



ECOLE TECHNIQUE
ECOLE DES MÉTIERS LAUSANNE

« SPICY INVADER »

Rapport projet : P_POO, P_UX & P_DB

Faire un réplica du célèbre jeu Space Invader



ADRIAN TOLEDO

28 AOÛT 2023

ADRIAN TOLEDO

Enseignant : Xavier Carriel, Aurélie Curchod & Mathieu Meylan

Table des matières

1. UX – Expérience utilisateur.....	2
1.1. Introduction	2
1.2. Analyse	2
1.3. Conception	6
2. BD - Basse de données.....	20
2.1. Introduction DB « db_space_invaders »	20
2.2. Importer les données	21
2.3. Gestion des utilisateurs	22
2.4. Requêtes de sélection.....	24
2.5. Index	29
2.6. Backup/Restore	31
2.7. MySQL – db_sapce_invaders danse le Highscore	31
3. POO - Programmation orientée objet.....	34
3.1. Introduction	34
3.2. Analyse fonctionnelle	34
3.3. Analyse technique.....	38
3.3.1. Diagramme de basse.....	38
3.3.2. Explication	38
3.4. Test	38
3.4.1. Unit Testing	38
3.4.2. Rapport de test.....	39
4. Conclusions	42
4.1. UX	42
4.2. DB.....	42
4.3. POO	42
5. Sources.....	43
6. Annex.....	44

1. UX – Expérience utilisateur

1.1. Introduction

UX, l'expérience de l'utilisateur, ce terme englobe deux éléments principaux, l'utilisateur et le produit, où l'utilisateur exprime son sentiment, sa satisfaction et sa sensation lors de l'utilisation d'un produit. L'UX étudie reconnaître les points d'utilisation afin d'adapter au mieux le produit à l'utilisateur.

Aujourd'hui, en informatique, l'UX est étudié et utilisé dans une large mesure pour représenter l'ensemble de l'interface et de l'interaction entre les utilisateurs et le programme, la page web ou les éléments informatique. En définitive, cette partie de l'utilisation et de la mesure de l'UX implique la création d'interfaces qui facilitent l'utilisation par les utilisateurs et les guident dans l'accomplissement du service auquel ils sont destinés. Ce terme est appelé UI, l'interface utilisateur.

Dans ce projet, l'étude de l'UX et de l'UI dans le monde informatique a été mise en œuvre pour créer de manière réaliste les menus et l'interface du jeu vidéo SpacyInvaders. La création de ces menus, les outils utilisés et les choix effectués pour créer un menu pour les futurs joueurs de ce jeu seront détaillés ci-dessous.

1.2. Analyse

Dans cette section, les créations et concepts préalables à la création de l'interface du jeu vidéo seront détaillés et analysés.

1.2.1. Persona

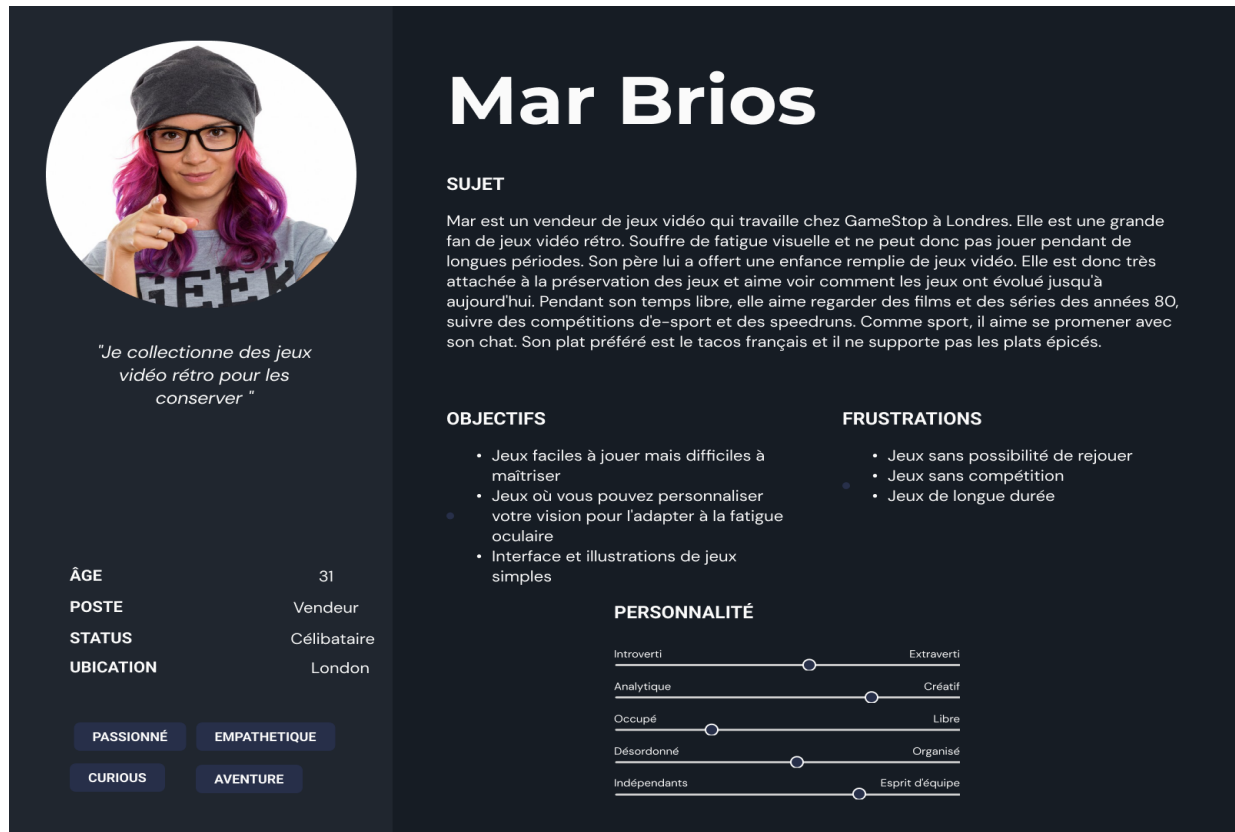
Pour commencer, un des éléments les plus importants pour la création d'un produit ou d'une interface est l'utilisateur. Une technique pour connaître nos utilisateurs est détailler le profil de l'utilisateur prévu. Cette méthode peut nous aider à voir l'efficacité de notre produit et à l'utiliser comme guide pour suivre un concept qui leur conviendra.

Ce processus, de création de Persona, consiste à créer des profils complets et détaillés des futurs utilisateurs potentiels. Ces profils doivent respecter des caractéristiques de base telles que l'identité de l'utilisateur, ses origines et sa localisation, son statut actuel et sa personnalité, et surtout, présenter ses centres d'intérêt et les points qu'il n'aime pas.

Deux profils Persona sont présentés ci-dessous :

- Persona 1 : Mar Brios

Mar Brios représente le profil d'une fan de jeux vidéo, en particulier de jeux rétro. C'est une jeune fille handicapée visuelle. Son principal intérêt pour les jeux vidéo est la simplicité, la rejouabilité et la possibilité d'adapter leur interface visuelle à son handicap.



The profile card for Mar Brios features a circular profile picture of a young woman with pink hair and glasses. To the right of the photo is her name 'Mar Brios' in large white text. Below the name is a 'SUJET' (Subject) section with a paragraph about her background as a video game seller and her visual impairment. To the left of the main text is a quote: 'Je collectionne des jeux vidéo rétro pour les conserver'. Below the quote are her personal details: Age 31, Post (Seller), Status (Single), and Location (London). At the bottom left are four interest tags: PASSIONNÉ, EMPATHETIQUE, CURIOUS, and AVENTURE. To the right of the main text are two sections: 'OBJECTIFS' (Goals) and 'FRUSTRATIONS' (Frustrations), each with a bulleted list. At the bottom right is a 'PERSONNALITÉ' (Personality) section with six horizontal sliders showing her traits: Introverti (towards Extraverti), Analytique (towards Créatif), Occupé (towards Libre), Désordonné (towards Organisé), Indépendants (towards Esprit d'équipe).

Mar Brios

SUJET

Mar est un vendeur de jeux vidéo qui travaille chez GameStop à Londres. Elle est une grande fan de jeux vidéo rétro. Souffre de fatigue visuelle et ne peut donc pas jouer pendant de longues périodes. Son père lui a offert une enfance remplie de jeux vidéo. Elle est donc très attachée à la préservation des jeux et aime voir comment les jeux ont évolué jusqu'à aujourd'hui. Pendant son temps libre, elle aime regarder des films et des séries des années 80, suivre des compétitions d'e-sport et des speedruns. Comme sport, il aime se promener avec son chat. Son plat préféré est le tacos français et il ne supporte pas les plats épicés.

OBJECTIFS

- Jeux faciles à jouer mais difficiles à maîtriser
- Jeux où vous pouvez personnaliser votre vision pour l'adapter à la fatigue oculaire
- Interface et illustrations de jeux simples

FRUSTRATIONS

- Jeux sans possibilité de rejouer
- Jeux sans compétition
- Jeux de longue durée

PERSONNALITÉ

Trait	Value
Introverti	60%
Analytique	70%
Occupé	30%
Désordonné	50%
Indépendants	80%
Extraverti	40%
Créatif	30%
Libre	70%
Organisé	50%
Esprit d'équipe	20%

AGE 31
POSTE Vendeur
STATUS Célibataire
UBICATION London

PASSIONNÉ **EMPATHETIQUE**
CURIOUS **AVENTURE**

"Je collectionne des jeux vidéo rétro pour les conserver"

Figure 1 – Mar Brios [en la página 44](#)

De ce profil, nous pouvons extraire des points importants pour la création de notre jeu. Par exemple, connaître leur intérêt pour la personnalisation de l'interface. Ça nous donne en conséquence plus de ressources sur l'accessibilité de l'interface et ses couleurs. D'autres points, tels que la rejouabilité et la compétitivité, nous donnent l'idée de vouloir enregistrer des informations telles que le score à comparer.

- Peronsa 2 : Zel Workukun

Ce deuxième profil est une personnes âgées dont la préoccupation est l'accessibilité aux jeux. Zel Workukun est un vieux retraité dont la vie tourne autour des jeux vidéo et de leur utilisation dans les salles d'arcade. Il essaie donc d'en profiter le plus possible malgré les problèmes d'arthrite dans l'une de ses mains et sa perte de vision due à l'âge. Il déteste également les jeux où la coopération n'est pas possible, car il avait l'habitude de jouer en compagnie de son partenaire décédé.

Zel Workukun

SUJET

Zel Workukun est un ancien employé de Nintendo. Il était principalement directeur d'usine pour le développement de cartouches de jeux vidéo pour la Nintendo Switch. Il est à la retraite depuis un an et recherche maintenant la paix et la tranquillité avec ses chiens. Son premier objectif en tant que retraité est de créer un espace de jeu d'arcade dans sa maison. En raison de son âge et de son excès d'efforts au travail, il souffre d'arthrite à la main droite et a du mal à lire les petits caractères. En plus des jeux vidéo, il aime lire sur les nouvelles technologies et déguster un bon ramen.

OBJECTIFS

- Jeux faciles à contrôler (avec un main)
- Les jeux devraient être disponible en gros caractères.
- La possibilité de personnaliser les personnages

FRUSTRATIONS

- N'aime pas les écrans de chargement
- Ne pas pouvoir terminer un jeu en raison de sa difficulté
- Les jeux vidéo qui ne sont pas possible cooperative

PERSONNALITÉ

Introverti	Extraverti
Analytique	Créatif
Occupé	Libre
Désordonné	Organisé
Indépendants	Esprit d'équipe

ÂGE 64

POSTE Retraité

STATUS Veuf

UBICATION Tokyo, Japon

ANALYTIQUE **EMPATHETIQUE**

ORGANISE **CRITIQUE**

"La salle d'arcade est le meilleur moyen de jouer à des jeux vidéo."

Figure 2 Zel Workukun [en page 45](#)

Cet exemple de Zel Workukun nous montre l'importance de la diversité des joueurs possibles, d'âges et de sexes différents. Workukun implique de penser aux personnes âgées, dont la seule préoccupation est de pouvoir profiter des jeux vidéo seul ou avec des amis. En outre, un point important est l'accessibilité du jeu, tant sur le plan visuel qu'en termes de contrôle. Ainsi, la création d'une interface avec des lettres plus grandes et/ou le contrôle du jeu d'une seule main peut impliquer un plus grand plaisir pour l'utilisateur. Enfin, la manière d'utiliser le jeu est également important, car ce profil aimerait profiter des jeux sur leurs vieilles machines d'arcade.

1.2.2. Color Palletes

Un autre point important pour la création de produits ou d'interfaces est l'utilisation des couleurs. En se basant sur la théorie des couleurs, savoir choisir une palette de couleurs est d'une grande importance, car les utilisateurs identifieront le produit en fonction de leur premier contact visuel.

En plus d'offrir une identification par sa couleur, il est également possible d'utiliser ces couleurs pour signaler des éléments, identifier des informations ou différencier différents points affichés.

Palette 1 :

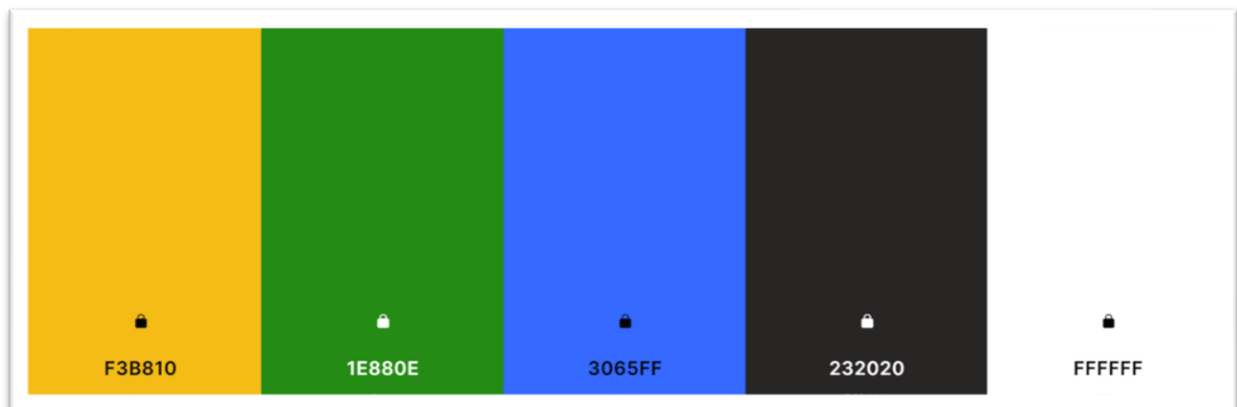


Figure 3 Palette de couleurs 1

A partir de cette palette, nous présentons les principales couleurs qui seront utilisées et mélangées pour l'élaboration de l'interface.

La couleur principale est le jaune F3B810 associé au vert 1E880E. Ces deux éléments seront utilisés pour concrétiser les éléments visuels et interactifs. Le bleu 3065FF sera la couleur utilisée pour l'alternance. Les deux dernières, le noir 232020 et le blanc FFFFFF, seront utilisées pour contraster et donner plus d'amplitude aux couleurs.

Palette 2 Monochrome:



Figure 4 Palette de couleurs 2

La première couleur, F3B810, est une couleur jaunâtre qui sera utilisée comme couleur prédominante pour l'interface et les éléments visuels. Il est donc nécessaire d'utiliser une autre palette où cette couleur est exposée dans différentes tonalités.

Cette couleur a été choisie comme couleur prédominante parce qu'elle exprime un sentiment d'illusion, d'avertissement, d'amusement et d'énergie. Ainsi, l'utilisateur pourra avoir comme référence frappante l'idée d'un jeu spatial explosif.

1.2.3. Accessibilité

C'est est vrai que si un produit est destiné à des utilisateurs, le premier point important sera de recueillir plus d'utilisateurs et grâce à l'utilisation de techniques et d'éléments d'accessibilité, nous pouvons nous assurer que ce produit peut être utilisé par un large éventail de profils.

Après la création des profils Persona, j'ai voulu mettre en place les éléments suivants qui peuvent garantir la satisfaction et l'utilisation du jeu vidéo :

- Changer la taille de la typographie : Dans les paramètres, il sera possible de changer la taille de la typographie pour la rendre plus grande ou plus petite.
- Personnaliser les couleurs de l'interface, des joueurs et des ennemis : la couleur par défaut est le jaune, mais l'utilisateur pourra modifier ces couleurs à volonté afin de trouver la combinaison de couleurs qui lui convient le mieux. Les couleurs disponibles sont le jaune, le vert, le bleu et le rouge.
- Contrôles simples : Enfin, la navigation dans les menus, la sélection des options et le contrôle du jeu sont simplifiés afin de pouvoir être utilisés uniquement avec un joystick ou un joystick à un bouton. Cette option offre la possibilité d'utiliser l'accessibilité ou des contrôles simples de type arcade.

1.2.4. Eco-conception

De nos jours, il est bien connu que la technologie informatique et son utilisation impliquent la consommation d'énergie électrique et ont donc un impact direct sur l'environnement. Grâce à son étude, il a été possible d'établir différents éléments qui peuvent contribuer à rendre les pages web, les programmes et les jeux vidéos plus durables pour nous et pour l'environnement.

Pour ce projet, diverses technologies ont été appliquées afin de réduire la consommation d'énergie de l'interface. Voici la liste des éléments eco-conception:

- Assets réutilisables : utilisation répétitive d'un même élément visuel comme les panneaux d'affichage d'informations.
- Taille d'écran adaptable : cela implique d'adapter le jeu en fonction de l'utilisation et de la taille de l'écran, optimisant ainsi son utilisation.
- Sons : avoir la possibilité de désactiver les sons du jeu et pouvoir utiliser un élément peu nécessaire.
- Pas de connexion internet : comme il s'agit d'un jeu où l'accès à internet n'est pas nécessaire, ce jeu présente une simplicité dans son code et son gameplay.
- Base de données limitée et simple : comme il fonctionne localement, sa base de données est simple et n'est modifiée qu'au fur et à mesure que le joueur joue et enregistre des informations limitées comme les 5 meilleurs coups.
- Aucune animation : aucune animation visuelle n'est utilisée pour le menu, seulement des images statiques qui peuvent changer de couleur.

1.3. Conception

Dans ce chapitre, nous présentons la maquette conçue pour la création de l'interface du jeu vidéo Spicy Invaders. L'élaboration de la conception ainsi que les

choix des éléments visuels seront détaillées. Les maquettes sont divisées en deux parties, la première montre des maquettes Low Fidelity où les éléments essentiels sont montrés et ensuite, les maquettes High Fidelity comme un résultat visuel de la représentation finale du jeu vidéo.

Pour chaque concept de jeu, les éléments présentés et leur contrôle ont été simplifiés autant que possible. En plus, chaque élément cliquable sera toujours situé sur le côté gauche de l'écran, et son information ou son résultat sur le côté droit de l'écran.

1.3.1. Low Fidelity

- Menu

Le menu présente 4 boutons sélectionnables avec le titre du jeu accompagné d'une image. Les 4 boutons permettent d'accéder au jeu en solo, au jeu en multijoueur, à la liste des meilleurs scores et aux options du jeu.



Figure 5 LF - Menu

- New Game

Lorsque vous jouez seul, la salle de jeu est créée et c'est possible sélectionner les options suivent : le nom du joueur en sélectionnant les lettres de l'alphabet avec les flèches, le vaisseau du joueur, son arme, la carte avec les ennemis différents débloquées, la difficulté et le bouton pour commencer à jouer o retourner. Pour chaque élément sélectionné, le niveau et la puissance seront affichés visuellement sur le côté droit de l'écran.

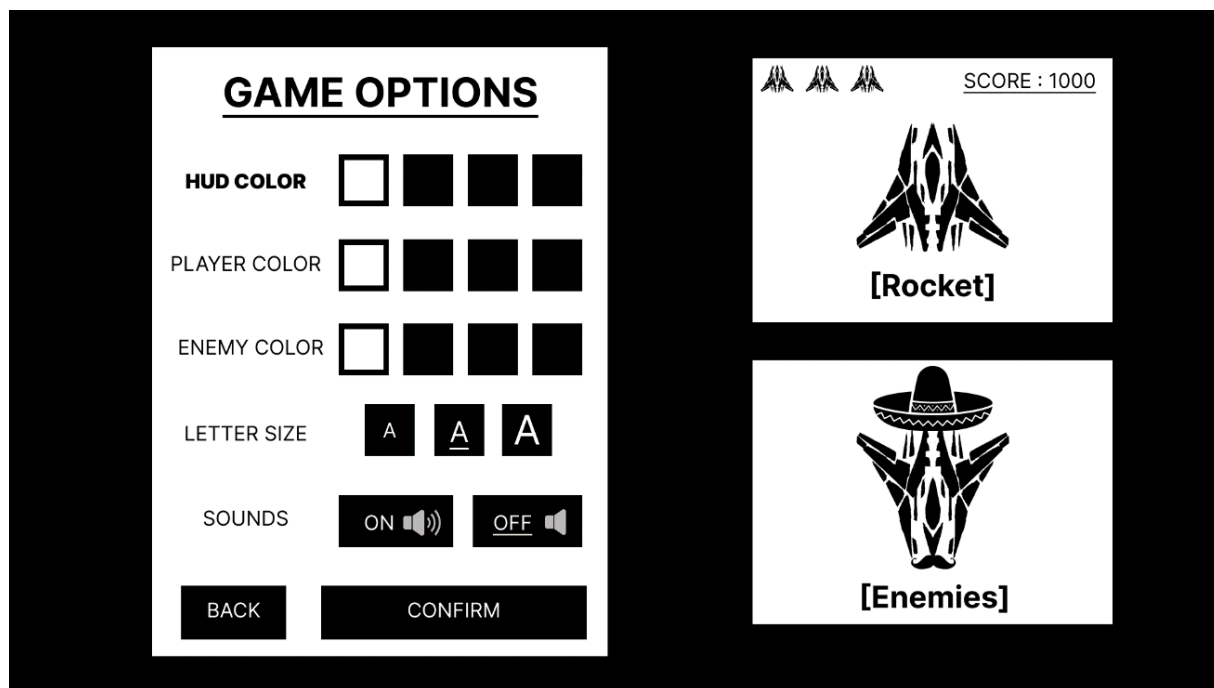


Figure 6 LF - New Game

- Multiplayer

Dans l'écran multijoueur, le joueur pourra ajouter un nouveau joueur avec un nom différent et sélectionner chaque élément pour chaque joueur.

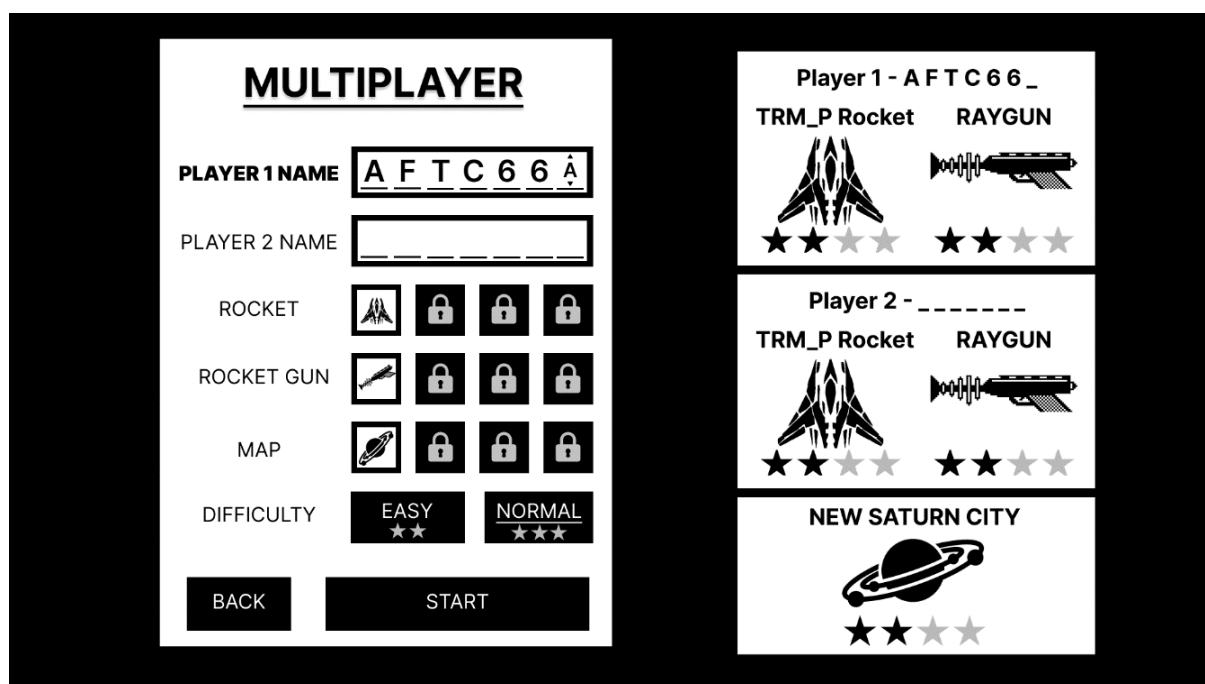


Figure 7 LF – Multiplayer

- Highscore

Cet écran est destiné à enregistrer les 5 meilleurs scores du jeu. Il enregistrera le nom, le score et sera trié par position décroissante. De plus, sur le côté droit, chaque profil sera affiché avec son navire.

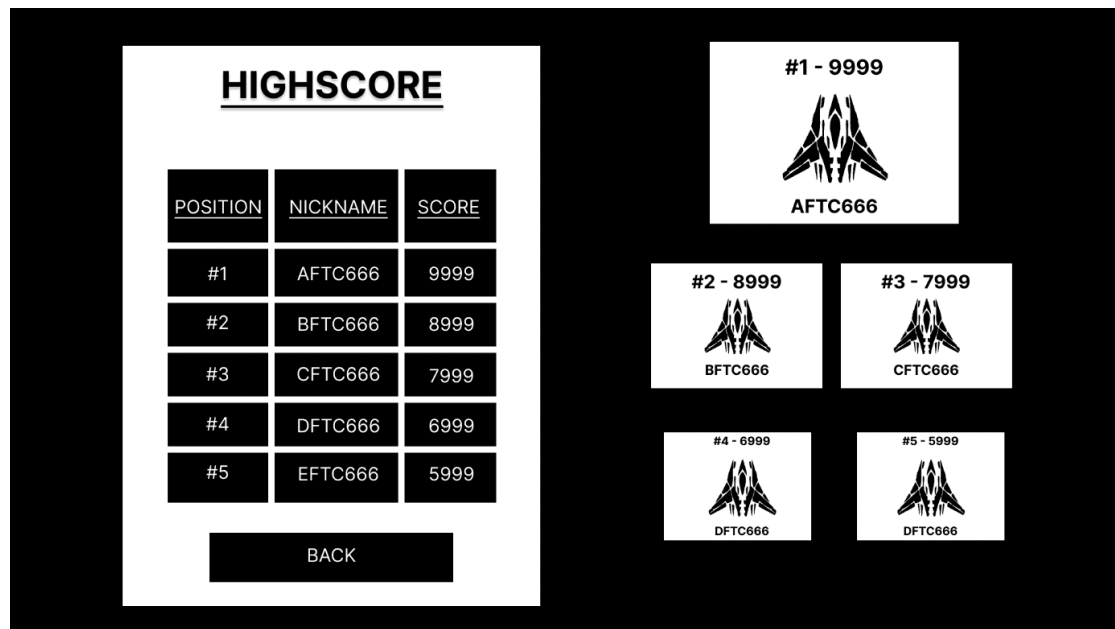


Figure 8 LF - Highscore

- GameOptions

Dans l'écran des options, vous pourrez personnaliser l'expérience de jeu en fonction des couleurs de l'interface, du joueur ou des ennemis. La taille des lettres, leur son et le réajustement de la taille de l'écran. Sur la droite, vous pouvez voir comment chaque changement se affichera dans le gameplay du jeu.

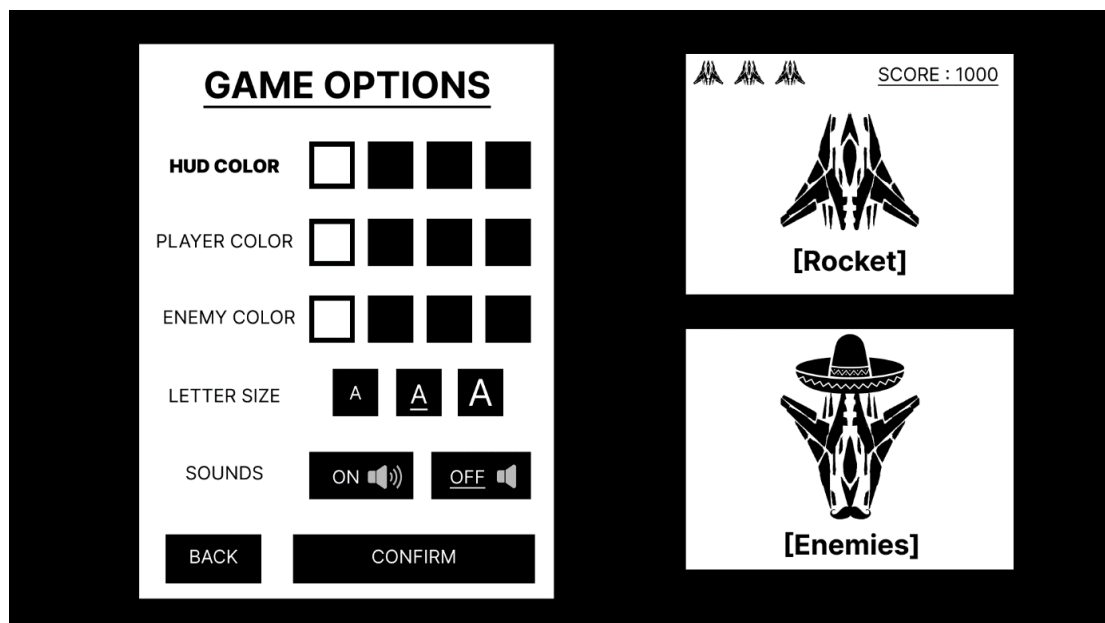


Figure 9 Game Options

- Assets

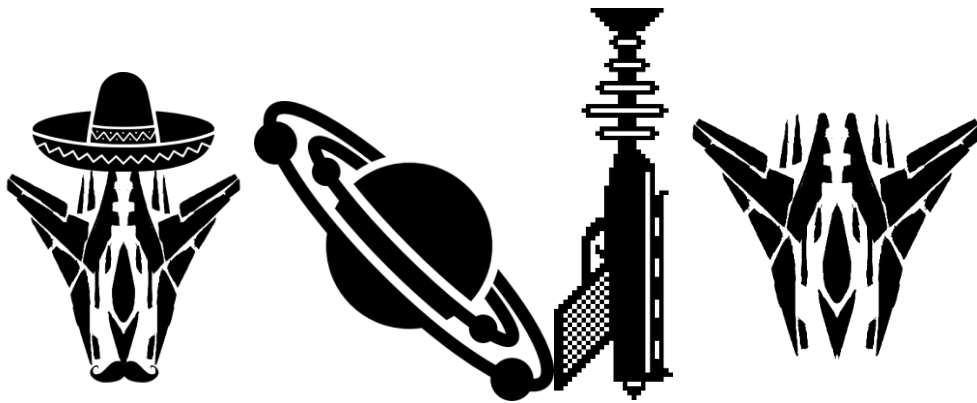


Figure 10 LF - Assets

1.3.2. High Fidelity

Pour l'élaboration de la High-Fidelity, j'ai utilisé des assets et des images de tiers sans droits d'auteur pour une utilisation libre. Ces maquettes ont été construites à partir de la Low-Fidelity en ajoutant des éléments finaux tels que les panneaux, la typographie, les images et les couleurs, tout en respectant les espaces de l'écran et leur intégration avec chaque élément.

- Assets HUD :

Pour différencier les éléments cliquables à gauche et les éléments statiques à droite, deux types de panneaux ont été utilisés, où l'utilisateur pourra interagir et, à droite, un autre type de panneau où seules les images ou le texte du jeu seront affichés.



Figure 11 HF - HUD Assets

- Typographie :

Il convient de noter que le choix du style de caractères a été fait pour représenter les textes des films sur l'espace des années 80. Cette police Laser Corps a été utilisée de différentes manières pour différencier les textes de chaque élément :

- Titre du jeu



Figure 12 Typographie - Titre du jeu

- Titre du panel



Figure 13 Typographie - Titre du panel

- Titre des options modifiables



Figure 14 Typographie - Options

- Informations sauvegarder



Figure 15 Typographie - Info

- Informations dans le jeu



Figure 16 Typographie - Dans le jeu

- Assets dans le jeu :

Suite, une liste d'images utilisées pour les vaisseaux spatiaux, les armes et les cartes

- Vaisseaux spatiaux



Figure 17 Vaisseaux spatiaux

- Armes, Cartes et effets visuels

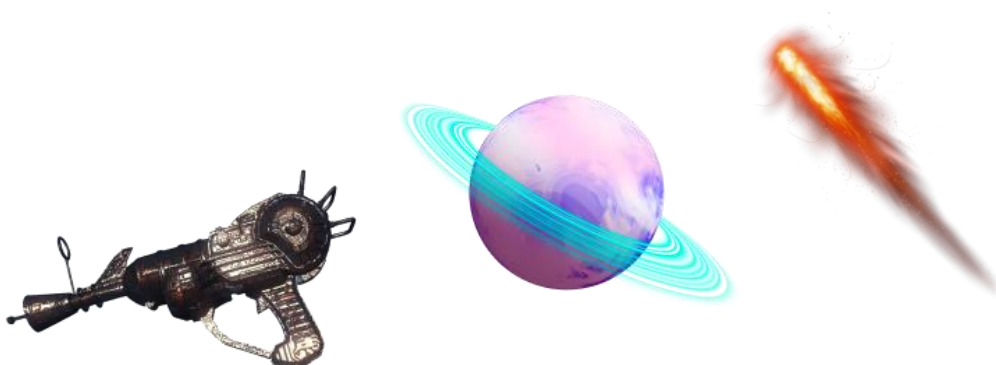


Figure 18 HF - Assets divers

- Background

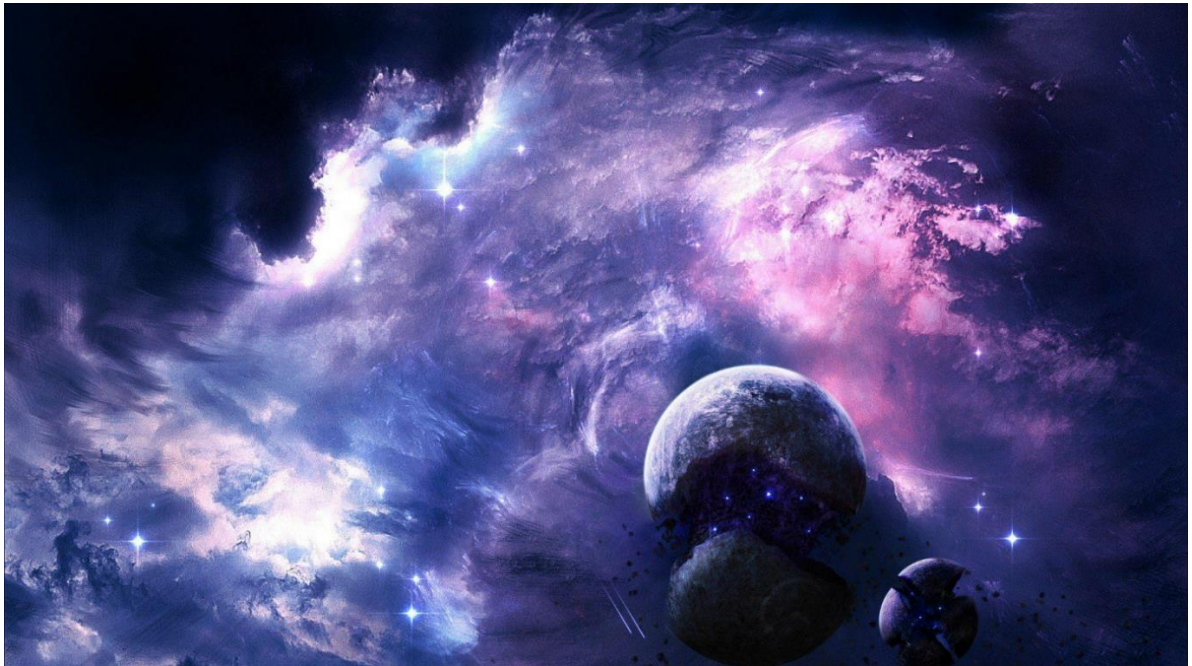


Figure 19 HF - Background

- Ecrans et menus :
 - Menu d'accueil

Dans ce menu, nous pouvons mettre en évidence la typographie du titre du jeu, accompagnée d'un vaisseau jouable du jeu. À gauche se trouvent les options cliquables où chaque élément est placé à l'intérieur d'un panneau et donne accès à la option o sortir du jeu.



Figure 20 HF - Menu d'accueil [en page 46](#)

- New Game

Dans l'écran du nouveau jeu, un titre pour le panneau "NEW GAME" est présenté où chaque élément à personnaliser est dans des carrés. La typographie blanche/verte représente les options possibles à modifier, telles que Pseudo ou Rocket, et le texte statique est confirmé en vert.

Sur la droite, il est montré dans différents panneaux, seulement l'information visuelle de chaque élément, avec la typographie qui sera appliquée à l'intérieur du jeu.



Figure 21 HF - New game [en page 47](#)

- Multiplayer

Le multijoueur respecte les mêmes éléments et choix que la section précédente, en ajoutant simplement un espace pour le deuxième joueur.



Figure 22 HF – Multiplayer [en page 48](#)

- Highscore

Sur cet écran, les informations de les 5 mieux match sont affichées en vert et à droite, un panneau pour chaque joueur avec son nom et son navire.



Figure 23 HF – Highscore [en page 49](#)

- Game Options

Enfin, cet onglet présente toutes les options finales et la manière dont leurs modifications seront appliquées au jeu dans le panneau de droite.



Figure 24 Game Options [en page 50](#)

1.4. Test

1.4.1. Test d'utilisabilité

Ci-dessous un tableau avec les activités et les tests possibles qui peuvent être réalisées lors de l'utilisation de l'interface.

Menu	Description	Conditions	Étapes de Test	Données	Résultat attendu
Menu d'accueil	Les boutons présélectionnés sont visuellement différenciés	Utilisation de la version High Fidelity dans Figma	1. Déplacez la souris ou utilisez les touches fléchées sur un bouton sans y accéder. 2. Passer à un autre bouton	-	Les boutons changent de couleur pour indiquer l'option présélectionnée.
Menu d'accueil	Les boutons sélectionnés permettent d'accéder au menu correspondant	Utilisation de la version High Fidelity dans Figma	1. Déplacez la souris ou utilisez les touches fléchées sur un bouton. 2. Sélectionner le bouton à l'aide de la touche Entrée ou cliquer	-	L'utilisateur pourra accéder au menu de son choix
New Game	Le joueur peut entrer son pseudo	Le joueur peut naviguer sur le menu à l'aide de clavier ou de la souris.	1. Être dans le Nickname 2. Utiliser les flèches vers le haut ou vers le bas ou le clavier pour sélectionner une lettre. 3. Enter pour confirmer	Nickname : AFTC666	Le nickname est confirmé et sauvegardé
New Game	Le joueur peut sélectionner ses caractéristiques de jeu.	Le joueur peut naviguer sur le menu à l'aide de clavier ou de la souris.	1. Être dans l'option Rocket, Gun, Map ou Difficulty. 2. Sélectionner l'une des cases disponibles 3. Enter pour sauvegarder la sélection		Les options sélectionnées sont enregistrées et affichées dans le panneau d'aperçu du jeu à droite.
New Game	Le joueur peut démarrer ou retourner au menu précédent.	Le joueur peut naviguer sur le menu à l'aide de	1. Sélectionner l'une des options ci-dessous entre quitter ou démarrer. 2. Appuyer sur Enter		Il retournera au menu ou démarrera le jeu avec les options sauvegardées.

		clavier ou de la souris.			
Multiplayer	Les joueurs peuvent entrer leur pseudo	Les joueurs peuvent naviguer sur le menu à l'aide de clavier ou de la souris.	1. Chaque joueur sera dans son nickname 1 ou 2 2. Utiliser les flèches vers le haut ou vers le bas ou le clavier pour sélectionner une lettre. 3. Entrer pour confirmer		Le nickname est confirmé, sauvegardé et affichées dans le panneau d'aperçu du jeu à droite
Multiplayer	Les joueurs peuvent sélectionner leurs propres caractéristiques de jeu.	Les joueurs peuvent naviguer sur le menu à l'aide de clavier ou de la souris.	1. Être dans l'option Rocket, Gun, Map ou Difficulty. 2. Sélectionner l'une des cases disponibles 3. Enter pour sauvegarder la sélection		Les options sélectionnées sont enregistrées et affichées dans le panneau d'aperçu du jeu à droite.
Multiplayer	Les joueurs peuvent revenir en arrière ou accéder au jeu.	Les joueurs peuvent naviguer sur le menu à l'aide de clavier ou de la souris.	1. Sélectionner l'une des options ci-dessous entre quitter ou démarrer. 2. Appuyer sur Enter		Il retournera au menu ou démarrera le jeu avec les options sauvegardées.
Highscore	Les 5 meilleurs scores sont affichés par ordre décroissant.	- Le highscore est affiché sous forme de tableau avec le rang, le surnom et le score.	1. Entrer à Highscore	Rank : 1 # Nickname : AFTC666 Score : 999	Les valeurs sont affichées dans le tableau et, sur le côté droit, un aperçu du navire utilisé dans le jeu.
Highscore	Revenir au menu d'accueil	Le joueur peut	1. Sélectionner l'option de retourner		Il retournera au menu.

		naviguer sur le menu à l'aide de clavier ou de la souris.	2. Appuyer sur Enter		
Option	Les couleurs du jeu peuvent être sélectionnées en fonction de l'interface, du joueur ou des ennemis.	Le joueur peut naviguer sur le menu options à l'aide de clavier ou de la souris.	1. Être dans l'option HUD, Player, Enemy Color 2. Sélectionner l'une des cases disponibles 3. Enter pour sauvegarder la sélection		Les changements de couleur seront affichés visuellement dans le panneau de prévisualisation du jeu sur le côté droit.
Option	La taille de la police peut être modifiée	Le joueur peut naviguer sur le menu options à l'aide de clavier ou de la souris.	1. Être dans l'option Letter Size 2. Sélectionner l'une des cases disponibles 3. Enter pour sauvegarder la sélection		Les changements de la police seront affichés visuellement dans le panneau de prévisualisation du jeu sur le côté droit.
Option	Le son peut être activé ou désactivé	Le joueur peut naviguer sur le menu options à l'aide de clavier ou de la souris.	1. Être dans l'option Sounds 2. Sélectionner l'une des cases disponibles 3. Enter pour sauvegarder la sélection		Les changements sera modifiée et cessera de sonner.
Option	La résolution de l'écran peut être modifiée.	Le joueur peut naviguer sur le menu	1. Être dans l'option Screen 2. Utiliser les flèches vers le haut ou vers le bas ou le clavier		Les changements de la résolution seront modifiée.

		options à l'aide de clavier ou de la souris.	pour sélectionner l'option. 3. Enter pour confirmer		
--	--	---	--	--	--

2. BD - Basse de données

2.1. Introduction DB « db_space_invaders »

« BD » est une base de données dans laquelle des informations sont stockées et peuvent être consultées ou modifiées. Ces données peuvent être représentées graphiquement grâce à la création d'un modèle logique de données (MLD), qui détaille le squelette de la base de données et sa répartition entre ses tables et ses valeurs. Ces bases de données sont utilisées en informatique pour enregistrer des informations et les gérer de manière plus appropriée à leur utilisation, par exemple, dans un jeu vidéo, elles pourraient être utilisées pour enregistrer les scores ou le niveau de progression de l'utilisateur.

Pour la création de Spicy Invaders, un schéma MLD représentant la base de données « db_space_invaders » a été présenté. À partir de cette base de données, toute sa structure sera utilisée pour simuler les utilisations possibles et dans la partie technique, la table t_joueur sera utilisée pour être implémentée dans le jeu réalisé.

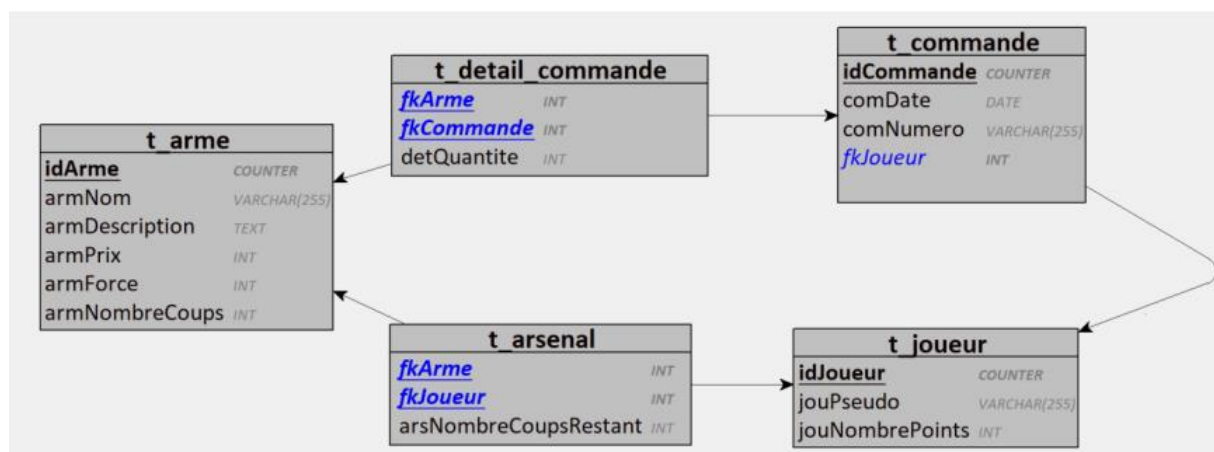


Figure 25 MLD

Ce MLD présente la situation d'un jeu vidéo dans lequel les joueurs sont enregistrés et leurs transitions d'achat et d'enregistrement d'armes. Sa fonction principale est d'identifier les joueurs par leur pseudonyme et de connaître les points gagnés afin de pouvoir acheter des armes de différents modèles, prix, cartouches de balles, forces et descriptions dans l'arsenal. En outre, la transition de l'achat enregistrera la date avec un numéro de commande et le détail de la quantité achetée.

2.2. Importer les données

Pour utiliser un serveur MySQL et gérer la base de données, nous utiliserons Docker. Docker est un programme qui déploie des conteneurs destinés à être gérés par un serveur. Dans ce cas, je vais utiliser Docker afin de créer un conteneur destiné à fonctionner comme un serveur MySQL.

Pour démarrer l'importation, nous aurons d'abord besoin de la base de données enregistrée dans un fichier. Mon chef de projet, Mathieu Meylan, m'a fourni le fichier "db_space_invaders.sql" qui correspond au schéma MLD présenté précédemment.

Une fois la base de données obtenue, il faut enregistrer le fichier dans le dossier du conteneur Docker et y accéder via la console du système d'exploitation "Invite de commandes" (CMD dans Windows). En utilisant la commande suivante dans la console, vous lancerez Docker pour importer la base de données dans votre serveur MySQL avec le nom d'utilisateur et son mot de passe :

```
docker exec -i (mysql-container-id) mysql -uroot -proot < db_space_invaders.sql
```

Pour valider cette importation, nous disposons de plusieurs méthodes. La première est de regarder dans le port de notre base de données locale comme MyPhpAdmin et de vérifier s'il est dans le schéma de la base de données.



Ou la deuxième option est à travers la console de conteneur dans Docker, il faut initialiser MySQL et mettre la commande suivante pour montrer toutes les bases de données enregistrées et faire accès à l'utilisation de la base de données « db_space_invaders ».

```
SHOW databases;
```

```
USE db_space_invaders;
```

```
+-----+
| Database
+-----+
| db_space_invaders
| information_schema
| mysql
| performance_schema
| sys
+-----+
```

2.3. Gestion des utilisateurs

Pour l'utilisation de MySQL, le serveur dispose de plusieurs utilisateurs par défaut qui utiliseront les bases de données en fonction de leurs privilèges et de leurs rôles. Pour le développement de ce projet, se commence par utiliser l'utilisateur "Root". Avec cet utilisateur, les permissions appropriées sont disponibles pour créer des utilisateurs, des rôles et assigner des permissions.

La situation nous présente 3 utilisateurs avec des profils et des missions différents.

- Administrateur du jeu
 - Peut créer, lire, mettre à jour et supprimer (CRUD) n'importe quelle table.
 - Gérer les utilisateurs et leurs privilèges.
- Joueur
 - Lire les informations des armes (pour voir quelles armes il peut acheter).
 - Créer une commande et lire toutes les commandes.
- Gestionnaire de la boutique
 - Lire les informations des armes (pour voir quelles armes il peut acheter).
 - Créer une commande.
 - Lire toutes les commandes.

Pour créer ces profils, il faut d'abord créer des rôles qui représentent le profil et ses missions, par exemple les lignes de commande suivantes :

```
CREATE ROLE 'r_admin';  
CREATE ROLE 'r_player';  
CREATE ROLE 'r_manager';
```

Ensuite, des utilisateurs fictifs seront ajoutés à la base de données avec son nom d'utilisateur, le host et le mot de passe sur IDENTIFIED. Exemple :

```
CREATE USER 'admin1'@'localhost' IDENTIFIED BY 'admin1';  
CREATE USER 'player1'@'localhost' IDENTIFIED BY 'player1';  
CREATE USER 'manager1'@'localhost' IDENTIFIED BY 'manager1';
```

Pour valider ce processus, il suffit d'utiliser la commande suivante qui affichera tous les utilisateurs et rôles créés :

```
SELECT user FROM mysql.user
```

```
+-----+  
| user |  
+-----+  
| db_user |  
| r_admin |  
| r_manager |  
| r_player |  
| root |  
| admin1 |  
| manager1 |  
| mysql.infoschema |  
| mysql.session |  
| mysql.sys |  
| player1 |  
| root |  
+-----+
```

L'étape suivante consiste à attribuer des privilèges à chaque rôle en fonction de sa mission. En fonction de chaque profil, les commandes suivantes doivent être saisies :

- Administrateur du jeu :

```
GRANT SELECT, UPDATE, INSERT, DELETE ON db_space_invaders.* TO r_admin;  
GRANT ALL PRIVILEGES ON db_space_invaders.* TO r_admin WITH GRANT OPTION;
```

- Joueur

```
GRANT SELECT ON db_space_invaders.t_arme * TO r_player;  
GRANT SELECT, INSERT ON db_space_invaders.t_arme * TO r_player;
```

- Gestionnaire de la boutique

```
GRANT SELECT ON db_space_invaders.t_joueur * TO r_manager;  
GRANT SELECT, UPDATE, DELETE ON db_space_invaders.t_arme * TO r_manager;  
GRANT SELECT db_space_invaders.t_commande * TO r_manager;
```

Pour finaliser l'attribution des rôles et des utilisateurs, procédez à la commande suivante pour confirmer leur attribution.

```
GRANT r_admin TO admin1@localhost;  
GRANT r_player TO player1@localhost;  
GRANT r_manager TO manager1@localhost;
```

Comme test de validation, les privilèges attribués aux utilisateurs et ceux attribués aux rôles peuvent être demandés au moyen d'une commande. Les images suivantes montrent ce résultat.

```
mysql> SHOW GRANTS FOR 'admin1'@'localhost';  
+-----+  
| Grants for admin1@localhost |  
+-----+  
| GRANT USAGE ON *.* TO `admin1`@`localhost` |  
| GRANT `r_admin`@`%` TO `admin1`@`localhost` |  
+-----+  
2 rows in set (0.00 sec)  
  
mysql> SHOW GRANTS FOR 'player1'@'localhost';  
+-----+  
| Grants for player1@localhost |  
+-----+  
| GRANT USAGE ON *.* TO `player1`@`localhost` |  
| GRANT `r_player`@`%` TO `player1`@`localhost` |  
+-----+  
2 rows in set (0.00 sec)  
  
mysql> SHOW GRANTS FOR 'manager1'@'localhost';  
+-----+  
| Grants for manager1@localhost |  
+-----+  
| GRANT USAGE ON *.* TO `manager1`@`localhost` |  
| GRANT `r_manager`@`%` TO `manager1`@`localhost` |  
+-----+  
2 rows in set (0.00 sec)
```



```
mysql> SHOW GRANTS FOR 'r_admin';
+-----+
| Grants for r_admin@% |
+-----+
| GRANT USAGE ON *.* TO `r_admin`@`%` |
| GRANT ALL PRIVILEGES ON `db_space_invaders`.* TO `r_admin`@`%` WITH GRANT OPTION |
+-----+
2 rows in set (0.00 sec)

mysql> SHOW GRANTS FOR 'r_player';
+-----+
| Grants for r_player@% |
+-----+
| GRANT USAGE ON *.* TO `r_player`@`%` |
| GRANT SELECT ON `db_space_invaders`.`t_arme` TO `r_player`@`%` |
| GRANT SELECT, INSERT ON `db_space_invaders`.`t_commande` TO `r_player`@`%` |
+-----+
3 rows in set (0.01 sec)

mysql> SHOW GRANTS FOR 'r_manager';
+-----+
| Grants for r_manager@% |
+-----+
| GRANT USAGE ON *.* TO `r_manager`@`%` |
| GRANT SELECT, INSERT, DELETE ON `db_space_invaders`.`t_arme` TO `r_manager`@`%` |
| GRANT SELECT ON `db_space_invaders`.`t_commande` TO `r_manager`@`%` |
| GRANT SELECT ON `db_space_invaders`.`t_joueur` TO `r_manager`@`%` |
+-----+
4 rows in set (0.00 sec)
```

2.4. Requêtes de sélection

Les requêtes de sélection sont des commandes qui facilitent la recherche d'informations dans des situations spécifiques. Dans ce qui suit, sont présentés 10 requêtes de sélection avec leur explication détaillée et la commande de recherche.

- Requête n°1

Dans cette première requête de la table t_joueur, les 5 premiers joueurs ayant obtenu le meilleur score sont affichés dans l'ordre du plus grand au plus petit score grâce à ORDER BY DESC et limitée par LIMIT. L'identité du joueur, son nom et son score sont affichés car s'utilise « * ».

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

idJoueur	jouPseudo	jouNombrePoints
5	SpaceRanger	6200
3	PlayerThree	6000
10	AstroAce	5750
9	StarStriker	5600
2	PlayerTwo	5500

- Requête n°2

La requête suivante recherche les prix les plus élevés avec MAX(), les prix les plus bas avec MIN() et la moyenne avec AVG() des armes. Pour effectuer ce processus, il faut accéder à la table t_arme et les données seront affichées dans la colonne correspondante avec leur nom déclaré en utilisant l'AS.

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

PrixMaximum	PrixMinimum	PrixMoyen
5000	100	1773.6842

- Requête n°3

En utilisant cette requête, il affichera à partir de la table t_commande le nombre de commandes effectuées par chaque utilisateur en indiquant leur identité avec GROUP BY et en les triant par ORDER BY selon le nombre de commandes, du plus au moins, avec DESC

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

IdJoueur	NombreCommandes
1	3
7	3
2	2
4	2
5	2
6	2
8	2
9	2
3	1
11	1

- Requête n°4

Le modèle suivant effectue la même recherche que le précédent, à la différence qu'il n'affiche que les utilisateurs dont les commandes sont supérieures à 2. Pour cette recherche, la condition HAVING sera appliquée pour ajouter la condition de recherche avec COUNT.

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

IdJoueur	NombreCommandes
1	3
7	3

- Requête n°5

Dans la requête suivante, chaque commande passée est affichée avec le nom de l'utilisateur et l'arme achetée. Pour cette recherche, le JOIN est utilisé pour réunir les tables t_commande, t_arme et t_joueur.

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

jouPseudo	armNom
PlayerOne	Laser basique
PlayerOne	Laser basique
PlayerOne	Laser basique
PlayerTwo	Missile nucl@aire
PlayerTwo	Missile nucl@aire
PlayerThree	Canon @ plasma
GalacticGamer	Bouclier magn@tique
GalacticGamer	Bouclier magn@tique
SpaceRanger	Bombes @ fragmentation
SpaceRanger	Bombes @ fragmentation
InvaderZapper	Rayon gelant
InvaderZapper	Rayon gelant
LunarLander	Torpilles cosmiques
LunarLander	Torpilles cosmiques
LunarLander	Torpilles cosmiques
CosmicCrusader	D@sint@grateur
CosmicCrusader	D@sint@grateur
StarStriker	Ondes soniques
StarStriker	Ondes soniques
NULL	Lance-roquettes quantique

- Requête n°6

Avec cette recherche, l'identité du joueur et le nombre de points dépensés dans la boutique seront affichés dans deux colonnes. Il est nécessaire d'accéder aux tables `t_commande`, `t_detail_commande` et `t_arme` pour recueillir les résultats. L'utilisation de `SUM()` calculera la somme de tous les prix des armes achetées et groupées avec `GROUP BY` pour montrer chaque utilisateur unique. Ces colonnes sont limitées à 10 résultats avec `LIMIT`.

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

idJoueur	TotalDepense
6	14000
4	8800
1	8100
7	7600
8	7600
2	7100
9	7000
5	3000
3	2200
11	2000

- Requête n°7

Dans la requête suivante, tous les joueurs seront affichés avec leur nombre de commandes exécutées ou non exécutées. Pour sa mise en œuvre, elle nécessite l'utilisation de `LEFT JOIN` pour utiliser l'ensemble de la table `t_joueur` et joindre ses valeurs à celles de la table `t_commande`. En plus, l'utilisation de la condition `WHERE` pour afficher tous les joueurs valides avec `IS NOT NULL`.

```
SELECT t_joueur.jouPseudo AS Joueur, t_commande.comNumeroCommande AS t_commande
FROM t_joueur LEFT JOIN t_commande ON t_commande.fkJoueur = t_joueur.idJoueur
WHERE t_joueur.jouPseudo IS NOT NULL ;
```

Joueur	t_commande
PlayerOne	CMD010
PlayerOne	CMD014
PlayerOne	CMD020
PlayerTwo	CMD011
PlayerTwo	CMD012
PlayerThree	CMD013
GalacticGamer	CMD001
GalacticGamer	CMD004
SpaceRanger	CMD005
SpaceRanger	CMD015
InvaderZapper	CMD006
InvaderZapper	CMD016
LunarLander	CMD003
LunarLander	CMD007
LunarLander	CMD017
CosmicCrusader	CMD008
CosmicCrusader	CMD018
StarStriker	CMD009
StarStriker	CMD019
AstroAce	NULL

- Requête n°8

Cette recherche est similaire à la précédente, à la différence qu'elle affiche toutes les commandes et les joueurs correspondants. Dans ce cas, si les commandes n'ont pas de joueur, celui-ci doit également être affiché comme NULL, c'est pourquoi le RIGHT JOIN est utilisé pour afficher tout le contenu de la colonne de droite.

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

Joueur	t_commande
GalacticGamer	CMD001
NULL	CMD002
LunarLander	CMD003
GalacticGamer	CMD004
SpaceRanger	CMD005
InvaderZapper	CMD006
LunarLander	CMD007
CosmicCrusader	CMD008
StarStriker	CMD009
PlayerOne	CMD010
PlayerTwo	CMD011
PlayerTwo	CMD012
PlayerThree	CMD013
PlayerOne	CMD014
SpaceRanger	CMD015
InvaderZapper	CMD016
LunarLander	CMD017
CosmicCrusader	CMD018
StarStriker	CMD019
PlayerOne	CMD020

- Requête n°9

La requête suivante affiche chaque joueur avec son nombre d'amras achetés. Pour effectuer cette recherche, il faut joindre les tables t_commande, t_detail_commande et un RIGHT JOIN pour afficher tous les joueurs de t_joueur. La fonction SUM() est également utilisée pour avoir le nombre d'armes achetées également groupées par GROUP BY en fonction de chaque joueur.

```
SELECT t_joueur.jouPseudo AS Joueur, SUM(t_detail_commande.detQuantiteCommande)
AS QuantiteArmes FROM t_commande JOIN t_detail_commande ON
t_detail_commande.fkCommande = t_commande.idCommande RIGHT JOIN t_joueur ON
t_commande.fkJoueur = t_joueur.diJoueur GROUP BY t_joueur.jouPseudo;
```

Joueur	QuantiteArmes
PlayerOne	7
PlayerTwo	7
PlayerThree	3
GalacticGamer	6
SpaceRanger	6
InvaderZapper	6
LunarLander	12
CosmicCrusader	7
StarStriker	7
AstroAce	NULL
NULL	5

- Requête n°10

Dans cette dernière requête, les joueurs qui ont acheté plus de 3 types d'armes différents seront affichés. Pour cette recherche, il faut utiliser et joindre avec JOIN les informations des tables t_commande, t_detail_commande et un RIGHT JOIN pour afficher tous les joueurs de t_joueur. En outre, l'option COUNT est utilisée avec un DISTINCT pour ne compter que le nombre de types d'armes différents. Enfin, le GROUP BY est utilisé sur les joueurs avec la condition HAVING pour limiter la recherche des types d'armes à plus de 3.

```
SELECT t_joueur.jouPseudo AS Joueur, COUNT (DISTINCT t_detail_commande.fkArme) as TypesArmes
FROM t_commande JOIN t_detail_commande ON t_detail_commande.fkCommande = t_commande.idCommande
RIGHT JOIN t_joueur ON t_commande.fkJoueur = t_joueur.idJoueur
GROUP BY t_joueur.jouPseudo HAVING COUNT(DISTINCT t_detail_commande.fkArme)>3;
```

Joueur	TypesArmes
GalacticGamer	4
LunarLander	4
PlayerTwo	4

2.5. Index

2.5.1 Index existant

Dans la base de données db_space_invaders.sql, aucun index n'a été créé pendant son utilisation. Par conséquent, pour vérifier que l'index est dans son contenu, la commande suivante doit être utilisée :

```
SHOW indexes FROM db_space_invaders.[table]
```

Ensuite, la console affichera tous les index pour chaque table. Ainsi, s'arrive à voir qu'il n'y a que les index par défaut créés par MySQL, les clés primaires et étrangères. Ces index sont utilisés pour la construction et la structuration de la base de données et de ses tables, précédemment utilisées pour lire les requêtes avec les jointures.

- Indexes de t_commande :

```
mysql> SHOW INDEXES FROM db_space_invaders.t_commande;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality
t_commande	0	PRIMARY	1	idCommande	A	20
t_commande	1	fkJoueur	1	fkJoueur	A	10

Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
NULL	NULL		BTREE			YES	NULL
NULL	NULL		BTREE			YES	NULL

- Indexes de t_joueur :

```
mysql> SHOW INDEXES FROM db_space_invaders.t_joueur;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality
t_joueur	0	PRIMARY	1	idJoueur	A	11

Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
NULL	NULL		BTREE			YES	NULL

- Indexes de t_arsenal :

```
mysql> SHOW INDEXES FROM db_space_invaders.t_arsenal;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality
t_arsenal	0	PRIMARY	1	fkArme	A	10
t_arsenal	0	PRIMARY	2	fkJoueur	A	10
t_arsenal	1	fkJoueur	1	fkJoueur	A	10

Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
NULL	NULL		BTREE			YES	NULL
NULL	NULL		BTREE			YES	NULL
NULL	NULL		BTREE			YES	NULL

2.5.2 Avantages vs Inconvénients

Avantages	Inconvénients
Optimiser le temps de recherche	Occuper une partie de la mémoire du système
Éviter les lectures séquentielles	Rétablit les raquettes d'insertion, de modification ou de suppression
Système d'arbre B avec arbre auto-équilibré et feuilles à la même profondeur	Il faut mettre à jour les index de chaque table.
Adapté à la gestion d'un grand nombre de tables	Inconvénient pour les petits tableaux

2.5.3 Indexes pertinentes.

Pour le jeu vidéo SpicyInvaders, une grille de recherche sera utilisée régulièrement pour afficher les 5 meilleurs joueurs avec leur nom et leur score. Ces informations seront affichées dans le menu Highscore. Pour cela, je pense qu'il est approprié de faire un index composite directement dans la table t_player pour leurs valeurs dans "jouPseudo" et "jouNamePoints". D'autre part, ils sont également appropriés pour vérifier le score de chaque joueur par rapport aux achats et aux commandes enregistrées. Voici la commande permettant d'effectuer cet index :

```
CREATE INDEX indx_joueurs ON t_joueur(jouPseudo, jouNombrePoints)
```

La base de données sera ainsi rationalisée en commençant la recherche par les valeurs fondamentales du joueur.

2.6. Backup/Restore

La Backup est une sauvegarde de documents, de fichiers ou, dans le cas présent, d'une base de données. Cette sauvegarde enregistrera la base de données sélectionnée pour être extraite dans un fichier différent qui contiendra toutes les informations de la base de données d'origine.

L'utilisation de la sauvegarde est très utile pour préserver et prendre soin des informations personnelles ou d'une entreprise. Son utilisation implique de se prémunir contre des erreurs futures telles qu'une mauvaise modification du fichier, des données corrompues ou une suppression involontaire de toutes les informations.

Il existe différents types de sauvegarde, mais dans cette pratique, s'utilisera la méthode simple qui consiste à copier et à extraire les informations vers un fichier externe. Et en même temps, pour pouvoir utiliser une Restore qui impliquerait la restauration de la base de données extraite pour être à nouveau implémentée dans notre bibliothèque de bases de données.

Sur les bases de données MySQL, il existe différentes commandes qui permettent d'effectuer à la fois la sauvegarde et la restauration. Dans ce cas, il est nécessaire de faire une copie de sauvegarde de la base de données que nous avons utilisée db_space_invaders et de pouvoir la restaurer pour pouvoir l'utiliser à nouveau.

La première commande pour effectuer la sauvegarde est la suivante :

```
Mysqldump -u root -proot -databases db_space_invaders > backup_db_space_invaders.sql
```

Avec l'utilisation du service Mysqldump, il nous offre la possibilité d'extraire la base de données et de la sauvegarder dans un fichier externe appelé backup_db_space_invaders.sql.

Une fois la sauvegarde obtenue, au cas où nous voudrions l'utiliser à nouveau ou la récupérer, il suffit de déployer la base de données dans notre bibliothèque MySQL et de la renommer sous son nom original "db_space_invaders".

```
Mysql -u root -proot db_space_invaders < backup.db_space_invaders.sql
```

2.7. MySQL – db_sapce_invaders danse le Highscore

Dans le programme Spacy_Invaders qui accompagne ce rapport, une section importante permet de sauvegarder les données des parties jouées. Pour chaque partie jouée, le joueur est invité à insérer un pseudo, qui sera utilisé ultérieurement pour être affiché avec le score obtenu dans la partie. De cette façon, en enregistrant à la fois le surnom et le score, un menu Highscore peut être créé où les 5 meilleurs scores seront affichés.

Pour la réalisation de cette section, il est nécessaire de connecter la base de données db_space_invaders avec le code de notre projet. Voici une explication détaillée de ce processus et comment l'utiliser pour notre intérêt.

Tout d'abord, nous devons configurer la distribution de notre projet avec un paquetage MySql.Data (qui peut être installé directement dans Microsoft Visual Studio) et l'activer avec un "using MySql.Data.MySqlClient" dans le code.

Ensuite, nous pouvons procéder à la création d'une classe DatabaseConnection. Dans cette classe, il sera important de créer l'instance avec les informations de notre serveur de base de données, par exemple :

```
private string connectionS = "Server=localhost;Port=6033;" +  
"Database=db_space_invaders;UserId = root; Password=root;"
```

Une fois que nous avons détaillé notre serveur, le port et la base de données, notre utilisateur et notre mot de passe, nous procédons à la création de deux méthodes principales "SavePoints" et "Highscore". La première méthode servira à sauvegarder les informations de chaque jeu, il faudra donc demander le nom et les points du jeu pour cette méthode. La seconde méthode sera utilisée pour afficher la liste des 5 meilleurs jeux.

Dans chaque méthode, il est nécessaire de se connecter à la base de données, il faudra donc créer un objet de la classe MySqlConnection auquel nous pourrons accéder en ajoutant un ".Open()" à l'objet créé.

Une fois la base de données initialisée, nous procédons à la création des requêtes. Pour enregistrer le score il faudra appliquer un INSERT INTO, dans mon cas il s'agit de "INSERT INTO t_joueur (jouPseudo, jouNamePoints) VALUE (@nickname, @points)". Cette requête va procéder à l'insertion des valeurs du pseudo du joueur et de son score en points. Et pour la deuxième méthode, l'affichage des scores se fait avec une commande SELECT qui dans mon cas est "SELECT jouPseudo, jouNicknamePoints FROM t_joueur ORDER BY jouNicknamePoints DESC LIMIT 5". Avec cette commande, seuls les 5 meilleurs scores seront affichés et dans l'ordre décroissant.

Après avoir déclaré la requête, il faut créer un objet de la classe MySqlCommand(), qui permet l'exécution de la requête dans la base de données. Dans la méthode d'enregistrement des scores, la commande doit être exécutée avec un ".ExecuteNonQuery()". Et dans la méthode pour le Highscore, elle doit être utilisée comme ".ExecuteReader()" uniquement pour lire et noter les informations de la table.

Une fois la construction des méthodes terminée, il ne reste plus qu'à les appliquer dans le code principal en appelant la classe principale DatabaseConnection et en appliquant ses méthodes.

Ci-dessous un exemple de code :

```
public void SavePoints(string nickName, int points)
{
    using (MySQLConnection connect = new MySqlConnection(connectionS))
    {
        try
        {
            connect.Open();
            string requete = "INSERT INTO t_joueur (jouPseudo, jouNombrePoints) VALUE (@nickName,@points)";
            MySqlCommand cmd = new MySqlCommand(requete, connect);
            cmd.Parameters.AddWithValue("@nickName", nickName);
            cmd.Parameters.AddWithValue("@points", points);
            cmd.ExecuteNonQuery();
        }
        catch (MySqlException ex)
        {
            Console.WriteLine("Erreur : " + ex.Message);
        }
    }
}
```

Figure 26 DB – SavePoints Méthode

```
public void Highscore()
{
    using (MySQLConnection connect = new MySqlConnection(connectionS))
    {
        try
        {
            connect.Open();
            string requete = "SELECT jouPseudo, jouNombrePoints FROM t_joueur ORDER BY jouNombrePoints DESC LIMIT 5";
            MySqlCommand cmd = new MySqlCommand(requete, connect);
            using (MySqlDataReader reader = cmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    playersName.Add(reader.GetString(0));
                    playersScore.Add(reader.GetInt64(1));
                }
            }
        }
        catch (MySqlException ex)
        {
            Console.WriteLine("Erreur : " + ex.Message);
        }
    }
}
```

Figure 27 DB - Highscore Méthode

3. POO - Programmation orientée objet

3.1. Introduction

Avec ce projet Spicy Invaders, un petit jeu vidéo sera réalisé dans Microsoft Visual Studio en code C#.

Le but est de réaliser un projet similaire au célèbre jeu vidéo Space Invaders, mais dans ce cas appelé Spicy Invaders. Il doit suivre les objectifs de base tels que le joueur qui tire et se déplace, 10 ennemis qui descendent sur l'axe Y et des projectiles. Outre les éléments jouables, il y a également un système de victoire, de comptage des points et des menus permettant d'arrêter chaque section correspondante.

Pour la présentation du projet, plusieurs menus principaux doivent être présentés, comme le menu principal où l'on peut accéder pour jouer, le highscore où il enregistrera nos parties et montrera les 5 meilleurs scores, le menu des contrôles du jeu et enfin pour quitter le programme.

Le jeu suivra le fonctionnement de base du moteur de jeu : initialisation, mise à jour dans Update, finalisation. Suivant cette idée, le processus de ce projet et son explication seront présentés ci-dessous.

3.2. Analyse fonctionnelle

Déplacer "Player"

En tant que utilisateur Je veux me déplacer de gauche à droite Afin contrôler le Player		
Tests d'acceptance:		
Déplacement droit	En cours de jeu On appuie sur "d" ou flèche droit	Le joueur se déplacer à droit
Déplacement gauche	En cours de jeu On appuie sur "a" ou flèche gauche	Le joueur se déplacer à gauche
Le mouvement est limité	Lorsque que je suis au bord du jeu	Quand je me déplace à droite ou a gauche encore vers le bord Le joueur ne fait rien

Importer les données et le schéma de base de données

En tant que utilisateur Je veux utiliser une base de données afin de pouvoir utiliser et gérer les informations		
Tests d'acceptance:		
Docker fonctionne correctement	Pour initialiser utiliser la base de donne, lorsque je dois importer la base de données sur mon pc je dois avoir le docker configuré et prêt avec 'db_space_invaders.sql'	

C'est possible d'accéder à MySQL pour l'utiliser	Lorsque je suis dans la console	Quand j'utilise les données de la base de données	Je dois pouvoir y accéder avec MySQL
La base de données peut être utilisée et affichée dans la console	Lorsque que je suis dans MySQL	Quand j'utilise la base de données	Je peux utiliser et afficher les informations

Mouvement Ennemie




En tant que utilisateur Je veux que les enemies aient une trajectoire afin d'avoir enemies dynamiques			
Tests d'acceptance:			
Déplacement droit	Au début du jeu	Automatiquement	L'ennemie se déplace à droite
Le mouvement est limité	En cours de jeu	Quand l'ennemie arrive à la limite du jeu	l'ennemi descend d'une ligne et change son mouvement dans l'autre direction

Mouvement de projectile

En tant que utilisateur Je veux que les balles aient une trajectoire afin de permettre au joueur d'attaquer			
Tests d'acceptance:			
Origine du projectile	En cours de jeu	Quand je presse la barre d'espace	un projectile apparaît à la pointe du joueur
Trajectoire linéaire	En cours de jeu avec un projectile lancé automatiquement		il monte en ligne droite
Élimination des projectile	En cours de jeu	Quand le projectile dépasse les limites de la zone	il est éliminé

Collisions

En tant que utilisateur Je veux que les enemies aient une collision afin d'avoir être touché par une projectile			
Tests d'acceptance:			
Le projectile est détruite	En cours de jeu	Quand le projectile touche un ennemi	le projectile est détruit

Joueur gagne des points	En cours de jeu Quand le projectile touche un ennemi le joueur gagne des points
L'ennemi change de forme	En cours de jeu Quand le projectile touche un ennemi le projectile modifie la forme de l'ennemi (regarder "enemiesFases")
<div style="display: flex; justify-content: space-around; align-items: center;"> <div>PV = 3</div>  </div> <div style="display: flex; justify-content: space-around; align-items: center;"> <div>PV = 2</div>  </div> <div style="display: flex; justify-content: space-around; align-items: center;"> <div>PV = 1</div>  </div>	

Système de défaite

En tant qu'utilisateur	Je veux un système de défaite	Quand les ennemis me battent
Tests d'acceptance:		
La partie est terminée lorsque le joueur est mort		Dans le jeu lorsque l'ennemi touche le joueur le jeu est terminé
La défaite sera présentée avec une information et le score		Dans le jeu lorsque le jeu est terminé un écran avec le score et "Game Over" s'affiche .
En cas de défaite, vous pouvez rejouer ou retourner au menu de départ		Dans le jeu lorsque le jeu est terminé il y a une option pour rejouer ou revenir au menu principal.

Système de points

En tant qu'utilisateur	Je veux un système de points	Quand je défais a un enemie
Tests d'acceptance:		
La somme des points	En jeu lorsqu'un ennemi est touché par une balle	Des points sont marqués
Score s'affiche	En jeu Pendant le déroulement du match un compteur permet de suivre le score.	

Sauvegarde des scores	Dans le jeu lorsque le jeu est terminé le résultat est sauvgarder por montre dans le Game Over ou Highscore
-----------------------	---

Menus et interface du programme

En tant qu'utilisateur Je veux que le jeu ait un menu qui permette de jouer, de voir le meilleur score, les commandes et de quitter le jeu. afin d'être accessible de pouvoir interagir avec l'interface du jeu.

Tests d'acceptance:

Menu d'accueil	Dans le Jeu Lors du premier accès le jeu affiche son titre et un menu avec des options de jeu, de score, de contrôle et de sortie.
Menu de highscore	Dans le Jeu Lorsqu'on accède à higscore le jeu affiche un tableau avec les résultats des 5 meilleurs joueurs.
Menu Contrôles	Dans le Jeu Lorsqu'on accède à Controles le jeu affiche des informations sur les contrôles utilisés pour jouer.
Option de sortie	Dans le Jeu Lorsqu'on est sur le menu d'accueil il y a une option pour quitter le jeu
HUD	Dans le Jeu Lorsqu'on joue les informations et le score du joueur sont présentés.

3.3. Analyse technique

3.3.1. Diagramme de basse

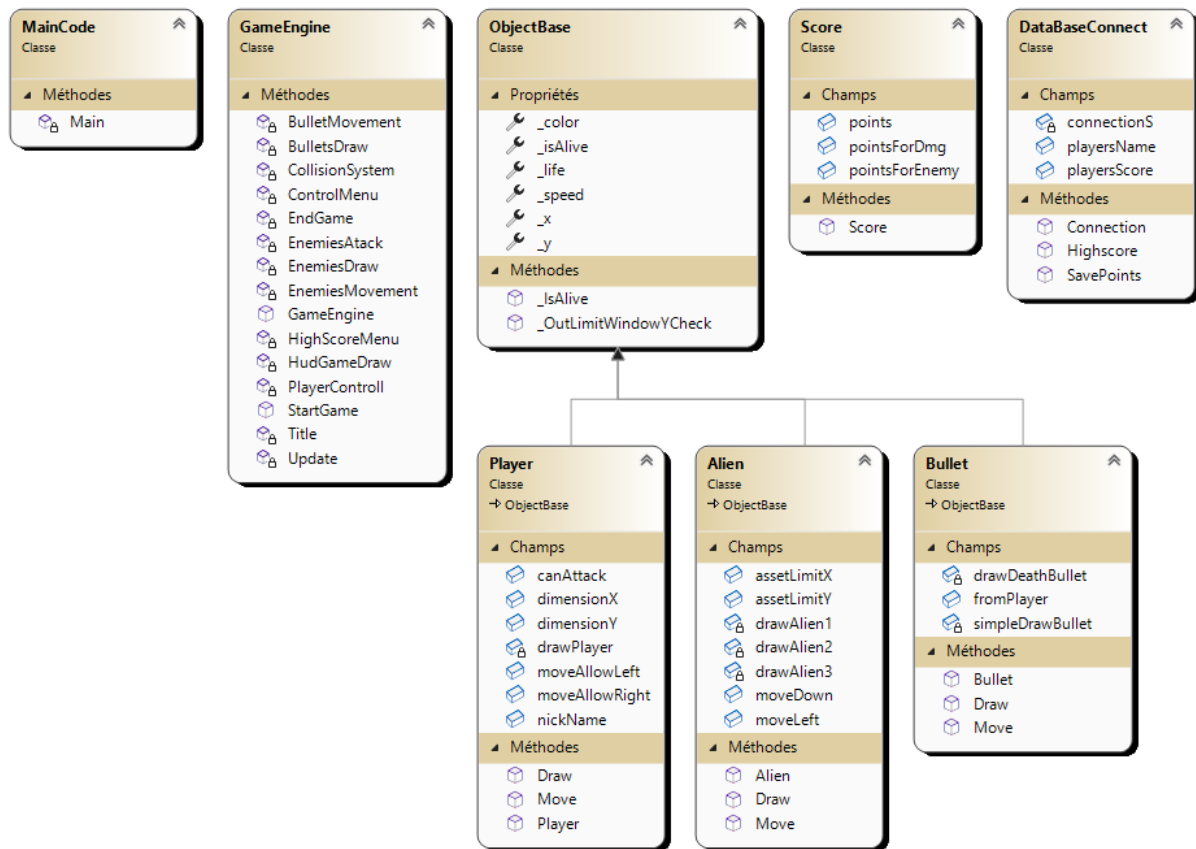


Figure 28 Diagramme de classe SpicyInvaders [en page 51](#)

3.3.2. Explication

Un document docfx a pour fonction de produire une documentation sur le code créé et de donner une explication détaillée de son fonctionnement avec chaque méthode et variable utilisée. Ce document est très utile pour pouvoir expliquer de manière simple notre programme et avoir accès à ses méthodes.

Le document docfx pour notre jeu vidéo se trouve dans le fichier Doc/Docfx/SpicyInvadersDockfx.pdf.

3.4. Test

3.4.1. Unit Testing

Le système UnitTesting est souvent utilisé en programmation pour vérifier si les méthodes et les fonctions d'un nouveau système fonctionnent correctement et transmettent le résultat souhaité.

Pour notre jeu vidéo SpicyInvaders, nous utiliserons le UnitTesting pour deux méthodes essentielles du jeu.

Les tests créés pour ce projet travailleront sur les objets de base créés dans le jeu, ObjectBase. Cette classe est le parent du système d'héritage présenté dans le code

et comporte deux méthodes qui seront utilisées ultérieurement dans ses classes enfants. Ces méthodes sont `_isAliveTest()` et `_OutLimitWindowYCheckTest()`.

Le premier test comporte une fonction permettant de déterminer si un objet est vivant ou mort. Pour cette fonction, vous devez connaître la variable `life` pour savoir combien de vies a l'objet. Si la vie est égale à 0, cette fonction renverra `False`, et si l'objet a des vies, elle renverra `true`. Dans l'exemple que nous avons utilisé pour faire le test, nous avons créé deux objets de la classe `Player` qui hérite des caractéristiques de son père. Pour le premier objet, la valeur de sa vie a été fixée à 0 et pour le second à 3. En lançant le test, nous pouvons voir que dans le premier cas, le résultat est négatif et dans le second positif, ce qui prouve que l'UnitTesting est valide.

Le deuxième test valide la fonction qui vérifie que l'objet se trouve dans les limites de la console. Pour présenter ce test, j'ai créé 3 classes de type bullets avec des positions différentes. En supposant que la limite de la console soit de 100 de hauteur, je recevrai deux valeurs qui me donneront une valeur positive pour être en dehors de la console, et une valeur négative qui sera dans l'intervalle de la hauteur de la console. En lançant l'explorateur de test, nous pouvons voir la validation correcte.

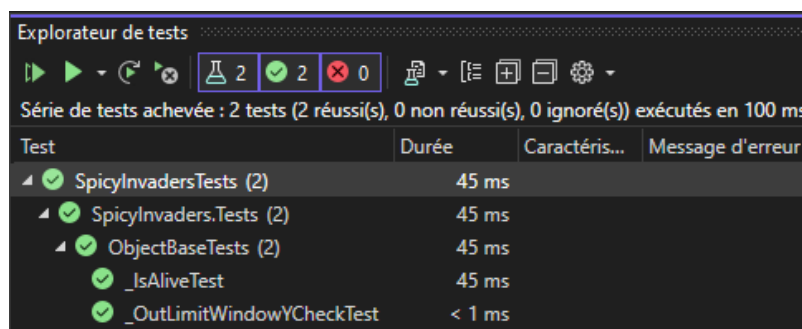


Figure 29 POO - UnitTesting

3.4.2. Rapport de test

Système de points

La somme des points	En jeu lorsqu'un ennemi est touché par une balle Des points sont marqués
Socre s'affiche	En jeu Pendant le déroulement du match un compteur permet de suivre le score.
Sauvegarde des scores	Dans le jeu lorsque le jeu est terminé le résultat est sauvgarder por montre dans le Game Over ou Highscore

Système de défaite

La partie est terminée lorsque le joueur est mort	Dans le jeu lorsque l'ennemi touche le joueur le jeu est terminé
La défaite sera présentée avec une information et le score	Dans le jeu lorsque le jeu est terminé un écran avec le score et "Game Over" s'affiche .

En cas de défaite, vous pouvez rejouer ou retourner au menu de départ	Dans le jeu lorsque le jeu est terminé il y a une option pour rejouer ou revenir au menu principal.
---	---

Collisions

Le projectile est détruite	En cours de jeu Quand le projectile touche un ennemi le projectile est détruit
Joueur gagne des points	En cours de jeu Quand le projectile touche un ennemi le joueur gagne des points
L'ennemi change de forme	En cours de jeu Quand le projectile touche un ennemi le projectile modifie la forme de l'ennemi (regarder "enemiesFases")
Chaque page comporte au moins un élément de pratiques économiques	Sur chaque page de High Fidelity dans son interface doit suivre une structure avec des pratiques économiques.

Importer les données et le schéma de base de données

Docker fonctionne correctement	Pour initialiser utiliser la base de donne, lorsque je dois importer la base de données sur mon pc je dois avoir le docker configuré et prêt avec 'db_space_invaders.sql'
C'est possible d'accéder à MySQL pour l'utiliser	Lorsque je suis dans la console Quand j'utilise les données de la base de données Je dois pouvoir y accéder avec MySQL
La base de données peut être utilisée et affichée dans la console	Lorsque que je suis dans MySQL Quand j'utilise la base de données Je peux utiliser et afficher les informations

Mouvement de projectile

Origine du projectile	En cours de jeu Quand je presse la barre d'espace un projectile apparaît à la pointe du joueur
Trajectoire linéaire	En cours de jeu avec un projectile lancé automatiquement il monte en ligne droite
Élimination des projectile	En cours de jeu Quand le projectile dépasse les limites de la zone il est éliminé

Mouvement Ennemie

Déplacement droit	Au début du jeu Automatiquement L'ennemie se déplace à droite
Le mouvement est limité	En cours de jeu Quand l'ennemie arrive à la limite du jeu l'ennemi descend d'une ligne et change son mouvement dans l'autre direction

Déplacer "Player"

Déplacement droit	En cours de jeu On appuie sur "d" ou fleche droit Le joueur se déplacer à droit
Déplacement gauche	En cours de jeu On appuie sur "a" ou fleche gauche Le joueur se déplacer à gauche
Le mouvement est limité	Lorsque que je suis au bord du jeu Quand je me déplace à droite ou a gauche encore vers le bord Le joueur ne fait rien

Menus et interface du programme

Menu d'accueil	Dans le Jeu Lors du premier accès le jeu affiche son titre et un menu avec des options de jeu, de score, de contrôle et de sortie.	???
Menu de highscore	Dans le Jeu Lorsqu'on accède à higscore le jeu affiche un tableau avec les résultats des 5 meilleurs joueurs.	???
Menu Contrôles	Dans le Jeu Lorsqu'on accède à Controles le jeu affiche des informations sur les contrôles utilisés pour jouer.	???
Option de sortie	Dans le Jeu Lorsqu'on est sur le menu d'accueil il y a une option pour quitter le jeu	???
HUD	Dans le Jeu Lorsqu'on joue les informations et le score du joueur sont présentés.	???

4. Conclusions

SpicyInvaders est un jeu vidéo qui a été travaillé selon 3 éléments importants de la programmation et du développement d'applications : la base de données, l'expérience utilisateur et la programmation orientée objet.

Au cours du développement de ce projet, son objectif principal a été de pouvoir intégrer ces éléments pour un développement correct d'un programme avec l'utilisation d'outils connus dans les expériences de travail et les entreprises.

Spicy Invaders est le résultat d'un travail qui a été contrôlé par nos professeurs de classe, en suivant la théorie de la matière, en appliquant ces connaissances en tant que matière et en développant le projet en suivant un processus correct via Github et IceScrum.

4.1. UX

En ce qui concerne le processus UX, on peut voir que grâce à la connaissance des théories et des méthodes correctes, nous pouvons développer une interface et un programme avec nos clients ou utilisateurs comme objectif principal. Grâce au travail UX, j'ai pu montrer et choisir une bonne construction d'éléments visuels et de fonctions adaptées aux utilisateurs potentiels.

4.2. DB

Connaître le fonctionnement de la base de données m'a aidé à progresser sur un point important de l'informatique. Au cours de ce projet, le fait de pouvoir toucher à tous les aspects liés à la base de données et à sa construction, tels que savoir comment créer des utilisateurs et des autorisations, des commandes requeta et être capable d'implémenter tout cela dans notre programme en code, a abouti à un jeu solide qui se développe au fur et à mesure que vous jouez et enregistrez vos données.


4.3. POO

La programación es el punto mas importante para la elaboración del juego. La creación del juego ha partido de hacer un juego con conocimientos básicos y métodos simples de programación a poder formar un código solido con elementos esenciales de programación orientada en objetos como la herencia. Gracias a estos nuevos conocimientos he podido desarrollar un juego que cumple su funcionamiento de principio a fin y con un código más profesional.

5. Sources

- Chat GPT n'a pas été utilisé pour la réalisation de ce projet
- <https://colorbrewer2.org/#type=diverging&scheme=Spectral&n=3>
- PNGEGG – <https://pngegg.com/>
- PNGWing - <https://www.pngwing.com/>
- Vecteezy - <https://es.vecteezy.com/>
- Github - <https://github.com/>
- IceScrum - <https://etml.icescrum.com/p/SPICY14>
- Figma - https://www.figma.com/file/KqfYpYpTRTZpqo60St5uR/Spicy-Invaders_UX?type=design&node-id=11-4&mode=design&t=0OXRiRDdAyvr3NKa-0
- Youtube - <https://www.youtube.com/>
- Microsoft Learn - <https://learn.microsoft.com/es-es/dotnet/csharp/methods>
- Deepl - <https://www.deepl.com/translator>
- Coolors - <https://coolors.co/>
- FontSpace - <https://www.fontspace.com/>

6. Annex



"Je collectionne des jeux vidéo rétro pour les conserver "

ÂGE 31

POSTE Vendeur

STATUS Célibataire

UBICATION London

PASSIONNÉ **EMPATHETIQUE**

CURIOUS **AVENTURE**

Mar Brios

SUJET

Mar est un vendeur de jeux vidéo qui travaille chez GameStop à Londres. Elle est une grande fan de jeux vidéo rétro. Souffre de fatigue visuelle et ne peut donc pas jouer pendant de longues périodes. Son père lui a offert une enfance remplie de jeux vidéo. Elle est donc très attachée à la préservation des jeux et aime voir comment les jeux ont évolué jusqu'à aujourd'hui. Pendant son temps libre, elle aime regarder des films et des séries des années 80, suivre des compétitions d'e-sport et des speedruns. Comme sport, il aime se promener avec son chat. Son plat préféré est le tacos français et il ne supporte pas les plats épicés.

OBJECTIFS

- Jeux faciles à jouer mais difficiles à maîtriser
- Jeux où vous pouvez personnaliser votre vision pour l'adapter à la fatigue oculaire
- Interface et illustrations de jeux simples

FRUSTRATIONS

- Jeux sans possibilité de rejouer
- Jeux sans compétition
- Jeux de longue durée

PERSONNALITÉ

Introverti	○	Extraverti
Analytique	○	Créatif
Occupé	○	Libre
Désordonné	○	Organisé
Indépendants	○	Esprit d'équipe

Figure 30 Mar Brios Meilleure qualité



Figure 31 Zel Workukun



Figure 32 HF - Menu d'accueil



Figure 33 HF - Menu new game



Figure 34 HF - Menu multiplayer



Figure 35 HF - Menu Highscore



Figure 36 HF - Menu Game Options

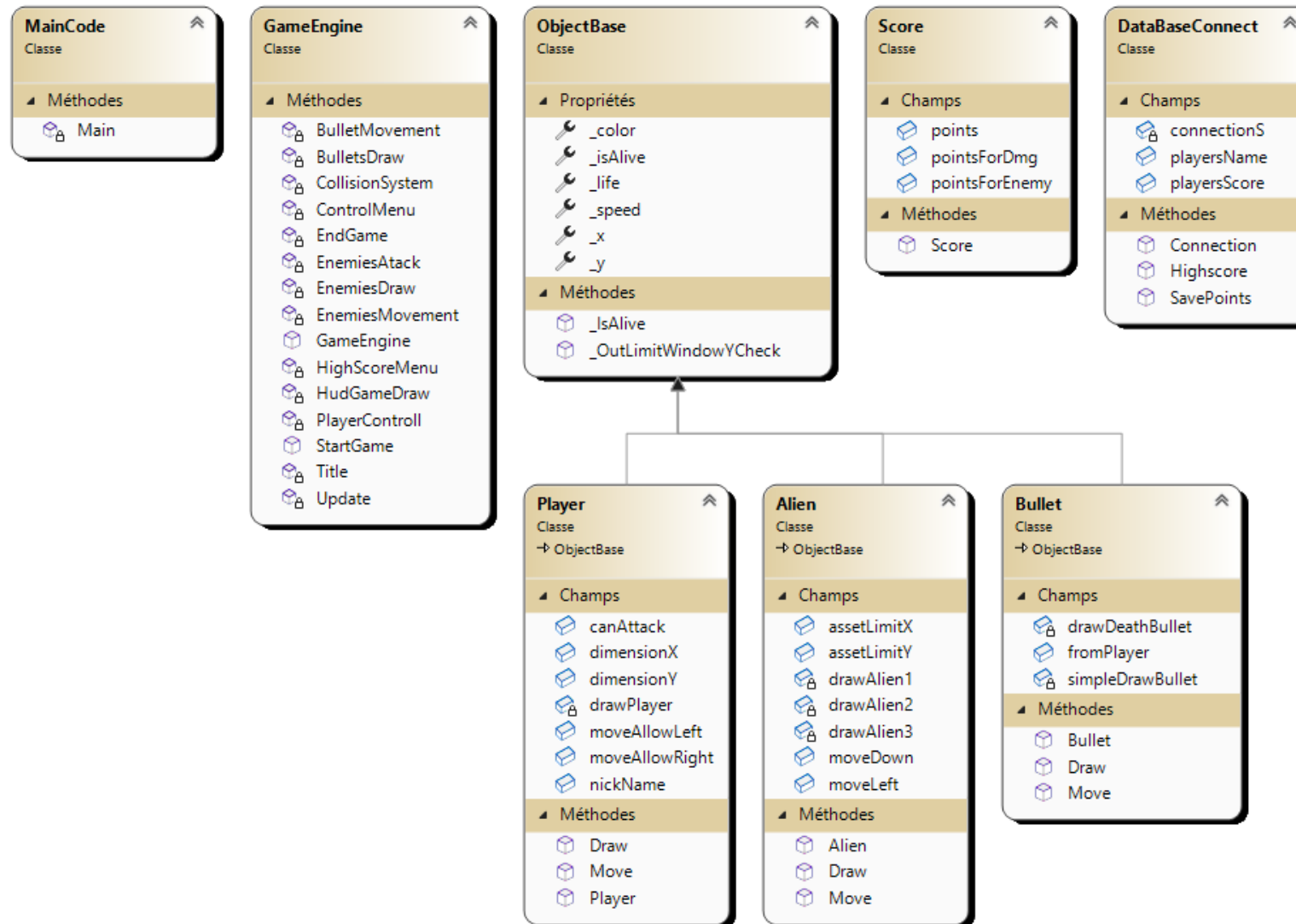


Figure 37 POO - Diagramme de classe