**Shoot Me Up**



**Elève :** GonzaloJavier Herrera Egoavil

**Classe :** CID2B

**Lieu :** Vennes

**Périodes :** 80p (24p + 24p + 32p)

**Chefs du projet :**

* M. Antoine Mveng
* M. Jonathan Melly
* Mme. Aurélie Curchod

**Année** : 2024 - 2025

Table des matières

[1 Spécifications 3](#_Toc178926939)

[1.1 Titre 3](#_Toc178926940)

[1.2 Description 3](#_Toc178926941)

[1.3 Matériel et logiciels à disposition 3](#_Toc178926942)

[1.4 Prérequis 3](#_Toc178926943)

[2 INTRODUCTION 4](#_Toc178926944)

[3 Planifications 4](#_Toc178926945)

[4 PARTIE UX 4](#_Toc178926946)

[4.1 Personas 4](#_Toc178926947)

[4.1.1 Persona 1 4](#_Toc178926948)

[4.1.2 Persona 2 4](#_Toc178926949)

[4.2 Palette graphique 4](#_Toc178926950)

[4.2.1 Photo 4](#_Toc178926951)

[4.3 Eco-conception 4](#_Toc178926952)

[4.4 Accessibilité 4](#_Toc178926953)

[4.5 Conception 4](#_Toc178926954)

[4.5.1 Maquettes base-fidélités 4](#_Toc178926955)

[4.5.2 Maquette haute-fidélité 4](#_Toc178926956)

[4.5.3 Fonctionnalités supplémentaires tirées des Personas + élément original 4](#_Toc178926957)

[4.5.4 Prototype cliquable (FIGMA) 4](#_Toc178926958)

[4.5.5 Choix effectués 4](#_Toc178926959)

[4.6 Evaluation 4](#_Toc178926960)

[5 partie poo 4](#_Toc178926961)

[5.1 Analyses fonctionnelles (Users Stories) 4](#_Toc178926962)

[5.2 Automatique du code (manuel de référence) 4](#_Toc178926963)

[5.3 Schémas des classes 4](#_Toc178926964)

[5.4 Implémentation spécifique 4](#_Toc178926965)

[6 PARTIE DB 4](#_Toc178926966)

[6.1 MCD 4](#_Toc178926967)

[6.1.1 Description des tables 4](#_Toc178926968)

[6.1.2 Description de 4](#_Toc178926969)

[6.1.3 Description de 4](#_Toc178926970)

[6.2 MLD 4](#_Toc178926971)

[6.3 Script SQL 4](#_Toc178926972)

[6.4 Base de données 5](#_Toc178926973)

[6.5 Mise en place des types d’utilisateurs 5](#_Toc178926974)

[6.5.1 Explication 5](#_Toc178926975)

[7 Rapport de test 5](#_Toc178926976)

[7.1 Tests UX 5](#_Toc178926977)

[7.2 Tests POO 5](#_Toc178926978)

[7.3 Tests DB 5](#_Toc178926979)

[8 LIVRAIBLES 5](#_Toc178926980)

[8.1.1 Fichiers/Links 5](#_Toc178926981)

[9 USAGE DE L’IA 5](#_Toc178926982)

[9.1.1 Explication 5](#_Toc178926983)

[10 REFERENCES 5](#_Toc178926984)

[10.1 Liens UX 5](#_Toc178926985)

[10.2 Liens POO 5](#_Toc178926986)

[10.3 Liens DB 5](#_Toc178926987)

# Spécifications

## Titre

Shoot Me Up !

## Description

Créer un jeu similaire à « Space Invaders » avec le nom « Shoot Me Up » à l’aide de la programmation en C#, la gestion de base de données et l’UX.

## Matériel et logiciels à disposition

* Un PC ETML
* Accès à internet
* Excel/IceScrum
* GitHub
* Word
* Visual Studio 2022 (Partie OO)
* Figma (Partie UX)
* Looping (Partie DB)
* Docker/Uwamp (Partie DB)
* MySQL (Partie DB)
* Visual Studio Code/bloc de notes (Partie DB)

## Prérequis

* Modules de programmation de base
* Modules de bases de données de base
* ICT-320 en cours
* ICT-322 en cours
* ICT-106 en cours

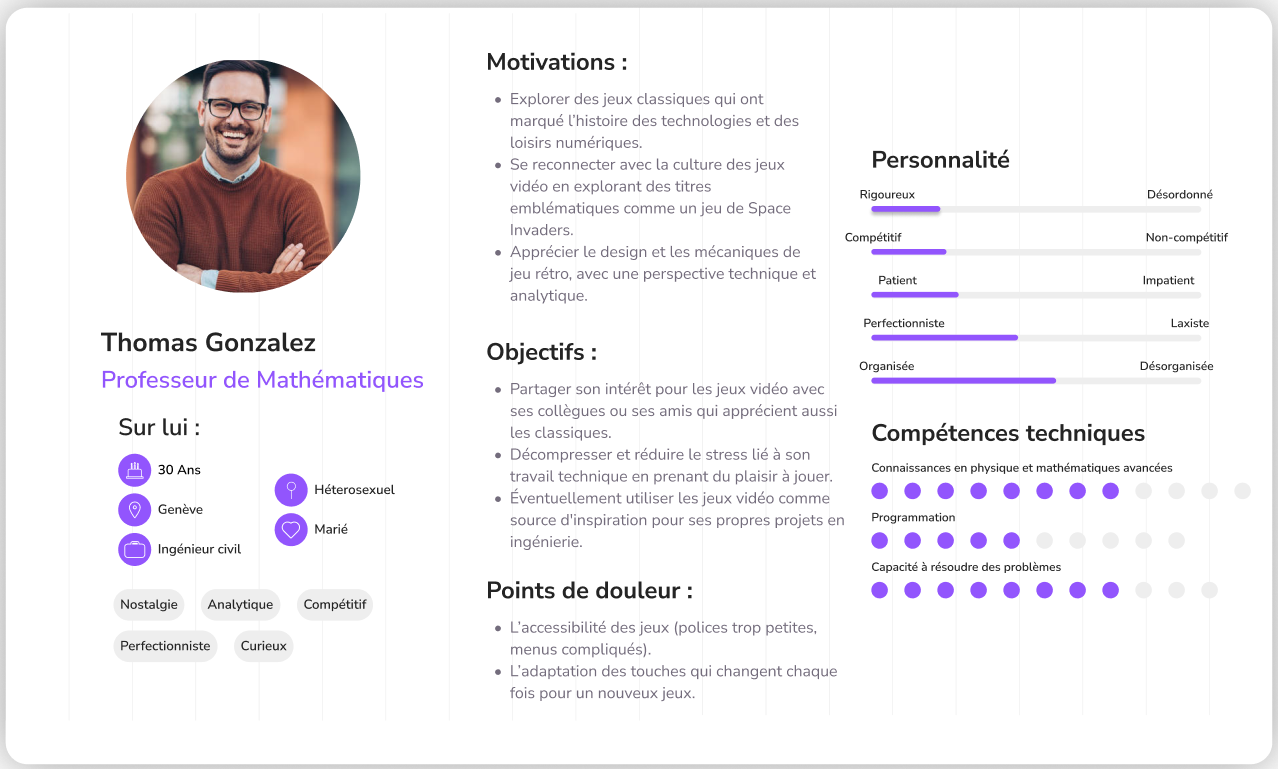
# INTRODUCTION

# Planifications

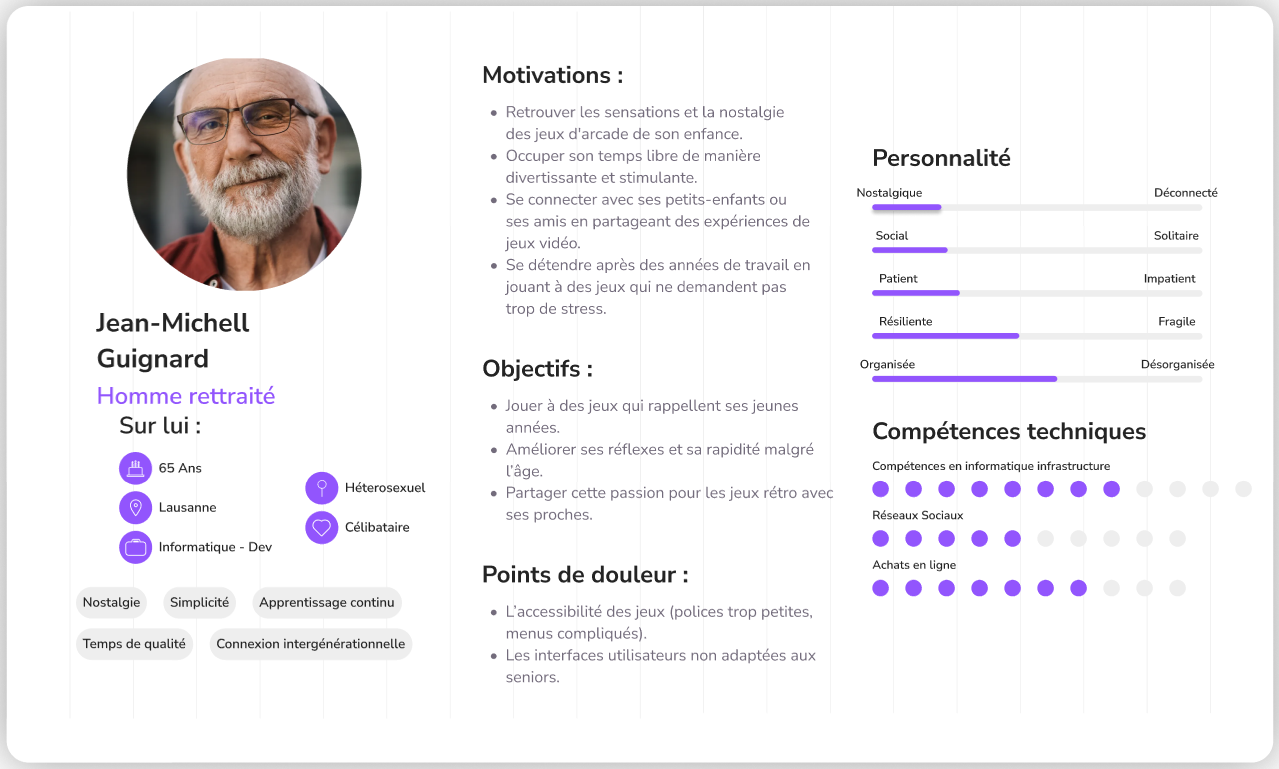
# PARTIE UX ([LIEN](https://github.com/GHerreraE/SHOOT-ME-UP-GONZALOHERRERA/tree/main/P_UX%20(Mme.%20Curchord)) DU DOSSIER UX)

## Personas

### Persona 1 ([**Lien**](https://www.figma.com/design/8BIWXqpve5LYrellf5aUNK/Personas_ShootMeUp?m=auto&t=L33tHLT6st1pxWDh-6) pour une meilleure visualisation)



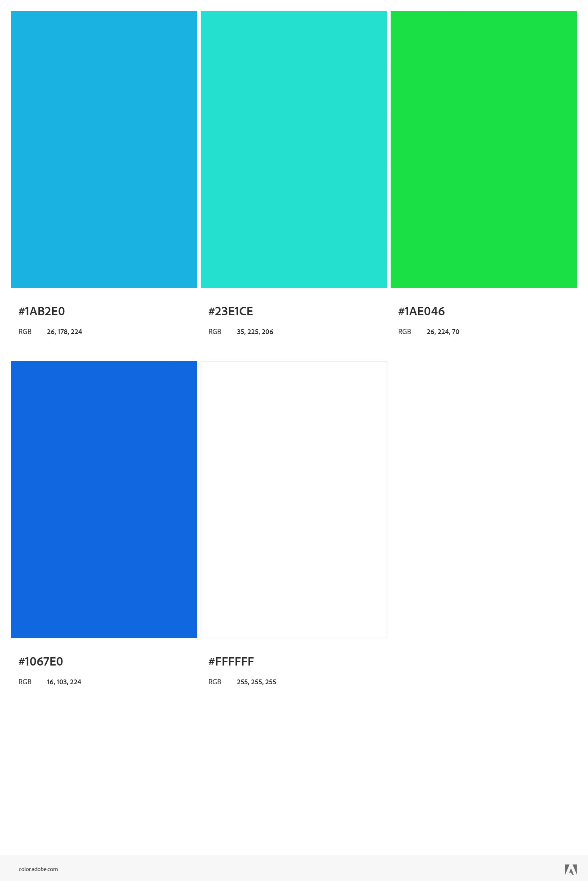
### Persona 2 ([**Lien**](https://www.figma.com/design/8BIWXqpve5LYrellf5aUNK/Personas_ShootMeUp?m=auto&t=L33tHLT6st1pxWDh-6) pour une meilleure visualisation)



## Palette graphique

### Description et photo

Après d’avoir essayé de coder mon jeu en console et avec WindForms, j’ai pu choisir et j’ai choisi de coder en mode console. Donc, la palette couleur est composé de couleurs plutôt vifs grâce à l’outil « [**Adobe Color**](https://color.adobe.com/fr/create/color-wheel)» pour bien remarquer les composants de mon programme. L’harmonie chromatique est du type Semblable, et le mode colorimétrique est RVB.



## Eco-conception

L’éco-conception dans ce projet est bien prise en compte, dès la partie programmation jusqu’à la partie UX. Voici deux pratiques d’éco-conception mises en place pour ce projet.

### Une technologie économe

Au moment de choisir le langage de programmation que j’allais utiliser, j’ai testé la programmation en C# en console et aussi avec WindForms. J’ai trouvé sympa l’option de faire avec WindForms puisque c’était cool le fait de pouvoir changer la forme avec des images ou icônes, mais j’ai fini pour choisir la programmation en console, le niveau d’éco-conception est beaucoup prise en compte avec lorsqu’on n’utilise pas d’images pour, on s’évite des classes supplémentaires pour la partie WindForms, et aussi si on veut, on peut créer un jeu simple.

### Simplicité

L’utilisation de ce jeu est simple et intuitive. Au moment qu’on lance le jeu, on peut se rendre compte au menu principal il y a des instructions pour savoir avec quelles touches, on peut jouer et ils sont avec une couleur qu’on peut bien visualiser comment interagir avec ce jeu.

## Accessibilité

L’implémentation d’accessibilité correspond bien aux besoins des personas. Par exemple :

**Thomas Gonzalez** est un professeur qui aimerait plutôt se détendre et se reconnecter avec des jeux qu’il jouait quand il était jeune. Par contre, il a des difficultés avec les polices trop petites et aussi, il a la peine avec les nouvelles dispositions des touches qui chaque fois sont implémentées dans chaque nouveau jeu. Donc, pour lui, j’ai implémenté un bouton au menu principal pour qu’il puisse cliquer et augmenter la taille des polices et aussi mettre en gras les éléments du jeu et pour la partie des touches, j’ai implémenté une option sur le menu principal ou l’utilisateur aura le choix de changer ses touches pour interagir avec le jeu.

**Jean-Michel Guignard** est un retraité qui veut juste reprendre ces jeux qui lui souviennent ses jeunes années mais vu qu’il est un peu âgé ses difficultés voyantes deviennent de plus en plus pire, donc l’implémentation des augmentation de taille des polices lui aidera beaucoup et aussi le fait de que je n’ai pas travaillé avec WindForms sera un point d’aide beaucoup plus agréable pour lui, puisque lui n’est pas adapté aux jeux qui ont beaucoup des graphiques comme Fortnite, Valorant, etc et ce jeu fait avec console, est beaucoup plus simple et minimaliste pour sa difficulté.

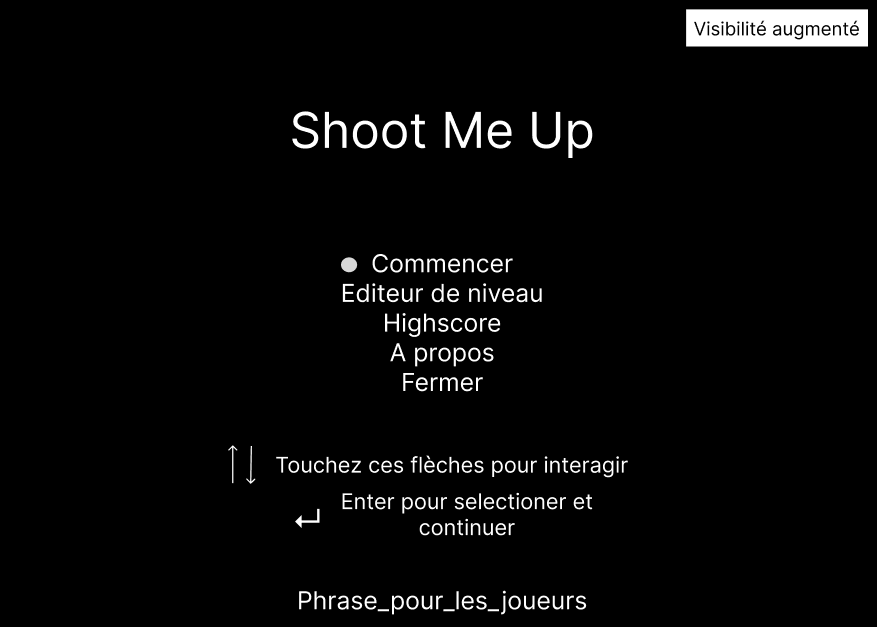
## Conception

### Maquettes base-fidélités ([**Lien**](https://www.figma.com/design/6LDRpce4gYfpBz2qYZNpz2/Maquette_BF_ShootMeUp_GHE?m=auto&t=TLMkRIyurZQH7xQH-6) pour une meilleure visualisation)

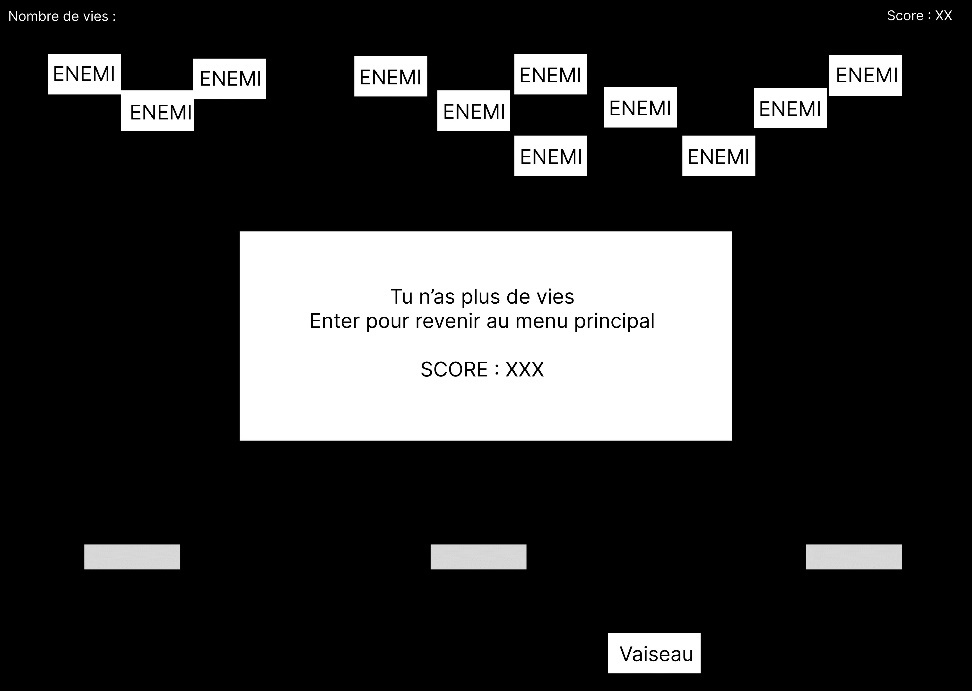
### Jeu en cours



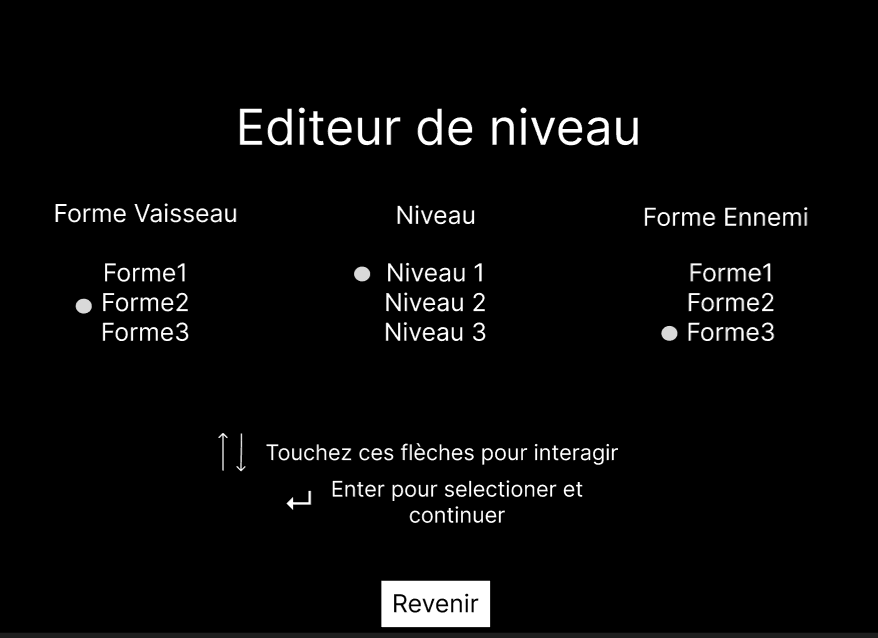
### Menu principal



### Fin de jeu



### Editeur de niveau



### Highscore

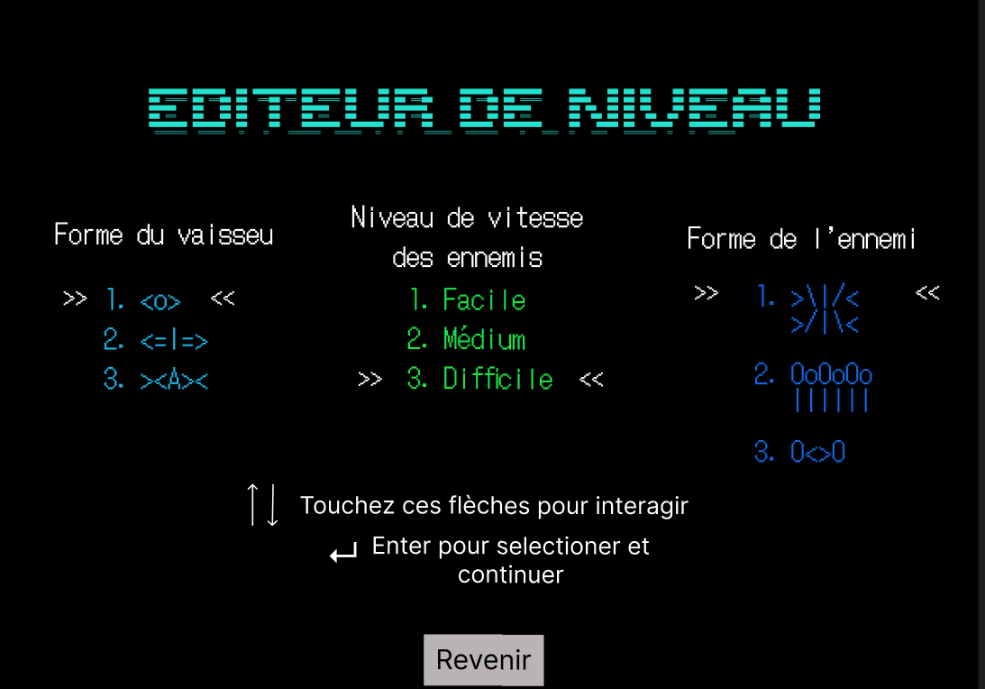


### A propos



### Maquette haute-fidélité ([**Lien**](https://www.figma.com/design/8zsOk8YzkmVsLJP5vLjhzi/Maquette_HF_ShootMeUp_GHE?m=auto&t=OTbOOt3kT1Nzcqv3-6) pour une meilleure visualisation)

Editeur de niveau



### Fonctionnalités supplémentaires tirées des Personas plus élément original

Menu principal : Bouton pour l’augmentation des tailles de polices + nouvelle option pour le changement des touches pour interagir avec le jeu.

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

Configuration : Nouvelle option pour les changements de touches qui seront par défaut les flèches, enter et espace. L’utilisateur aura le choix de changer cela pour les touches qu’ils veulent.

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

### Choix effectués

* **Journal de travail/Planification :** J’ai choisi de faire mon journal de travail et planification dans un fichier Excel, car je connaissais déjà comment ça marchait puis comment c’était un projet de plusieurs semaines et un projet regroupe trois autres projets, alors je me suis dit que séparer un fichier par chaque partie du projet serait bien.
* **Maquette basse-fidélité :** Puisque j’ai rencontré des difficultés en C# pour travailler avec WindForms, j’ai dû de choisir de travailler sur console, donc cela ne m’a pas laissé beaucoup de choix par rapport au design de ma maquette de basse-fidélité, alors j’ai fait ma maquette d’accord à comment j’ai visée travailler en C#.
* **Palette de couleur :** J’ai choisi ma palette de couleurs avec des couleurs vives, puisque l’écran de la console est déjà noir, donc il faut que le texte, vaisseau et autres types de composants soient visible, alors ce type de couleur me plaisait et je les ai appliquées sur ma maquette de haute-fidélité.
* **Maquette haute-fidélité :** Après d’avoir sélectionné ma palette de couleur, j’ai fait l’éditeur de niveau et utilisé ces couleurs. J’ai recherché un convertisseur de texte en ASCII pour afficher le titre et j’ai trouvé cool le style. Pour le texte, j’ai choisi un autre type de style plus sérieux, car je me suis dit que ce jeu, c’est une réplique du jeu « Space Invaders » alors, il ne faut pas ajouter trop de style dans ce type de jeu.
* **Personas :** Pour la création de mes personas, j’ai pensé à un ami qui était un ancien professeur de mathématiques au Pérou, car il parlait toujours en classe des jeux qu’il jouait à ses époques. Pour l’autre persona j’ai pensé plutôt à mon grand-père, car il va bientôt prendre sa retraite et j’ai entendu plein de fois qu’il disait que quand il prendra sa retraite allait acheter des comics, des jeux vieux comment la première Play Station ou console avec des manettes.

# partie poo

## Analyses fonctionnelles (Users Stories)

## Automatique du code (manuel de référence)

## Schémas des classes

## Implémentation spécifique

# PARTIE DB

## MCD

### Description des tables

### Description de

### Description de

## MLD

## Script SQL

## Base de données

* Au moins deux niveaux de jeu
* Deux joueurs correspondants à vos Personas UX
* Un joueur devra être présent par niveau
* Les informations concernant les ennemis et les joueurs devront correspondre exactement à ce qui sera implémenté dans votre jeu (ex: si vous aurez 10 ennemis, nous devons retrouvez ces 10 ennemis dans votre jeu de données. Idem pour les obstacles, etc.)

## Mise en place des types d’utilisateurs

### Explication

Vous devez expliquer, comment mettre en place ces différents « types d’utilisateurs » afin de rendre la gestion des utilisateurs la plus simple possible. Pour cela, vous devez définir un ensemble de requêtes SQL. Vous devez créer au moins un utilisateur par « type d’utilisateurs ».

# Rapport de test

## Tests POO

## Tests DB

# USAGE DE L’IA

### Explication

# REFERENCES

## Liens UX

* [Support de cours du module et annexes I322](https://eduvaud.sharepoint.com/:f:/r/sites/ETML_INF-GRP2D-24-25_Teams/Documents%20partages/I322-ACD/Theorie?csf=1&web=1&e=7hChne)
* [Cahier de charges de ce projet](https://eduvaud.sharepoint.com/:b:/r/sites/ETML_INF-GRP2D-24-25_Teams/Documents%20partages/I322-ACD/Shoot%20Me%20Up-CdC-UXV2.pdf?csf=1&web=1&e=hHQy7Z)
* [ChatGPT](https://chatgpt.com/)
* [Adote Color Palette](https://color.adobe.com/fr/create/color-wheel)
* [ASCII Art Archive](https://www.asciiart.eu/text-to-ascii-art)
* [115 Bonnes pratiques – Eco-Conception](https://collectif.greenit.fr/ecoconception-web/115-bonnes-pratiques-eco-conception_web.html)

## Liens DB