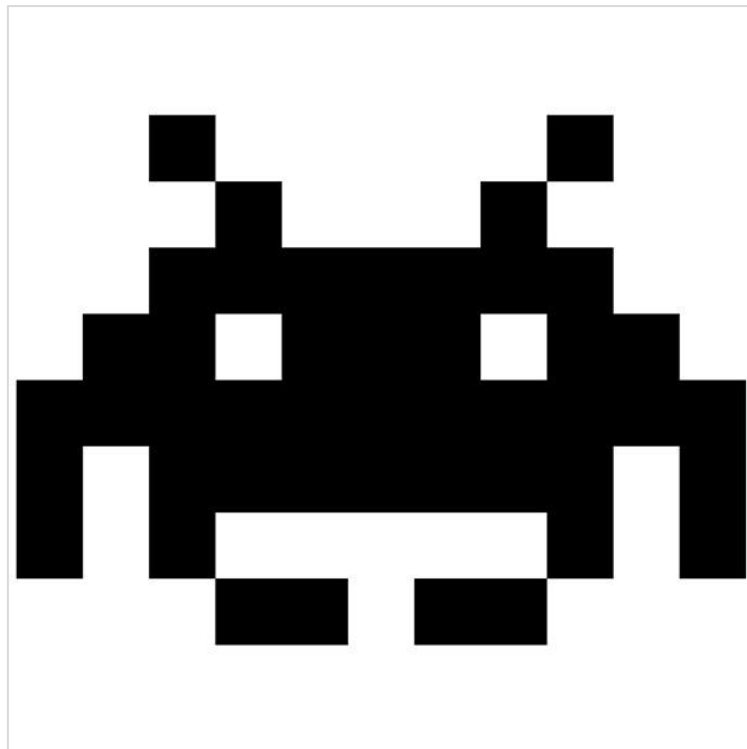


# P\_SpicyInvaders



Lucas Simões Pólvora - MID2 / GRP2D

Vennes – ETML

28.08.2023 – 03.11.2023

Mathieu Meylan, Aurélie Curchod, Xavier Carrel

## Table des matières

1.	Programmation Orienté Object.....	1
1.1	Introduction.....	1
1.2	Analyse .....	1
1.3	Conclusion .....	3
2.	Base de Données .....	3
2.1	Importer les données et le schéma de base de données.....	3
2.2	Gestion des utilisateurs .....	3
2.2.1	Création des Users.....	3
2.2.2	Rôle administrateur .....	4
2.2.3	Rôle joueur .....	4
2.2.4	Rôle gestionnaire de la boutique .....	4
2.3	Requêtes de sélection .....	5
2.4	Création des index.....	8
2.5	Backup et Restore.....	9
3.	Expérience Utilisateur .....	9
3.1	Introduction.....	9
3.2	Analyse .....	10
3.2.1	Persona .....	10
3.2.2	Palette Graphique .....	11
3.2.3	Eco-conception .....	11
3.2.4	Accessibilité .....	11
3.3	Conception .....	12
3.3.1	Design de l'application .....	12
3.3.2	Persona.....	21
3.4	Test .....	22
3.5	Conclusion .....	23

# 1. Programmation Orienté Object

## 1.1 Introduction

Ce projet a été fait afin d'accroître nos connaissances dans la Programmation Orienté Objet (POO) et qu'on connaisse plus sur la programmation de jeux vidéo. La POO est une technologie qui simplifie beaucoup la programmation de n'importe quelle application car cela nous fait raccourcir notre code et ce code est plus simple à structurer.

Afin d'avoir un projet encore plus instructif, j'ai décidé de faire ce projet en WPF, ce qui inclus avoir une manière de penser plus poussée et de nouvelles commandes inconnues jusqu'à maintenant.

Dans ce projet, j'ai donc utilisé Visual Studio 2022 pour faire la partie programmation, docfx pour faire mon rapport et IceScrum pour avoir une gestion du projet 1. Ce rapport aura une analyse de mon code et une conclusion pour finir le projet.

## 1.2 Analyse

Afin de pouvoir avoir tous les aspects de mon code, je vous invite à ouvrir les PDF qui montrent la structure du code. Ces PDF sont dans le même dossier que ce dernier sur GitHub.

Pour la MainWindow, qui est la page du jeu, j'ai fait une gameloop qui rassemblait toutes les updates et toutes les manières de tuer ou être tué. Ce qui veut dire que si un ennemi faisait une collision avec un joueur, le joueur mourrait et serait amené dans la page du Game over. Pour aller sur le Game Over j'ai utilisé un tick, ce qui permet de faire tous les refresh et faire la méthode à chaque refresh. Cette page se connecte aux balles, ennemis, au joueur et au score afin de pouvoir attribuer des valeurs, par exemple accroître le score, diminuer le score des ennemis, et encore d'autres choses.

Pour le GameOverLose, c'est très simple. On affiche que les scores et on fait un calcul pour avoir la valeur du score net. Ce calcul est juste score obtenu – ennemis restants \* 10. Cette page contient aussi un bouton pour aller sur une page afin de mettre son pseudo, elle sera décrite plus tard dans le rapport.

Le GameOverWin, est la même chose, je l'avais créé avec un espoir de rajouter quelques fonctionnalités, mais dû au temps qu'il me restait et la difficulté de coder la fonctionnalité, j'ai abandonné.

La page Help est juste une page avec les touches afin de savoir comment jouer. Elle a juste un bouton pour revenir au menu d'accueil.

La page pour le highscore est une page qui fait une connexion à la DB afin d'avoir les meilleurs scores des joueurs, elle affiche les 10 meilleurs joueurs avec leur nom et leur score. Un bouton est aussi disponible en haut à gauche de l'écran afin de revenir au menu d'accueil.

La page MettrePseudo, est une page simple, elle permet de mettre son pseudo dans une textBox pour ensuite cliquer sur continuer et mettre sur la DB son nom et score.

La page Menu est la page du menu d'accueil, elle contient les boutons :

- « Quit », ce qui fait quitter le programme
- « Play », ce qui lance le jeu
- « Highscore », montre la page du Highscore
- « Help », qui montre la page pour aider le joueur

Elle contient aussi un titre : « Space Invaders ».

La page App, est une page mise par défaut qui contient des infos custom (fontes personnalisé, ...) et aussi la page de base où le joueur va lorsqu'il lance le programme.

Pour les model, tout ce qui est Programmation Orienté Objet, j'ai divisé toutes les classes en 6 :

- Ennemis : Tout ce qui est en lien avec les ennemis
- Joueur : Tout ce qui est en lien avec le joueur
- Score : Tout ce qui est en lien avec le score
- Database : Tout ce qui est en lien avec la base de données.
- Bullet : Tout ce qui est en lien avec les balles ennemies ou du joueur
- Config : Tout ce qui est pour la configuration de l'écran

Les ennemis ont quatre méthodes :

- Display : Montre les ennemis en début de partie
- Movement : fait le mouvement des ennemis
- MoreBoost : Faite les ennemis aller plus vite
- Update : sert à faire l'update des ennemis, dans ce cas, il n'y a que le mouvement qu'on a besoin d'update

Les joueurs ont quant à eux cinq méthodes :

- Display : Montre le joueur en début de partie
- MovementOn : Fait en sorte que le mouvement débute lorsque la touche est touchée
- MovementOFF : Fait en sorte que le mouvement s'arrête lorsqu'on retire la touche
- MovementAction : Fait le mouvement
- Update : Fait l'update du joueur : dans ce cas il n'y a que le movementAction

Le score a trois méthodes :

- Update : Fait l'update lorsque le joueur tue un ennemi
- writeScore : écris le score sur l'écran
- downADeadValue : Lorsque le joueur rate une balle, un ennemi vaut un point de moins qu'avant

La classe database a cinq méthodes :

- ConnectionToDBReading : Fait toutes les requêtes pour lire la base de données
- tryConnection : Elle est utilisé pour tester la connection avant d'entrer sur la page Highscore et s'il n'est pas connecté affiche un popup en disant qu'il y a une erreur de la db
- writeScore : écris le score et le pseudo du joueur sur la db
- ShowHighscoreNames : elle utilise la première méthode pour avoir tous les noms des 10 meilleurs dans la db
- ShowHighscoreScore : Elle utilise aussi la première méthode mais maintenant c'est pour avoir les 10 meilleurs scores dans la db

La classe Bullet a 7 méthodes car elle regroupe les balles ennemies et du joueur, ce sont les suivantes :

- EnemyBulletMaker : Crée les balles ennemies
- EnemyBuletCooldown : Met un cooldown au nombre de balles tirées par les ennemis
- enemyBulletMovement : Affiche le mouvement des balles ennemis
- PlayerBulletCooldown : Donne une balle au joueur après X temps
- playerBulletMaker : Crée une balle lorsque le joueur clique sur espace

- PlayerBulletMovement : Fait le mouvement de la balle du joueur
- Update : fait un update de tous les cooldowns et tous les mouvement des balles ennemies ou du joueur

La classe config est spéciale car elle n'a aucune méthode. En effet, elle n'a que des variables qui servent à mettre la fenêtre dans un endroit spécifique de l'écran, la taille de l'écran, la vitesse des ennemis, du joueur, des cooldowns des balles et du nombre d'ennemis. J'ai fait cela afin que toutes les pages aient les mêmes paramètres et si je veux changer quelque chose il faut juste que j'aille dans cette classe.

### 1.3 Conclusion

Ce projet a été très instructif pour moi car j'ai appris à coder en WPF, mais aussi en POO, ce qui fait que le projet a été plus que bénéfique pour ma connaissance en développement d'applications en graphique. J'ai compris comment utiliser la POO et surtout compris à quel point la POO améliore notre code et la visibilité de notre code. Je remercie donc les auteurs de ce projet, pour m'avoir enseigné tous ces aspects techniques.

Signature :

Lucas Simões Pólvora

## 2. Base de Données

### 2.1 Importer les données et le schéma de base de données

Pour débiter cette étape il faut vérifier que l'utilisateur est dans le groupe docker-users. S'il n'y est pas, il faut aller sur PowerShell en admin et faire la commande suivante :

```
« net localgroup docker-users "eleve-windows-id (P...)" /ADD »
```

Ensuite il faut mettre le container sur docker, pour cela il faut mettre le fichier docker sur le disque C:\ et dans le dossier où il y a le fichier .yaml, on lance le cmd. Puis faire la commande :

```
« docker exec -it "id du container" bash » et « mysql -uroot -p »
```

Et ensuite mettre le mot de passe (par défaut : root). Pour finir on peut mettre le fichier SQL dans le container en faisant :

```
« docker exec -i "mysql-container-id" mysql -uroot -proot < "fichier.sql" »
```

### 2.2 Gestion des utilisateurs

#### 2.2.1 Création des Users

En premier, j'ai créé l'utilisateur qui va servir de test pour le rôle « administrator » :

```
CREATE USER 'userAdmin'@'localhost' IDENTIFIED BY '123';
```

Cette commande commence par crée un user (utilisateur) qui va s'appeler « userAdmin » et qui va être stocké localement par « localhost ». Il va être identifié par 123 ce qui veut dire que le mot de passe va être 123. La même commande sera utilisée pour les deux prochains mais avec des noms différents.

Ensuite, celui qui servira de test pour le rôle « Player » :

```
CREATE USER 'userPlayer'@'localhost' IDENTIFIED BY '123';
```

Et pour finir celui qui servira de test pour le rôle « ShopManager » :

```
CREATE USER 'userShopManager'@'localhost' IDENTIFIED BY '123';
```

### 2.2.2 Rôle administrateur

Pour commencer, j'ai créé le rôle en faisant la commande suivante :

```
CREATE ROLE 'r_administrator';
```

Cette commande est très simple, il y a la création du rôle avec « CREATE ROLE » et le nom du rôle qui, dans ce cas, sera « administrator »

Ensuite, j'ai attribué à ce rôle tous les privilèges pour la base de données db\_space\_invaders :

```
GRANT ALL PRIVILEGES ON db_space_invaders.* TO 'r_administrator' WITH GRANT OPTION
```

Cette commande est divisée en 4 étapes :

- 1) GRANT ALL PRIVILEGES : Donner tout type de privilèges
- 2) ON db\_space\_invaders.\* : Les privilèges donnés précédemment seront limités à la base de données à db\_space\_invaders
- 3) TO 'administrator' : Les personnes impactées par cette commande seront celles qui sont et seront dans le rôle « administrator »
- 4) WITH GRANT OPTION : Cette partie donne la permission aux personnes impactées de donner des privilèges aux autres dans la base de données précisée à l'étape 2.

Pour terminer, j'ai mis l'utilisateur créé précédemment dans ce rôle :

```
GRANT 'r_administrator' TO 'userAdmin'@'localhost';
```

Cette commande donne tous les privilèges de « administrator » (la commande précédente nous montre quels sont ces privilèges) à userAdmin créé au point 2.2.1.

### 2.2.3 Rôle joueur

Pour cette étape nous allons faire les mêmes commandes mais avec un nom de rôle différent.

```
CREATE ROLE 'r_player';
```

Ensuite, il faut donner maintenant les privilèges, on va donc donner les privilèges de création et lecture dans la table commande et les privilèges de lecture sur la table armes :

```
GRANT INSERT, SELECT ON db_space_invaders.t_commande TO 'r_player';
```

Pour terminer, je vais mettre l'utilisateur dans le rôle :

```
GRANT 'r_player' TO 'userPlayer'@'localhost';
```

### 2.2.4 Rôle gestionnaire de la boutique

Pour débiter créons le rôle :

```
CREATE ROLE 'r_shopManager';
```

Ensuite nous allons attribuer ses privilèges :

```
GRANT SELECT ON db_space_invaders.t_joueur TO 'r_shopManager'  
GRANT SELECT ON db_space_invaders.t_detail_commande TO 'r_shopManager'  
GRANT SELECT ON db_space_invaders.t_commande TO 'r_shopManager'  
GRANT ALTER, SELECT, DELETE ON db_space_invaders.t_arme TO 'r_shopManager';
```

1<sup>ère</sup> ligne : On donne le droit de lire dans la table t\_joueur.

2<sup>ème</sup> ligne : Même chose que la première mais sur t\_detail\_commande

3<sup>ème</sup> ligne : Même chose que les deux précédentes mais sur la table t\_commande

4<sup>ème</sup> ligne : On donne le droit de modifier, lire et supprimer dans la table t\_arme.

Pour finir ce point nous allons donner le rôle à son utilisateur :

```
GRANT 'r_shopManager' TO 'userShopManager'@'localhost';
```

### 2.3 Requêtes de sélection

Requête n°1 :

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

Sur cette commande on va faire en sorte que la colonne pseudo dans la table joueurs soit montrée avec le "SELECT jouPseudo FROM 't\_joueur'". Ensuite, on va organiser les réponses pour faire en sorte qu'elles soient dans l'ordre décroissant. Pour finir, on va limiter à 5 réponses pour que la console nous montre les 5 meilleurs.

Requête n°2 :

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

Sur cette commande, on va avoir 3 colonnes :

1. Le prix maximum de l'arme avec le "MAX(armPrix)" et ensuite on va la renommer en "PrixMaximum" avec le "as"
2. Le prix minimum de l'arme avec le "MIN(armPrix)" et ensuite on va la renommer en "PrixMinimum" avec le "as"
3. Le prix moyen des armes avec le "AVG(armPrix)"(average) et ensuite on va la renommer en "PrixMoyen" avec le "as"

Ensuite, on va tirer toutes ces informations de la table "armPrix" avec "FROM 't\_arme'"

Requête n°3 :

```
SELECT fkJoueur as idJoueur,  
COUNT(idCommande) as NombreCommandes  
FROM `t_commande`  
GROUP BY fkJoueur;
```

Sur cette commande je commence par appeler la "fkJoueur" avec "SELECT fkJoueur" et je le renomme "idJoueur" avec "as idJoueur". Ensuite, je compte le nombre de fois qu'une commande est appelée par le même joueur avec "COUNT(idCommande)" et le renomme "NombreCommandes". Je retire toutes ces colonnes depuis la table "t\_commande" et les groupe par fkJoueur avec "GROUP BY fkJoueur"

Requête n°4 :

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

Cette commande ressemble à la précédente mais on rajoute le "HAVING" qui est une condition pour le "GROUP BY". Cette condition permet d'afficher que les joueurs ayant fait plus de 2 commandes

Requête n°5 :

```
SELECT t_joueur.jouPseudo, t_arme.armNom  
FROM `t_joueur`  
INNER JOIN t_commande  
ON t_joueur.idJoueur = t_commande.fkJoueur  
INNER JOIN t_detail_commande  
ON t_commande.idCommande = t_detail_commande.fkCommande  
INNER JOIN t_arme  
ON t_detail_commande.fkArme = t_arme.idArme;
```

Sur cette commande, on sélectionne le pseudo du joueur depuis la table "t\_joueur" et le nom de l'arme depuis la table "t\_arme". Ensuite, on peut constater qu'on part depuis la table 'joueur', après on la relie avec un "INNER JOIN" la table commande pour avoir aussi les informations de cette table. Et on répète cette commande jusqu'à être dans la table arme pour ensuite pouvoir afficher toutes les armes de chaque joueur.

Requête n°6 :



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

Sur cette commande, on choisi d'afficher l'id du joueur avec toutes ses dépenses lorsqu'il a acheté des armes. Pour ce faire on affiche déjà l'id du joueur et la somme des commandes fois le prix de l'arme et on renomme cette colonne en TotalDepense. Et ensuite nous allons faire la même chose que la commande d'avant pour arriver à la table de l'arme. Ensuite, on va grouper les sommes d'argents que les joueurs ont dépensé pour que les joueurs n'apparaissent qu'une seule fois dans le tableau avec GROUP BY "t\_joueur.idJoueur". Et pour finir on va organiser les résultats pour qu'il n'y ait que les 10 joueurs qui ont le plus dépensé.

Requête n°7 :

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

Sur cette commande, on va afficher l'id des joueurs, leur pseudo, leurs points et le numéro de leurs commandes. Pour avoir toutes les informations, on va joindre les tableaux afin que les informations soient cohérentes (les commandes des joueurs sont à côté de leur ID).

Requête N°8 :

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

Sur cette commande on va sélectionner le pseudo des joueurs et le numéro de leurs commandes. Ici, j'ai utilisé un "RIGHT JOIN" parce qu'ainsi, je pourrais voir toutes les commandes et qui les a faites. Le "RIGHT JOIN" sert à avoir toutes les informations de la deuxième table et que les informations de la première table qui ont un rapport avec la deuxième. Il y aura donc plusieurs fois le même pseudo.

Requête N°9 :

```
SELECT t_joueur.jouPseudo, SUM(t_detail_commande.detQuantiteCommande)
FROM t_joueur
LEFT JOIN t_commande
ON t_joueur.idJoueur = t_commande.fkJoueur
LEFT JOIN t_detail_commande
ON t_commande.idCommande = t_detail_commande.fkCommande
LEFT JOIN t_arme
ON t_detail_commande.fkArme = t_arme.idArme
GROUP BY t_joueur.idJoueur;
```

Sur cette commande, on va afficher les pseudos des joueurs et le nombre d'armes qu'ils ont acheté. Pour cela on va faire comme à la requête N°6 mais nous allons faire avec le "LEFT JOIN" pour que même si le joueur n'a rien acheté il apparaisse sur la table. Le "LEFT JOIN" sert à avoir toutes les informations de la première table et les informations de la deuxième table qui sont liées à la première.

Requête n°10 :

```
SELECT t_joueur.jouPseudo, COUNT(DISTINCT t_arme.idArme)
FROM t_joueur
LEFT JOIN t_commande
ON t_joueur.idJoueur = t_commande.fkJoueur
LEFT JOIN t_detail_commande
ON t_commande.idCommande = t_detail_commande.fkCommande
LEFT JOIN t_arme
ON t_detail_commande.fkArme = t_arme.idArme
GROUP BY t_joueur.idJoueur
HAVING COUNT(DISTINCT t_arme.idArme) > 3;
```

Sur cette commande, on va sélectionner le pseudo du joueur et le nombre d'armes différentes il a. Comme dans la requête précédente, on va faire un "LEFT JOIN" de joueur jusqu'à arme. On va ensuite regrouper toutes les réponses par l'id du joueur (le pseudo aurait aussi pu être utilisé) et on va faire en sorte que les joueurs qui s'affichent aient plus que 3 armes avec la condition de la dernière ligne.

## 2.4 Création des index

a) Car ce sont des primary Key ou des foreign Key.

- b) Les avantages des Index sont que les recherches d'infos sont beaucoup plus rapides et un inconvénient majeur est le ralentissement des écritures. Donc il faut bien réfléchir si nous allons plus écrire ou nous allons plutôt lire la base de données avant de faire un index. Par exemple si c'est une archive alors nous allons plutôt lire donc on pourra mettre des index
- c) Les joueurs car au fur et à mesure que le jeu sera évolué plus il y aura de joueurs donc il faudra plus de temps pour rechercher un joueur précis

## 2.5 Backup et Restore

Afin d'éviter toute perte de données, on doit faire un backup pour avoir toujours une base de données sauve. Pour faire cela nous allons exécuter la commande suivante :

```
mysqldump -u votre_nom_utilisateur -p db_space_invaders > backup_db_space_invaders.sql
```

Cette commande utilise la commande mysqldump qui sert à faire des backups et encore d'autres choses. Après on va s'authentifier afin d'avoir accès à la base données et pour finir on dit que la base de données va être créée avec le nom qu'on lui a donné à la fin (bd > backup).

Enfin pour restorer la base de données, on va faire la commande suivante :

```
mysql -u votre_nom_utilisateur -p db_space_invaders < backup_db_space_invaders.sql
```

La commande est presque la même, les seules 2 différences sont :

- Au lieu de faire une commande mysqldump on va faire une commande mysql
- Au lieu de faire un signe > nous allons faire un signe < qui va permettre d'implémenter la base de données.

## 3. Expérience Utilisateur

### 3.1 Introduction

Ce projet a été fait afin de modéliser le jeu space invader afin qu'il soit écologique et qu'on ait une idée de maquette à faire pour notre projet en POO. L'UX (expérience utilisateur) est important dans un projet lorsque l'on veut que le produit plaise à la majorité des personnes.

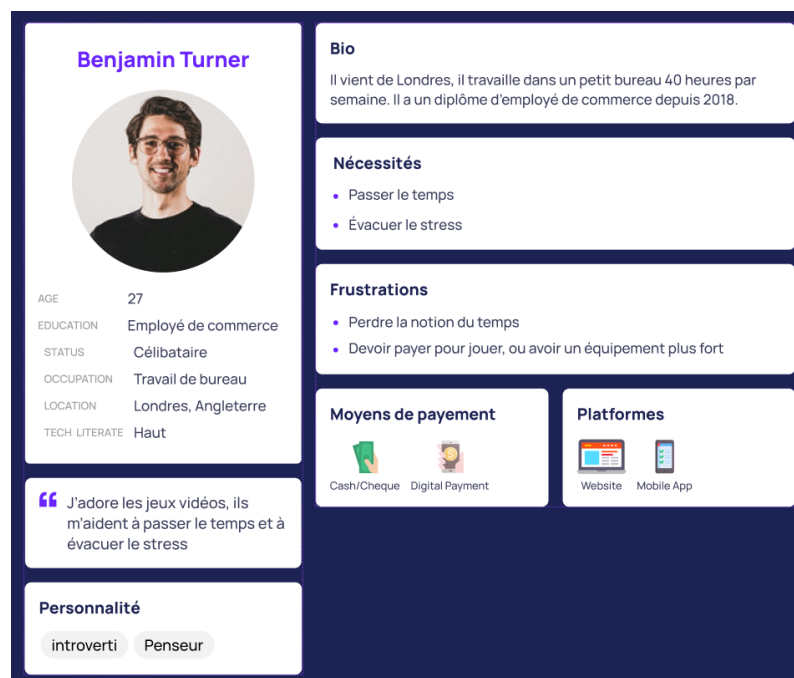
Dans ce projet nous avons utilisé une interface de design, dans mon cas c'est FIGMA. Je l'ai utilisé afin de faire le design des Persona, que l'on va parler plus tard, et le design de l'application.

Ce rapport aura une analyse de toutes les caractéristiques abordées dans ce sujet (Persona, palette graphique, éco-conception et accessibilité), les tests d'utilisabilité et une conclusion.

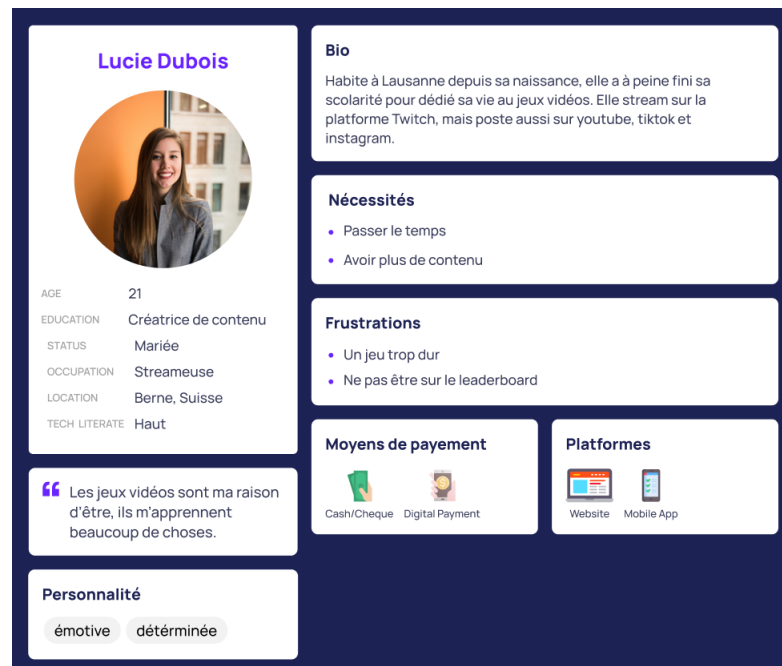
## 3.2 Analyse

### 3.2.1 Persona

Pour ma première persona, je voulais faire quelqu'un dont la profession n'avait rien avoir avec les jeux vidéo, mais qui joue quand même aux jeux vidéo. Afin d'avoir un peu d'originalité, j'ai demandé à chat GPT de me donner un nom anglophone et pris une photo de unsplash. Voici la persona qui en est ressorti :

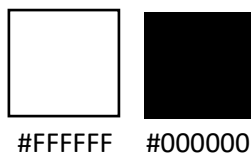


Pour ma deuxième persona, je voulais quelqu'un dont la vie tournait aux jeux vidéo, pour avoir un contraste entre les deux à la seule différence que le nom est francophone. Le nom et l'image viennent des mêmes sources que la première persona. Voici le résultat final :

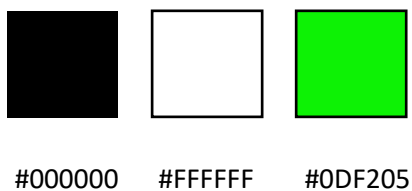


### 3.2.2 Palette Graphique

La palette graphique claire est très simple, elle est juste composée des couleurs suivantes :



J'ai aussi fait une palette graphique foncée, elle est composée des couleurs suivantes :



La deuxième palette graphique a comme couleur principale le noir tandis que la première est le blanc

### 3.2.3 Eco-conception

Pour l'éco-conception j'ai pensé à faire un fond foncé ce qui utilise moins d'énergie, l'utilisation d'aucune vidéo ou image : Aucune image ou vidéo ne sera dans le programme, afin que le chargement soit moins lourd.

### 3.2.4 Accessibilité

Pour l'accessibilité, j'ai ajouté la fonctionnalité de changer de langue, les langues que j'ai faites sont le français et l'anglais, mais il y a toujours la possibilité d'ajouter si nécessaire. Une deuxième

option pour l'accessibilité que j'ai choisie et le faite de pouvoir changer de palette de couleurs, pour les personnes qui sont sensibles à la luminosité peuvent mettre le thème plus foncé.

### 3.3 Conception

Le design de l'application et des Persona que j'ai conçu ont été fait sur FIGMA.

#### 3.3.1 Design de l'application

En commençant le design, j'ai voulu quelque chose de classique comme un vrai space invader. C'est un des premiers jeux, donc le design est très basique. J'ai donc opté pour un fond sombre avec des petits points blancs pour représenter les étoiles. Voici ce que cela a donné :



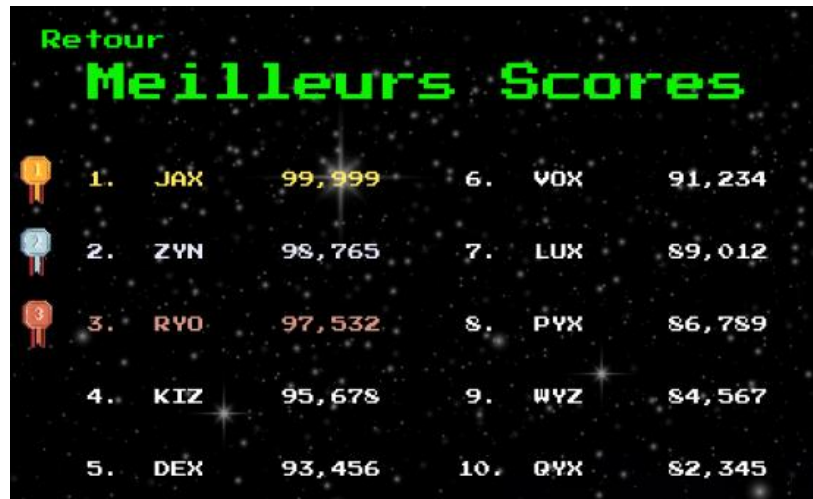
Ici nous avons un prototype cliquable du menu d'accueil, en effet, nous pouvons constater un titre et plusieurs labels qui ensuite seront redirigés aux pages indiquées.

Si l'on clique sur jouer, nous allons être dirigé vers la page suivante :



Ici nous avons la possibilité de jouer tout seul ou à deux. Nous avons aussi le choix de changer les ennemis, il faut juste cliquer sur l'ennemi pour qu'il change. Nous pouvons aussi revenir au menu avec le bouton retour.

Si on revient au menu et qu'on clique sur Meilleurs Score, la page suivante apparaîtra :



Ici nous pouvons avoir les meilleurs scores effectués dans le jeu. Nous pouvons voir le pseudo et le nom. Comme sur la page jouer, on peut voir un bouton retour afin de revenir au menu d'accueil.

Ensuite si on clique sur le bouton Shop dans le menu, la page suivante apparaît :



Sur cette page nous pouvons voir les offres disponibles, les statistiques des vaisseaux et surtout ce qui nous intéresse le bouton pour acheter et le vaisseau qui pointe vers la droite. En effet, si on clique sur le bouton pour acheter, le popup suivant apparaît :



Si on clique sur le vaisseau qui pointe vers la droite, une animation pour changer les offres apparaît et cela ressemblera à la page suivante :



Nous pouvons faire toujours cela jusqu'à ce qu'il n'y ait plus d'offres.

Si nous revenons au menu, et appuyons sur crédits, un défilé des crédits apparaît. Lucas Lordon m'a fourni l'exemple des crédits. La page ressemble à ceci :





Ici nous n'avons rien de sp  cial    part le bouton pour retourner au menu d'accueil.

Pour finir avec les pages, nous allons sur la page aide :



Cette page ne sert qu'   montrer les touches pour jouer

Ensuite, nous avons aussi l'option de faire changer le th  me. Nous pouvons le changer en haut    droite :





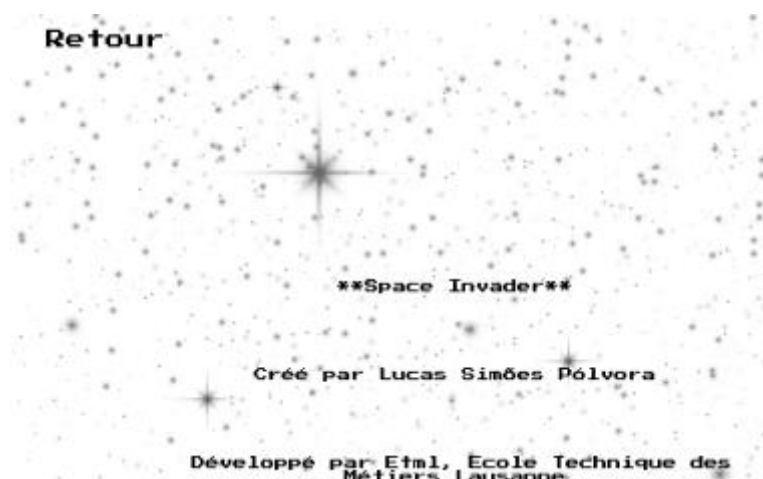
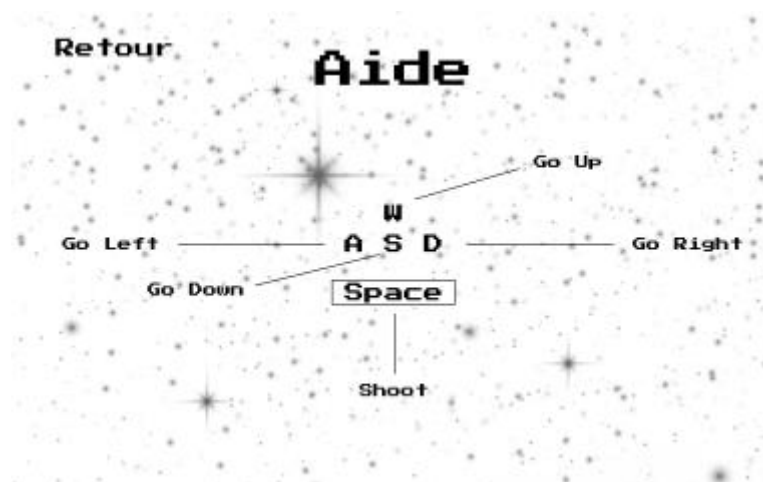
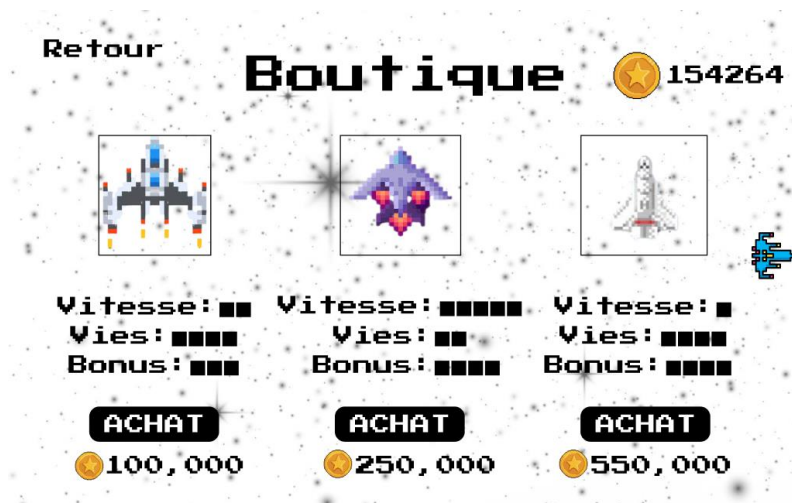
Le r  sultat pour les autres pages est la suivante :



Retour

## Meilleurs Scores

	1. JAX	99,999	6. VOX	91,234
	2. ZVN	98,765	7. LUX	89,012
	3. RYO	97,532	8. PVX	86,789
	4. KIZ	95,678	9. WYZ	84,567
	5. DEX	93,456	10. QYX	82,345

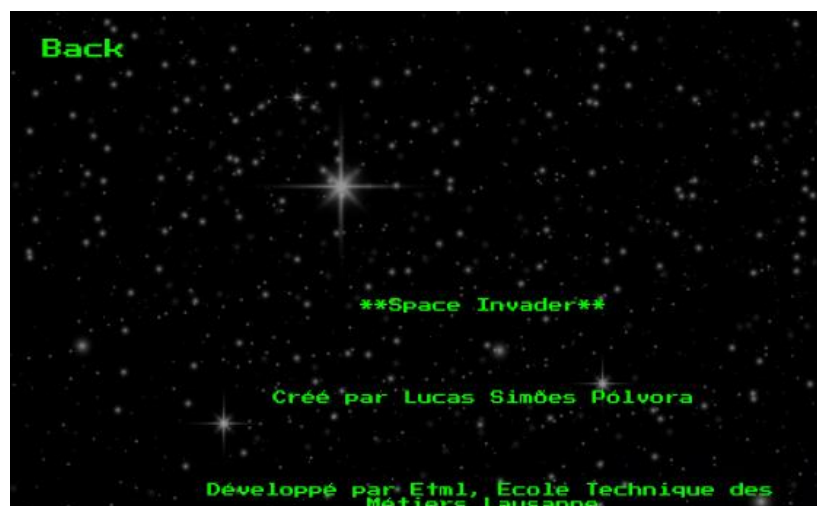


Pour finir nous pouvons aussi changer la langue en haut   gauche du menu d'accueil :



Le r  sultat pour les autres pages est la suivante :

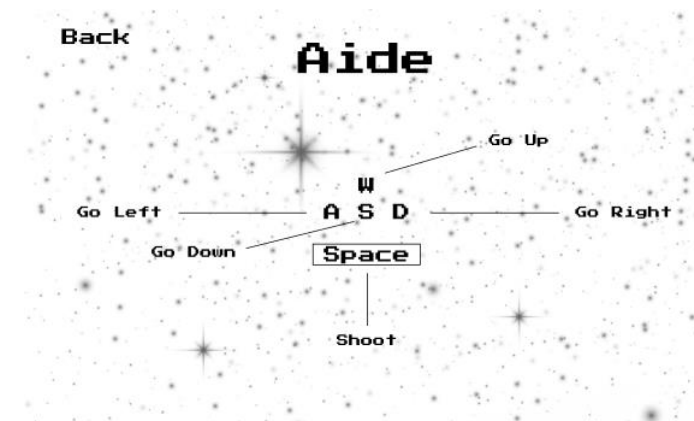




Le thème clair est aussi disponible en anglais, voici le résultat :








### 3.3.2 Persona

Les persona ont aussi été faits sur FIGMA. J'ai pris un asset d'une autre personne afin d'avoir un beau model de persona. L'asset vient de ce lien : <https://www.figma.com/community/file/881830156311997001>. J'ai juste eu à changer les infos afin d'avoir ma propre persona. Voici les deux persona que j'ai créé :

Benjamin Turner



AGE27

EDUCATIONEmployé de commerce

STATUSCélibataire

OCCUPATIONTravail de bureau

LOCATIONLondres, Angleterre

TECH LITERATEHaut

“ J'adore les jeux vidéos, ils m'aident à passer le temps et à évacuer le stress ”

Personnalité

introvertiPenseur

Bio

Il vient de Londres, il travaille dans un petit bureau 40 heures par semaine. Il a un diplôme d'employé de commerce depuis 2018.



Nécessités

- Passer le temps
- Évacuer le stress

Frustrations



- Perdre la notion du temps
- Devoir payer pour jouer, ou avoir un équipement plus fort

Moyens de paiement

Cash/ChequeDigital Payment

Plateformes

WebsiteMobile App

Lucie Dubois



AGE21

EDUCATIONCréatrice de contenu

STATUSMariée

OCCUPATIONStreameuse

LOCATIONBerne, Suisse

TECH LITERATEHaut

“ Les jeux vidéos sont ma raison d'être, ils m'apprennent beaucoup de choses. ”

Personnalité

émotivedéterminée

Bio

Habite à Lausanne depuis sa naissance, elle a à peine fini sa scolarité pour dédié sa vie au jeux vidéos. Elle stream sur la plateforme Twitch, mais poste aussi sur youtube, tiktok et instagram.

Nécessités

- Passer le temps
- Avoir plus de contenu

Frustrations

- Un jeu trop dur
- Ne pas être sur le leaderboard

Moyens de paiement




Cash/ChequeDigital Payment

Plateformes




WebsiteMobile App

### 3.4 Test

Nom	Description	Préconditions	Étapes du test	Données du test	Réponse voulue	Réponse reçue
Affichage du Highscore	Affichage de la fenêtre Highscore	Installer l'application	Cliquer sur le bouton Highscore dans le menu	-	Affichage du Highscore	Affichage du Highscore
Affichage du help	Affichage de la fenêtre Help	Installer l'application	Cliquer sur le bouton Help dans le menu	-	Affichage du Help	Affichage du Help



Affichage du Shop	Affichage de la fenêtre Shop	Installer l'application	Cliquer sur le bouton Shop dans le menu	-	Affichage du Shop	Affichage du Shop
Déroulement du shop	Affichage de plus d'offres dans le shop	Installer l'application et aller dans la page Shop	Cliquer sur le vaisseau qui va vers la droite dans la page Shop	-	Affichage de plus d'offres	Affichage de plus d'offres
Déroulement retour du Shop	Affichage des offres précédentes dans le Shop	Installer l'application et aller dans la page Shop	Cliquer sur le vaisseau qui va vers la droite dans la page Shop	-	Affichage des offres précédentes	Affichage des offres précédentes

### 3.5 Conclusion

Ce projet a été très instructif car j'ai appris à utiliser une interface de design, appris à penser aux clients de mon application et non seulement à mes avis. J'ai eu de la difficulté avec FIGMA au début mais au fur et à mesure, j'ai appris à aimer utiliser FIGMA, plus le temps passait, plus je découvrais des choses. C'était donc très enrichissant ! Apprendre à utiliser et appliquer les persona a été aussi un point clé de ce projet, ceci m'a aidé à avoir plusieurs points de vue différents. Cela est très important pour pouvoir plaire au maximum de clients et surtout pouvoir les divertir grâce au jeu fait pour que ce soit agréable à regarder et avoir une expérience agréable

Je remercie donc aux créateurs de ce projet, d'avoir fait ce projet instructif, afin de nous enseigner les bases de l'UX et d'outils de design.

Signature :

Lucas Simões Pólvora