



ECOLE TECHNIQUE
ECOLE DES MÉTIERS LAUSANNE

« SPICY INVADER »

Rapport projet : P_POO, P_UX & P_DB

Faire un réplica du célèbre jeu Space Invader



28 AOÛT 2023

ADRIAN TOLEDO

Enseignant : Xavier Carriel, Aurélie Curchod & Mathieu Meylan

Table des matières

1. UX	2
1.1. Introduction	2
1.2. Analyse	2
1.3. Conception	7
2. DB	19
2.1. Introduction DB « db_space_invaders »	19
2.2. Importer les données	20
2.3. Gestion des utilisateurs	21
2.4. Requêtes de sélection.....	23
2.5. Index	28
2.6. Backup/Restore	30
3. POO.....	32
3.1. Introduction	32
3.2. Analyse fonctionnelle	32
3.3. Analyse technique.....	32
3.3.1. Diagramme de base.....	32
3.3.2. Explication (docfx)	32
3.4. Test Unitaire.....	32
4. Conclusions	32
5. Sources.....	32

1. UX

1.1. Introduction

UX signifie l'expérience de l'utilisateur. En quelques mots, 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 donc l'utilisation d'un produit destiné à un utilisateur et tente de 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, l'un des éléments les plus importants pour la création d'un produit ou d'une interface est l'utilisateur qui l'utilisera. Plus précisément, le fait de détailler le profil de l'utilisateur prévu 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 objectifs 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.



"Je collectionne des jeux vidéo rétro pour les conserver"

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.

ÂGE	31
POSTE	Vendeur
STATUS	Célibataire
UBICATION	London

PASSIONNÉ

EMPATHETIQUE

CURIOUS

AVENTURE

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

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 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 celui de 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.



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

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

ÂGE 64

POSTE Retraité

STATUS Veuf

UBICATION Tokyo, Japon

ANALYTIQUE

ORGANISE

EMPATHETIQUE

CRITIQUE

PERSONNALITÉ

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

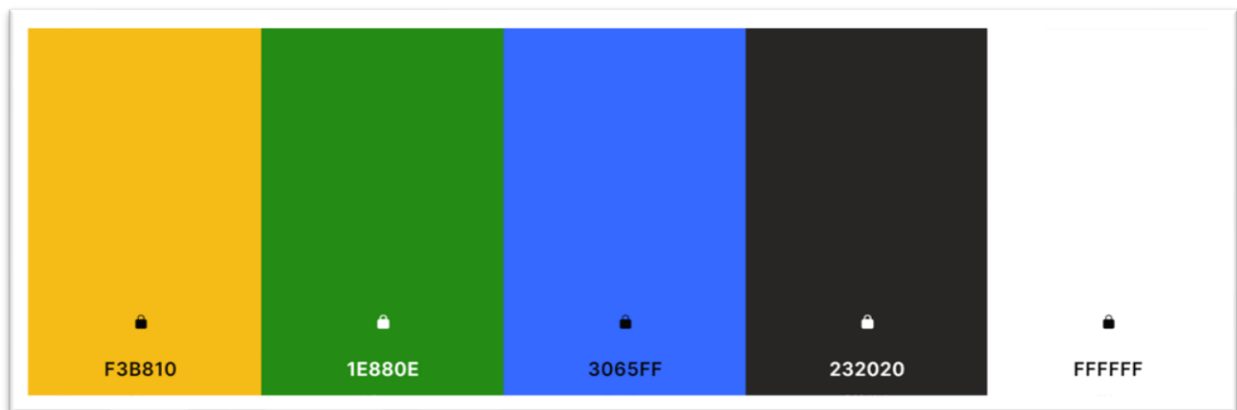
Cet exemple de Zel Workukun nous montre l'importance de la diversité des joueurs possibles, d'âges et de sexes différents. Workukun implique de penser aux personnes âgées, dont la seule préoccupation est de pouvoir profiter des jeux vidéo seul ou avec des amis. En outre, un point important est l'accessibilité 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 importante, 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.

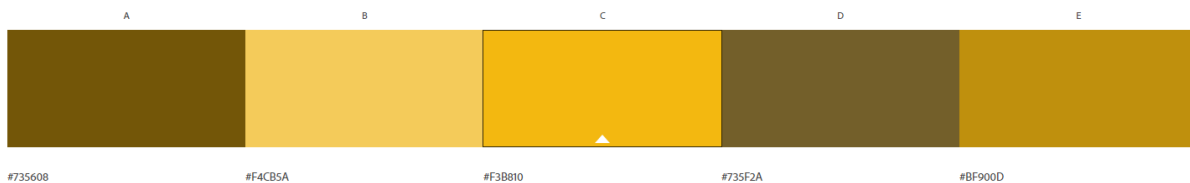
Palette 1 :



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:



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

L'accessibilité dans l'informatique et les jeux vidéo s'est améliorée et a mis en œuvre de plus en plus de ressources dans leurs services. Il 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:

- Actifs 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

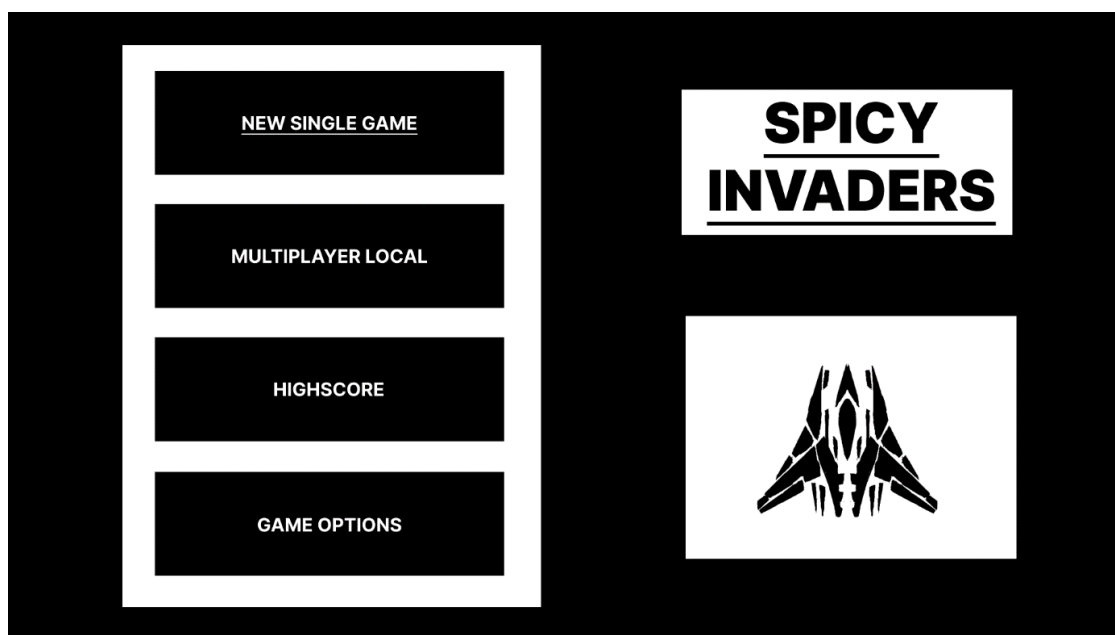
Dans ce chapitre, nous présentons la maquette conçue pour la création de l'interface du jeu vidéo Spicy Invaders. L'élaboration de la conception ainsi que les choix des éléments visuels seront détaillées. Les maquettes sont divisées en deux parties, la première montre des maquettes Low Fidelity où les 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. Par conséquent, 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

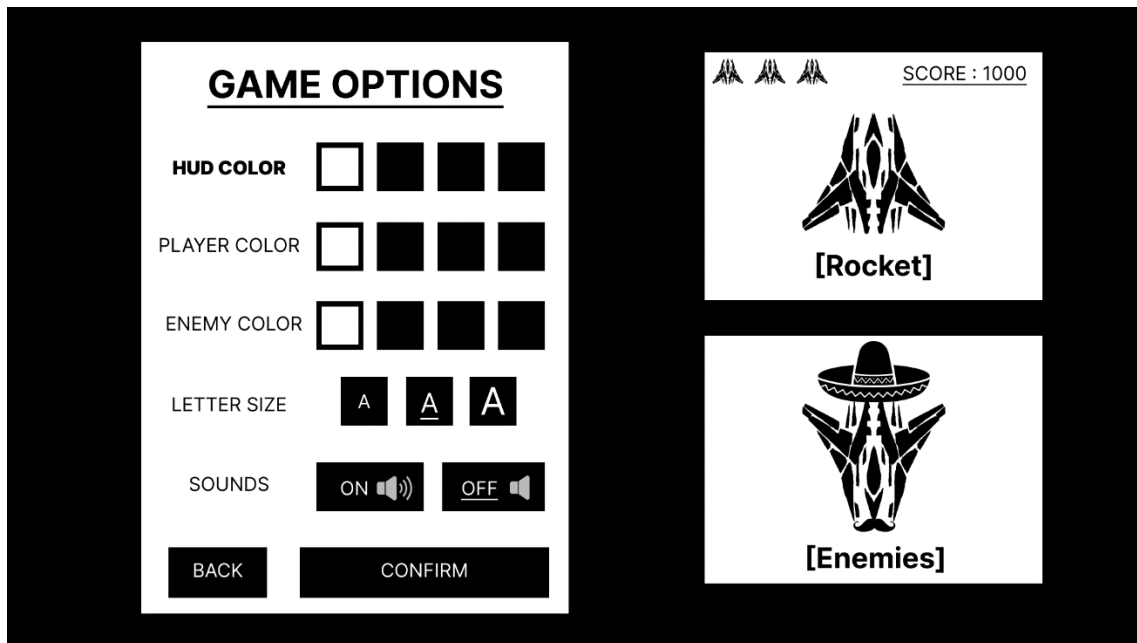
- Menu

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.



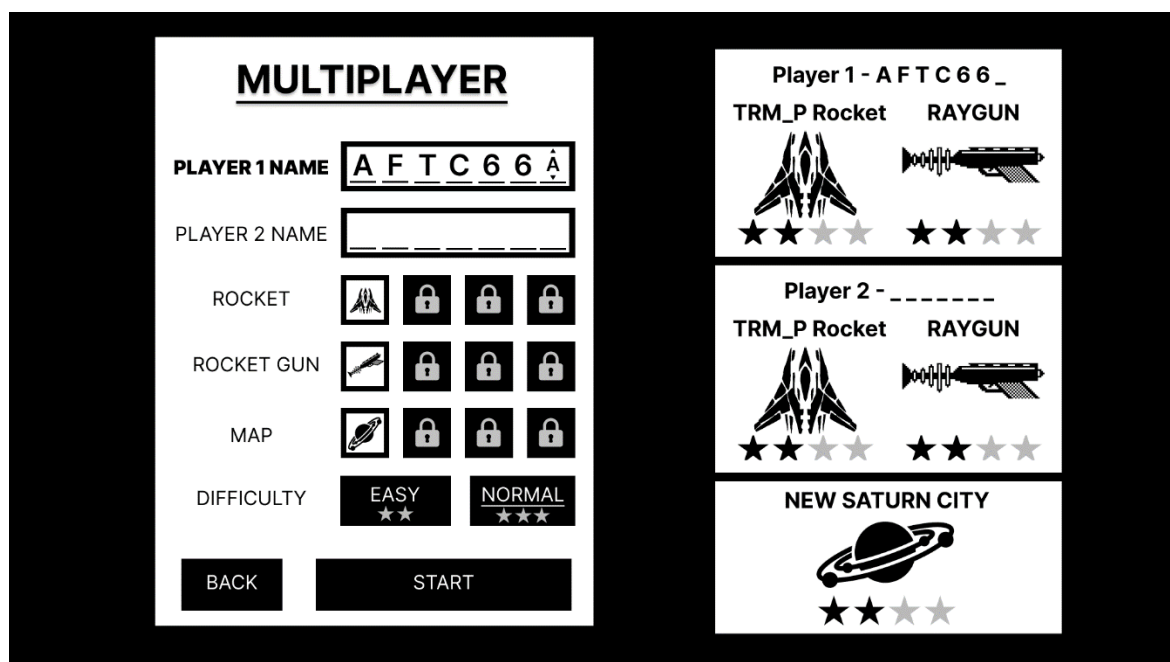
- New Game

Lorsque vous jouez seul, la salle de jeu est créée et c'est possible sélectionner les options : le nom du joueur en sélectionnant les lettres de l'alphabet au-dessus ou au-dessous, le vaisseau du joueur, son arme, la carte avec les ennemis différents débloqués, 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.



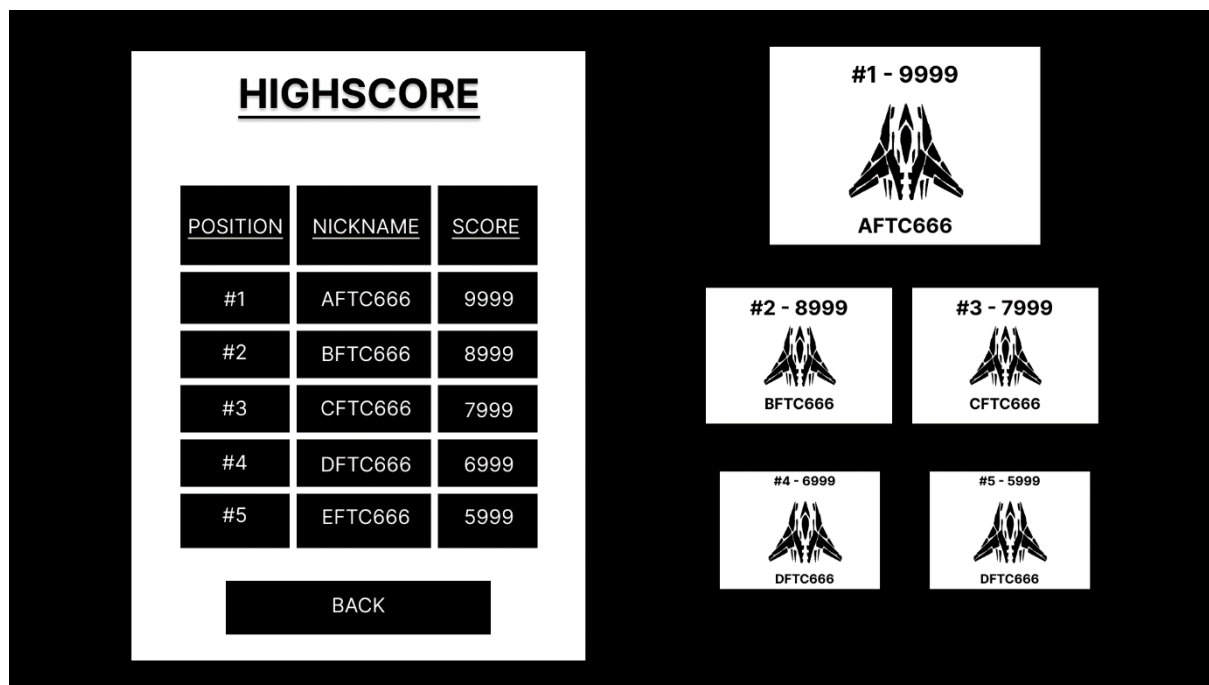
- Multiplayer

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



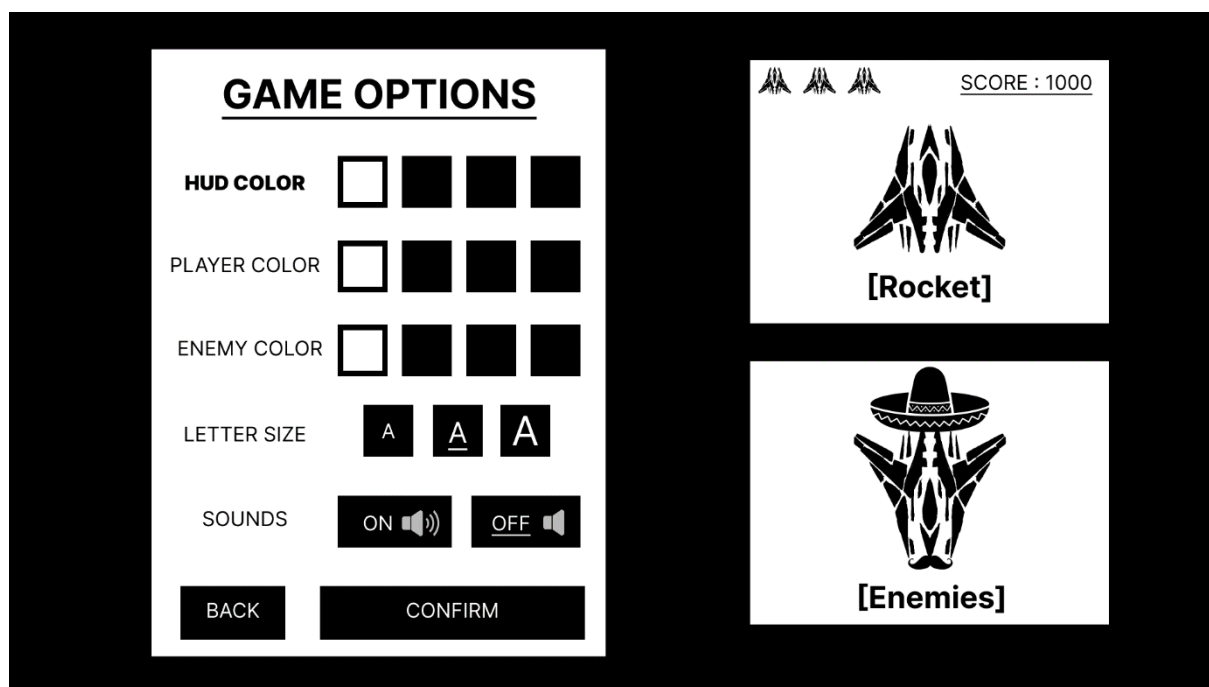
- Highscore

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.

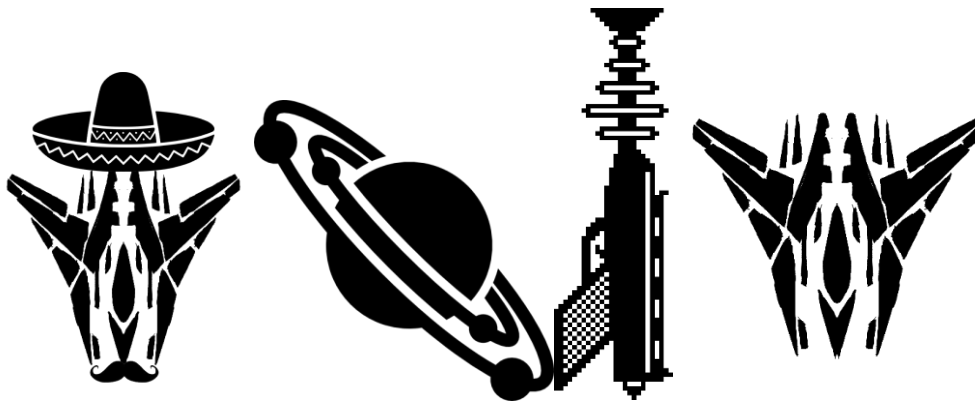


- GameOptions

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 traduirait dans le gameplay du jeu.



- 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 :



- 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 a été utilisée de différentes manières pour différencier les textes des films de l'espace :

- Titre du jeu



- Titre du panel

NEW GAME

- Titre des options modifiables

NICKNAME

- Informations sauvegarder

AFTC666

- Informations dans le jeu

1# - AFTC666

- Assets dans le jeu:

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

- Vaisseaux spatiaux



- Background



- Armes, Cartes et effets visuels



- 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 qui fonctionnera comme des boutons.



- 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.



- Multiplayer

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



- Highscore

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



- 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.



1.4. Test

1.4.1. Test d'utilisabilité

Menu	Description	Conditio ns	Étapes de Test	Données	Résultat attendu
Menu d'accue il	Les boutons présélection nés sont visuellement différenciés	Utilisatio n 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 présélectionn ée.
Menu d'accue il	Les boutons sélectionnés permettent d'accéder au menu corresponda nt	Utilisatio n 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 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.
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. Enter 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	1. Être dans l'option Rocket, Gun, Map ou Difficulty. 2. Sélectionner l'une des cases disponibles		Les options sélectionnées sont enregistrées et affichées dans le panneau d'aperçu du jeu à droite.

		ou de la souris.	3. Enter pour sauvegarder la sélection		
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 Hishcore	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 naviguer sur le menu à l'aide de clavier ou de la souris.	1. Sélectionner l'option de retourner 2. Appuyer sur Enter		Il retournera au menu.
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.

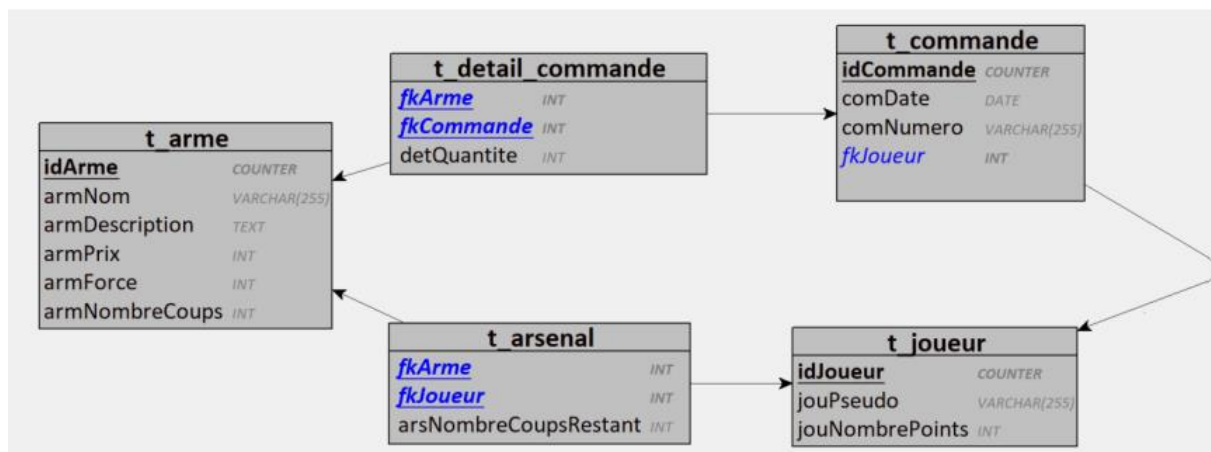
Option	La taille de la police peut être modifiée	Le joueur peut naviguer sur le menu options à l'aide de clavier ou de la souris.	1. Être dans l'option Letter Size 2. Sélectionner l'une des cases disponibles 3. Enter pour sauvegarder la sélection		Les changements de la police seront affichés visuellement 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 options à l'aide de clavier ou de la souris.	1. Être dans l'option Screen 2. Utiliser les flèches vers le haut ou vers le bas ou le clavier pour sélectionner l'option. 3. Enter pour confirmer		Les changements de la résolution seront modifiée.

2. DB

2.1. Introduction DB « db_space_invaders »

Une « 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 (MDL), 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. Ce schéma ne sera utilisé que pour analyser une base de données supposée pour une version de Spicy Invaders différente de celle construite dans le cadre de ce projet.



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 et vous donnera 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 :

```
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, nous allons initialiser MySQL et nous pouvons demander avec 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, nous commençons par utiliser l'utilisateur "Root". Avec cet utilisateur, nous aurons les permissions appropriées 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, nous allons 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. 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. L'identité du joueur, son nom et son score sont affichés.

```
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 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.

```
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@aie
PlayerTwo	Missile nucl@aie
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.idJoueur 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, nous avons vu 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, nous utiliserons 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.

3. POO

3.1. Introduction

Con el siguiente proyecto, se realizara de forma abierta un pequeño videojuego en consola en Microsoft Visual Studio en código C#.

El objetivo es realizar un proyecto similar al famoso videojuego Space Invaders, pero en este caso, Spacy Invaders. Debe seguir los objetivos básicos como el jugador, enemigos diversos y proyectiles. A parte de los elementos jugables, también hay que añadir un sistema de victoria y registro de puntuación.

Para la presentación del proyecto, se deberá presentar diversos menus principales como el menú principal, highscore y el juego.

3.2. Analyse fonctionnelle

3.3. Analyse technique

3.3.1. Diagramme de basse

3.3.2. Explication (docfx)

- Main Programm
- Motor de Jeu
- Player
 - IsAlive
 - Draw
 - Move
- Enemies
 - IsAlive
 - Draw
 - Move
- Bullet
 - Draw
 - Move
 - Destroy
- UI
- Base de données

3.4. Test Unitaire

4. Conclusions

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/>