'apparence	Dans les menus de fin de partie Quand je sélectionne un sous menu Le sous menu sélectionner change de couleur
Affichage menu game win	Dans ma fenêtre de jeu Quand je gagne une partie La fenêtre affiche le menu game win (voir maquette)



## C. Analyse technique

Diagramme de classe :



Explication docfx :

Les explications concernant les méthodes ont directement été générées par docfx, nous le retrouverons en annexe dans le dossier documents.

#### D. Tests Unitaire

Introduction :

Les tests unitaires permettent de garantir la fiabilité du code, ils vérifient que chaque partie du code fonctionne correctement et soit compatible avec le reste. Dans notre projet, ces tests sont d'une grande importance pour assurer la cohérence et la sûreté de notre jeu.

Test Constructor Alien / Player :

Les tests suivants vérifient si les données qui sont attribuées à l'objet sont effectivement prises en compte. Cela permet d'éviter la prise en compte de données indésirées.

```
[TestMethod()]
public void AlienTest()
{
    // Arrange
    int x = 1;
    int y = 2;
    int hpDefault = 100;

    // Act
    Alien alien = new Alien(x, y, hpDefault);

    // Assert
    Assert.AreEqual(x, alien.x);
    Assert.AreEqual(y, alien.y);
    Assert.AreEqual(hpDefault, alien.hpDefault)
}
```

```
[TestMethod()]
public void PlayerTest()
{
    // Arrange
    int x = 50;
    int y = 34;
    int hpDefault = 50;

    // Act
    Player player = new Player(x, y, hpDefault);

    // Assert
    Assert.AreEqual(x, player.x);
    Assert.AreEqual(y, player.y);
    Assert.AreEqual(hpDefault, player.hpDefault);
}
```

### Test Move Alien gauche / droite :

Les tests suivants vérifient que l'Alien se déplace correctement horizontalement en modifiant sa position x et dès qu'il atteint les bords, on vérifie qu'il se déplace en vertical en modifiant sa position y. Nous avons mis ces tests en un seul, mais on peut aussi les diviser en deux tests. Un test qui vérifiera le déplacement horizontal et un autre pour vérifier le décalage vertical.

```
[TestMethod()]
public void moveLeftTest()
{
    // Arrange
    Alien alien = new Alien(137, 2, 100);
    alien.invaderDirection = false;
    int x = alien.x;
    int y = alien.y;

    // Act
    alien.moveLeft();

    // Assert

    Assert.AreEqual(x - 1, alien.x);

    if (alien.x == 1)
    {
        Assert.AreEqual(y + 4, alien.y);
        Assert.IsTrue(alien.invaderDirection);
    }
}
```

```
[TestMethod()]
public void moveRightTest()
{
    // Arrange
    Alien alien = new Alien(1, 2, 100);
    alien.invaderDirection = true;
    int x = alien.x;
    int y = alien.y;
    int width = Alien.width - 13;

    // Act
    alien.moveRight();

    // Assert
    Assert.AreEqual(x + 1, alien.x);

    if (alien.x == width)
    {
        Assert.AreEqual(y + 4, alien.y);
        Assert.IsFalse(alien.invaderDirection);
    }
}
```

### Test UpdateMissile Alien / Player :

Les tests suivants vérifient que le missile augmente correctement la position y, lorsque le missile est en cours de déplacement.

```
[TestMethod()]
public void UpdateMissileTest()
{
    // Arrange
    missileAlien missile = new missileAlien(50);
    int y = missile.y;
    missile.missileIsLaunched = true;

    // Act
    missile.UpdateMissile();

    // Assert
    Assert.AreEqual(y + 1, missile.y);
}
```

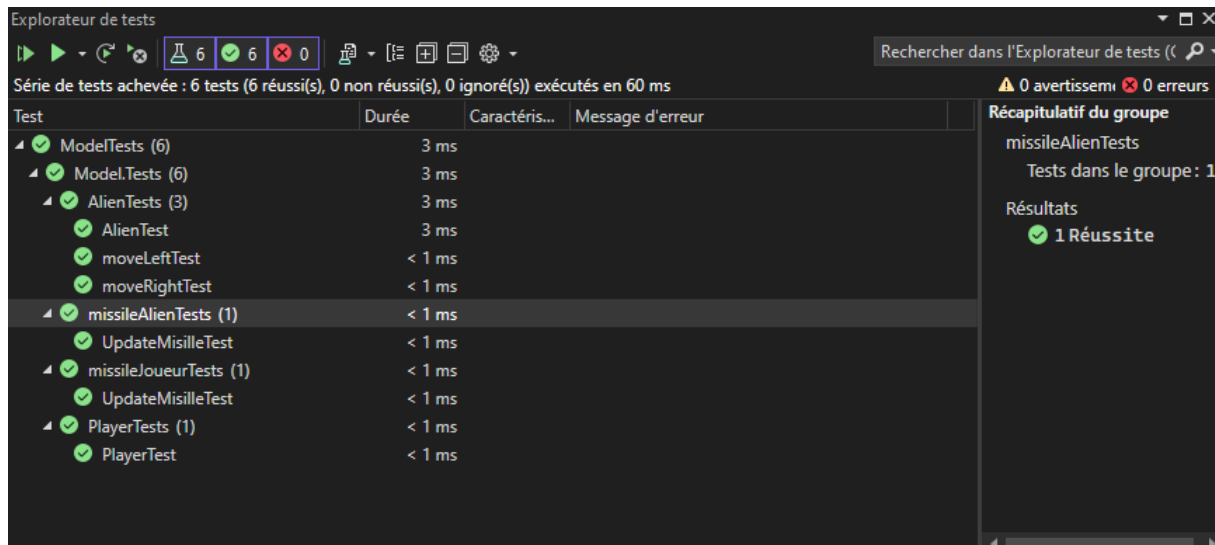
```
[TestMethod()]
public void UpdateMissileTest()
{
    // Arrange
    missileJoueur missile = new missileJoueur(50);
    int y = missile.y;
    missile.missileIsLaunched = true;

    // Act
    missile.UpdateMissile();

    // Assert
    Assert.AreEqual(y - 1, missile.y);
}
```

Test réussi :

Voici la preuve que tous nos tests que nous avons effectué un peu plus haut ont réussi :



#### E. Utilisation ChatGPT

Durant le projet, j'ai utilisé plusieurs fois ChatGPT pour résoudre des problèmes liés à mon code. ChatGPT m'a aidé lorsque j'avais des erreurs dans mon code, que je n'arrivais pas à résoudre ou qui me semblait incompréhensible. Dans ces cas, j'utilisais ChatGPT pour les identifier et les comprendre.

Par exemple, j'ai dû à un moment utiliser une boucle "for" à l'envers pour parcourir mes listes dans leurs sens inverse. Étant donné que cela était nouveau pour moi j'ai recouru à ChatGPT m'aider et m'expliquer à mettre en place cette boucle "for" inversée. Le deuxième cas où je l'ai utilisé, c'est quand j'ai dû connecter le code C# et la base de données. Pour cela ChatGPT m'a donné un petit tutoriel sur la mise en place de cette connexion., cela m'a permis de faire cette même connexion et il m'a fait comprendre comment elle fonctionnait.

En résumé, ChatGPT m'a aidé pour résoudre des erreurs dans mon code et pour m'apprendre et me faire comprendre des choses que je n'avais jamais faites.

#### F. Conclusion

Pour conclure ce projet m'a permis de développer mes compétences en programmation orientée objet, il m'a permis de comprendre l'importance de ce concept dans la construction de jeux vidéo ou encore de logiciel. Il m'a aussi fait découvrir comment effectuer la liaison entre mon code en C# et une base de données. Cette chose qui m'était totalement inconnu et que je n'aurais jamais pensé faire.

En regardant les choix que j'ai effectué durant mon projet, je pense que des améliorations peuvent être apportées comme l'utilisation de GitHub plus tôt dans mon projet. En effet au début de mon projet je n'utilisais pas l'outil GitHub pour stocker mon programme, par la suite j'ai commencé à l'utiliser et je me suis alors rendu compte de son importance.



Si j'avais fait cela plus tôt cela aurait certainement simplifié la gestion du code de mon projet. Une autre chose que je pourrais améliorer, c'est la création de "user stories" avant de commencer à coder. Cela m'aurait certainement permis de pouvoir mieux m'organiser et de mieux planifier le projet.

Pour finir ce projet a été une bonne expérience, il m'a permis de développer mes compétences techniques et mes compétences de gestion de projet. Je pense que si ce projet était à refaire je le referais avec plaisir, je changerais seulement ma façon travailler avec GitHub et IceScrum.