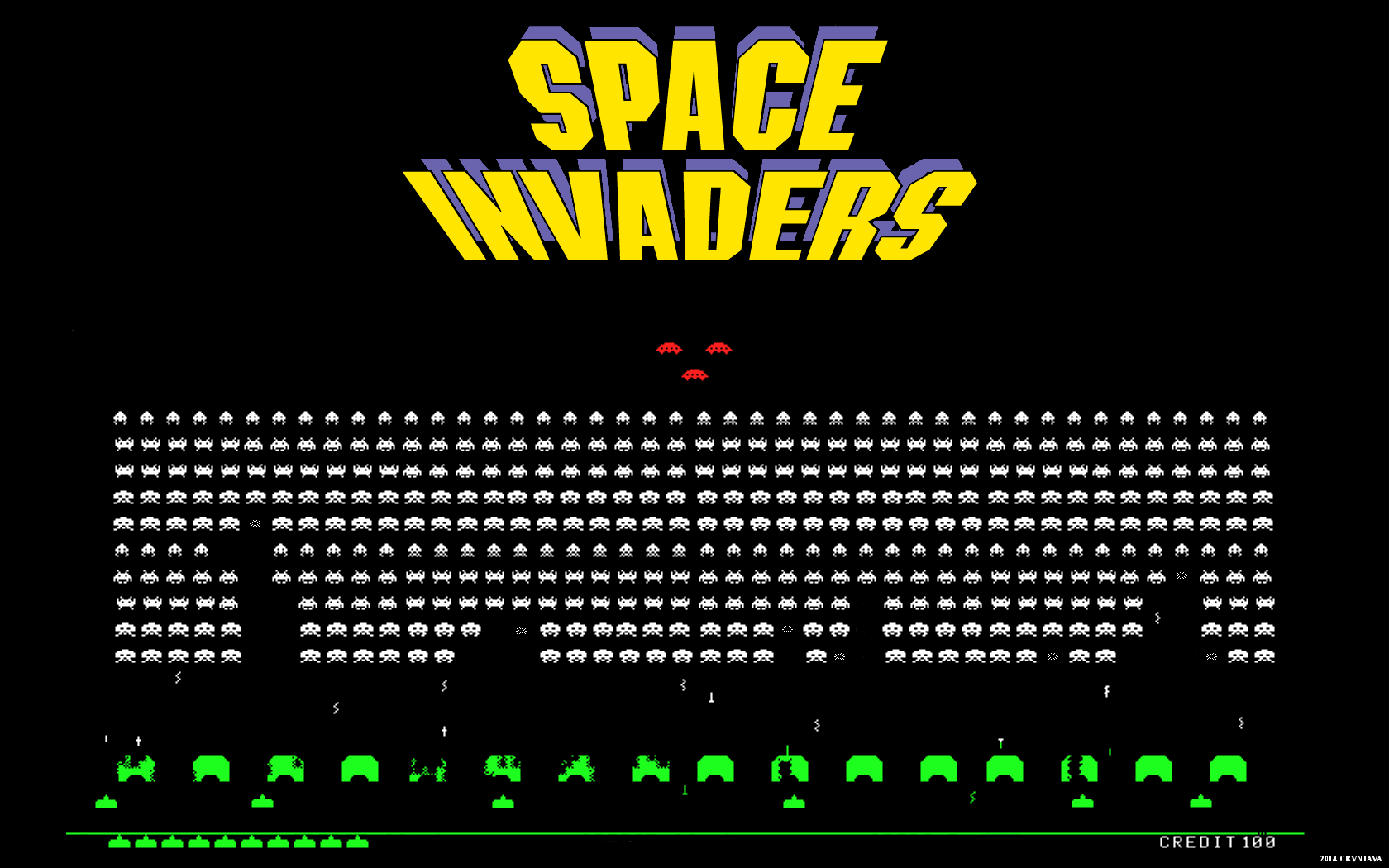
P\_ShootMeUp



Romain Denis – CID2B

Vennes

28.08.24 – 01.11.24

Antoine Mveng, Aurélie Curchod, Jonathan Melly

Table des matières

Table des matières

[1 Spécifications 3](#_Toc181369151)

[1.1 Titre 3](#_Toc181369152)

[1.2 Description 3](#_Toc181369153)

[1.3 Matériel et logiciels à disposition 4](#_Toc181369154)

[1.4 Prérequis 4](#_Toc181369155)

[1.5 Les points suivants seront évalués 4](#_Toc181369156)

[1.6 Validation et conditions de réussite 4](#_Toc181369157)

[2 Spécificités DB 4](#_Toc181369158)

[2.1 MCD 4](#_Toc181369159)

[2.2 MLD 5](#_Toc181369160)

[2.3 Script SQL de création de la base de données 5](#_Toc181369161)

[2.4 Un jeu de données spécifique à votre jeu contenant : 5](#_Toc181369162)

[ Au moins deux niveaux de jeu 5](#_Toc181369163)

[ Deux joueurs correspondants à vos personnas UX 5](#_Toc181369164)

[ Un joueur devra être présent par niveau 5](#_Toc181369165)

[ Les informations concernant les ennemis et les joueurs devront correspondre exactement à ce qui sera implémenté dans votre jeu 5](#_Toc181369166)

[2.5 Création des indexes 5](#_Toc181369167)

[En étudiant le dump MySQL de votre DB vous constaterez que vous ne trouvez pas le mot clé INDEX. 5](#_Toc181369168)

[1. Pourtant certains index existent déjà. Pourquoi ? 5](#_Toc181369169)

[2. Quels sont les avantages et les inconvénients des index ? 5](#_Toc181369170)

[**3.** Sur quel champ (de quelle table. Pensez à l'utilisation de votre DB depuis C# par exemple), cela pourrait être pertinent d’ajouter un index ? Justifier votre réponse. 6](#_Toc181369171)

[2.6 Backup / Restore 6](#_Toc181369172)

[2.7 Gestion de utilisateurs 6](#_Toc181369173)

[3 Spécificités POO 6](#_Toc181369174)

[3.1 Analyse fonctionnelle 6](#_Toc181369175)

[3.1.1 MenuForm.cs 6](#_Toc181369176)

[3.1.2 LevelSelect.cs 6](#_Toc181369177)

[3.1.3 Form1.cs 7](#_Toc181369178)

[3.1.4 HUD.cs 7](#_Toc181369179)

[3.1.5 Timers 7](#_Toc181369180)

[3.1.6 Movement 7](#_Toc181369181)

[3.1.7 End of game 7](#_Toc181369182)

[3.2 Tests unitaires 7](#_Toc181369183)

[3.3 Automatique du code (manuel de référence) 8](#_Toc181369184)

[3.4 Schémas des classes 8](#_Toc181369185)

[3.5 Au moins un détail d’implémentation spécifique 8](#_Toc181369186)

[3.6 Bugs connus 9](#_Toc181369187)

[4 Spécificités UX 9](#_Toc181369188)

[4.1 Analyse de l’UX : 9](#_Toc181369189)

[4.1.1 Conception centrée utilisateur 9](#_Toc181369190)

[4.1.2 Création de deux profils de joueurs sous forme de “Personas” 10](#_Toc181369191)

[4.1.3 Création d’au moins un « Personas » original  spécificité liée à l’accessibilité et profil qui ne correspond pas aux « classique » du joueur 10](#_Toc181369192)

[4.1.4 Choix de la palette graphique 10](#_Toc181369193)

[4.1.5 Eco-conception 10](#_Toc181369194)

[4.1.6 Accessibilité 10](#_Toc181369195)

[4.2 Conception 11](#_Toc181369196)

[4.2.1 Définition de tous les écrans - maquettes base-fidélité / low-fidelity wireframes 11](#_Toc181369197)

[4.2.2 Ecran “Éditeur de niveau” - maquette haute-fidélité / high-fidelity wireframe 12](#_Toc181369198)

[4.2.3 Ajout de fonctionnalités supplémentaires tirées des Personas un élément original 13](#_Toc181369199)

[4.2.4 Choix effectués 14](#_Toc181369200)

[5 Utilisation AI 15](#_Toc181369201)

[Au cours de ce projet, j'ai utilisé l'IA pour accélérer le processus de certaines tâches : il s'agissait de : 15](#_Toc181369202)

[UX : créer des personas car le fait de le faire faire par un humain n'a aucune valeur ajoutée. Je pense que c'est une utilisation équitable car cela accélère mon projet et me permet de faire plus de choses. 15](#_Toc181369203)

[Pendant POO, j'ai utilisé de l'IA pour clarifier le fonctionnement des choses, car mon projet est assez unique et je ne pouvais pas demander à mes camarades de classe. 15](#_Toc181369204)

[Je n'ai pas utilisé d'IA pendant la partie base de données de ce projet. 15](#_Toc181369205)

[6 Conclusion 15](#_Toc181369206)

[6.1 Bilan des fonctionnalités demandées 15](#_Toc181369207)

[6.2 Bilan personnel 15](#_Toc181369208)

[7 Livrables 16](#_Toc181369209)

[8 Divers 16](#_Toc181369210)

[8.1 Journal de travail 16](#_Toc181369211)

[9 Annexes 16](#_Toc181369212)

# Spécifications

## Titre

P\_ShootMeUp

## Description

Concevoir un jeu 2D modulaire de tir à la troisième personne et en réaliser une partie.

## Matériel et logiciels à disposition

* Un PC ETML
* Accès à Internet
* IceScrum (<https://etml.icescrum.com>)
* Figma (https://figma.com)

## 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

## Les points suivants seront évalués

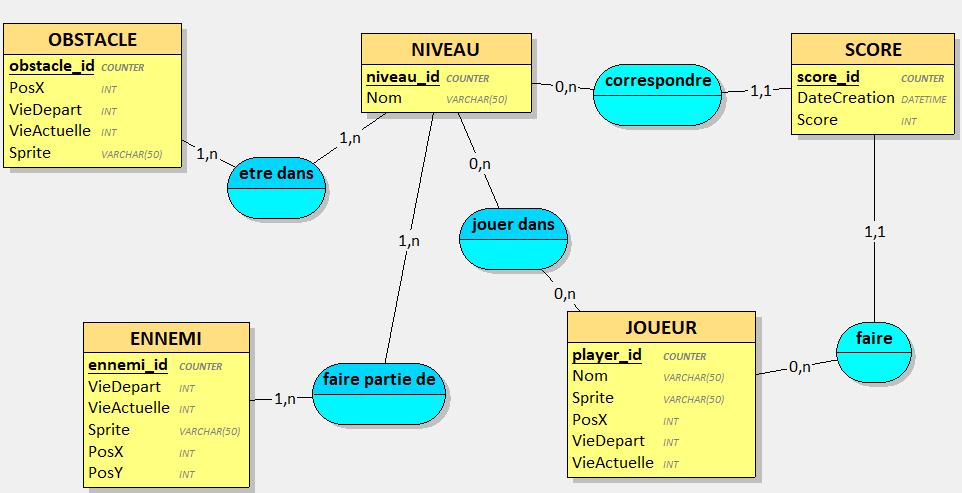
* Le rapport
* Les planifications (initiale et détaillée)
* Le journal de travail
* Le code et les commentaires
* Les documentations de mise en œuvre et d’utilisation

## Validation et conditions de réussite

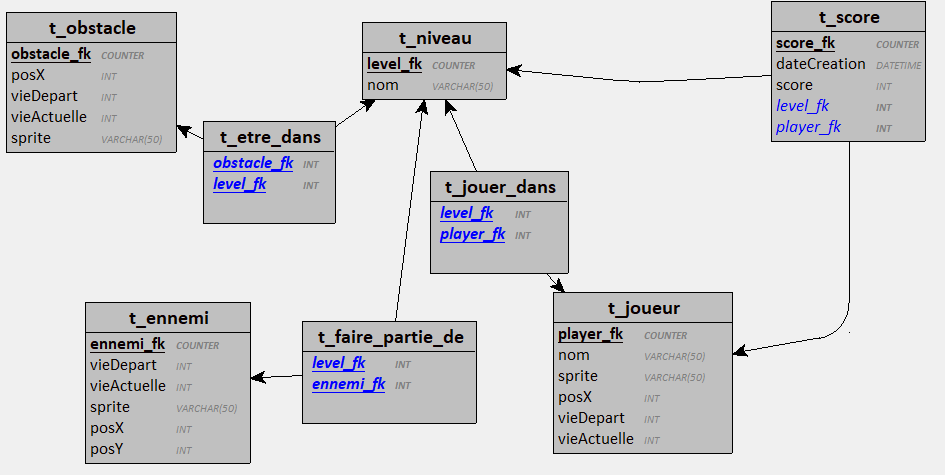
* Compréhension du travail
* Possibilité de transmettre le travail à une personne extérieure pour le terminer, le corriger ou le compléter
* Etat de fonctionnement du produit livré

# Spécificités DB

## MCD



## MLD



## Script SQL de création de la base de données



## Un jeu de données spécifique à votre jeu contenant :

### Au moins deux niveaux de jeu

### Deux joueurs correspondants à vos personnas 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

## Création des indexes

### En étudiant le dump MySQL de votre DB vous constaterez que vous ne trouvez pas le mot clé INDEX.

### Pourtant certains index existent déjà. Pourquoi ?

Les indexes existes déjà par default sur les clefs primaires, clefs uniques et les clefs étrangères. Ces index permettent à MySQL d’accélérer la recherche, la vérification d’unicité, et les relations entre les tables sans nécessiter de création manuelle.

### Quels sont les avantages et les inconvénients des index ?

Avantages **:**

1. Les index accélèrent la recherche et la récupération des données
2. Lorsqu'on utilise des index, MySQL peut réduire le nombre de lectures de disques et donc le temps d’exécution des requêtes.
3. Les index sur les clés étrangères facilitent les jointures entre les tables.

Désavantages :

1. Les index occupent de l’espace disque supplémentaire. Plus il y a d'index, plus l'espace de stockage requis augmente.
2. Ralentissement des opérations INSERT, UPDATE, et DELETE

### **3.** Sur quel champ (de quelle table. Pensez à l'utilisation de votre DB depuis C# par exemple), cela pourrait être pertinent d’ajouter un index ? Justifier votre réponse.

Je pense que c’est mieux de mètre un index sur la colonne player\_fk dans la table scores. Dans le cadre d'une application C# qui nécessite souvent d'afficher les classements ou l'historique des scores d'un joueur, ça serait judicieux d’améliorer les temps des requêtes. C’est la requête qui sera le plus utilise donc c’est évident de change celui-là.

## Backup / Restore

*Nous souhaitons réaliser une sauvegarde (Backup) de votre base de données. Ensuite, nous souhaitons nous assurer que cette sauvegarde est correcte en la rechargeant dans MySQL (opération de restauration). Donner la commande permettant de faire :*

Commande Backup: mysqldump -u [username] -p db\_spaceinvaders > db\_spaceinvaders\_backup.sql

Commande Restore: mysql -u [username] -p db\_spaceinvaders < db\_spaceinvaders\_backup.sql

## Gestion de utilisateurs

Voici les commandes que j’ai utilisé dans ma base de donnes :



# Spécificités POO

## Analyse fonctionnelle

### MenuForm.cs

In my code, the first thing you see after executing it is that a menu (menuForm.cs) pops up. If you click on quit, the menu closes and if you click on play the menu closes and a new window appears.

### LevelSelect.cs

This window is the levelselect (levelSelect.cs) screen, where you can choose to play level 1 2 or 3. when you end up choosing, the program transfers a "levelIndex" variable (originally in levelSettings.cs) to the form1.cs class, to tell it which level has been selected. This works because i have a list with all the settings per entry so the form1.cs knows which level has been selected. This changes speed of the enemy, requiredScore to win and the sprite of the Invader.cs class.

### Form1.cs

After that, the game can start. The form1.cs code first creates the ship window and sets it up to be the size of the spaceship.cs. After that it spawns a new wave of enemies by calling SpawnNewWave (); which is a for loop that instantciates new invaders.

### HUD.cs

After that, form1.cs instanciates a new HUD (with the number of lives and score) so that the player can track their progress during the game, and then starts all timers and calls the SpawnObstacles (), method. From here on out the game fully follows logic.

### Timers

In my timers i have four timers:

I have the movetimer which moves the invaders across the screen, which ticks as \_moveTimer.Tick += MoveInvaders; MoveInvaders (); is just a for loop that that adds to this.top foreach alive invader.

Next i have a collision timer that checks collisions between every object (obstacles, the ship, bullets, enemy bullets). In this one the bullets that are detected as a hit are added to a toremove list which then clears everything that has hit something of the screen. It does this for everything aside from the ship, which instead runs a DecreaseLives (); function that removes a life and updates the HUD.

Next, I have evilBulletShootTimer, that just shoots an evil bullet with the shootEvilBullet (); that shoots an evil bullet.

### Movement

I also have a protected override for the movement of the ship in form1.cs. This override moves the Ship left and right, and pressing space starts a timer called shootTimer, then calls the ShootBullet (); so that there is a cooldown to each shot.

### End of game

In the checkCollisions (); method there is also a check for that checks the number of lives the player has, if the player has no more lives, it calls GameOverText.cs with the message “Game Over!” in the constructor. The class then uses the message to create a new window with the message. When the message window is closed the menu opens again. The same thing happens if you kill all enemies at the same time or reach a certain number, with the message you win.

## Tests unitaires

|  |  |  |
| --- | --- | --- |
| Function | When | Test |
| MenuForm.cs | When the game starts | Does the menu open ? |
| LevelSelect.cs | When a level is selected | Do the settings correspond ? |
| Form1.cs | After a level is selected | Do all the windows open ? |
| HUD.cs | When the score increases / when a life is reduced | Does the HUD update |
| Timers | When a timer is instanciated | Do the enemies move ? |
| Movement | When a movement key is clicked | Is a bullet shot ? |
| End of Game | When the game is finished | Can we restart the game? |

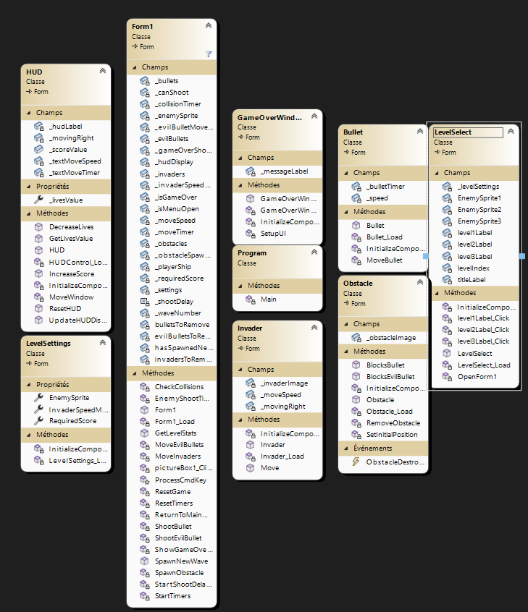
## Reference Manual

Here is the dockx file:



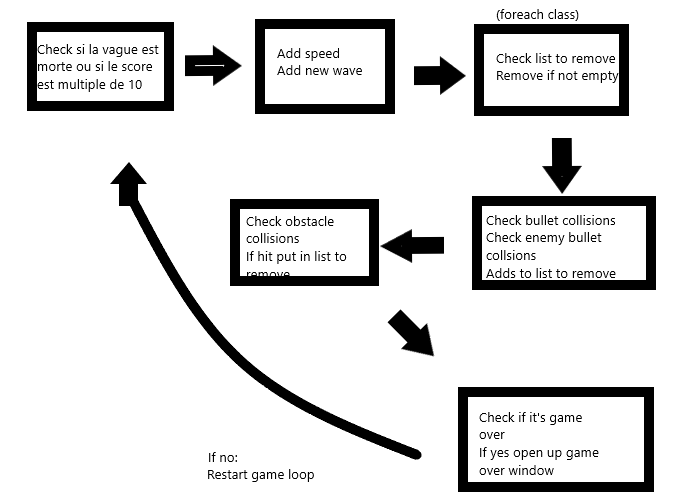
## Scheme of classes

Here is my scheme of classes :



## Implementation

Here is my main game loop:



## Known bugs

Collision Bug :

In the game it’s possible that bullets hit twice. This applies to both the player’s bullets and the enemy’s bullets. This is due to inconsistent timers and FPS. I cannot realistically fix the bug as I would have to change the entire collision system.

Bug projet C#

Dans le projet C#, si on ouvre la class MenuForm.cs, il y aura une erreur. Ça a quelque chose a voire avec le code qui n’est pas dans le designer, mais je n’ai pas le temps de la fixer. Pour ouvrir le code de la class il faut clicker sur **accéder au code**, et **pas ignorer et continuer**.

In the C# project, if we open MenuForm.cs, there is an error. It is an error that has something to do with the code not being in the designer. To open the code of the class you must click on **access code**, and not click **ignore and continue**.

# Spécificités UX

## Analyse de l’UX :

### Conception centrée utilisateur



Mon premier Persona, Steven change beaucoup de choses dans mon programme. C'est la raison pour laquelle j'ai un écran de score élevé, car il est extrêmement compétitif et adore améliorer son score. J'ai également ajouté différents niveaux à cause de lui, car je pense qu'il s'ennuierait s'il n'y avait qu'un seul niveau.

Ma deuxième personas, Maya change aussi beaucoup, c'est dû à sa mauvaise vue. J'ai donc ajouté un mode pour les personnes ayant des problèmes oculaires. Cela rend le menu beaucoup plus grand et plus audacieux pour qu'ils puissent mieux voir. Le gameplay lui-même est également codé par couleur pour les aider. Maya est aussi en partie la raison pour laquelle j'ai des niveaux, car elle préfère s'amuser dans les niveaux les plus faciles plutôt que d'essayer d'augmenter son score ou sa difficulté

### Création de deux profils de joueurs sous forme de “Personas”



### Création d’au moins un « Personas » original  spécificité liée à l’accessibilité et profil qui ne correspond pas aux « classique » du joueur

Ma deuxième persona, maya est malvoyante.

### Choix de la palette graphique



Voici la palette que j’ai choisi. J’ai choisi de prendre une palette avec des couleurs simples qui vont bien avec le thème simple du jeu. J’a choisi cette couleur de bleu par ce qu’elle me rappelait de l’intérieur d’une fusée

### Eco-conception

*L'éco-conception est une méthode de design axée sur la durabilité environnementale, visant à minimiser l'impact écologique des produits et services numériques tout en optimisant l'expérience utilisateur. Elle intègre des critères de convivialité et de performance écologique lors du développement de produits numériques, comme des sites web, applications mobiles et logiciels. Toutefois, pour un projet de design en jeu vidéo, l'éco-conception n'est pas réellement applicable.*

### Accessibilité

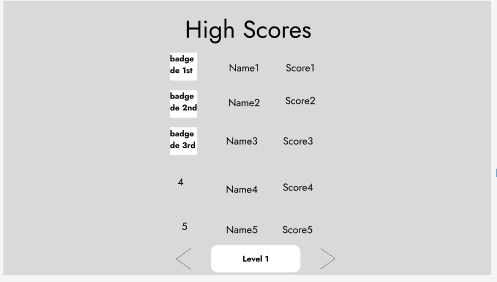
J’ai ajouté un mode malvoyant a ma maquette pour les personnes qui sont partiellement aveugles. Ce mode change l’UI et rend les boutons plus lisibles. (Voire maquette)

Le mode peut être activé depuis le menu principal, en cochant le bouton mode malvoyant en haut à gauche.

## Conception

### Définition de tous les écrans - maquettes base-fidélité / low-fidelity wireframes

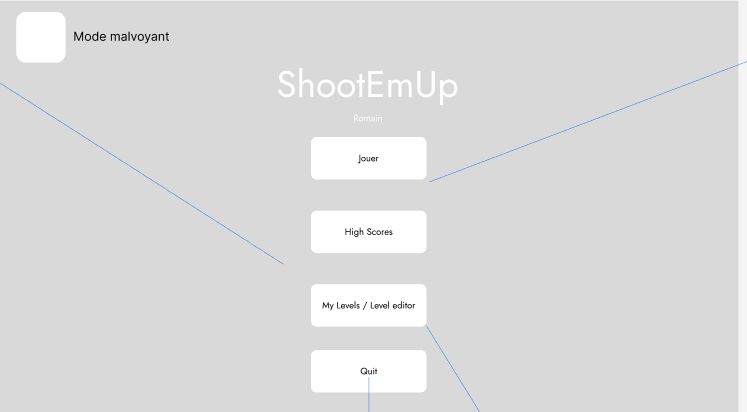
*Fenêtre high score*

**

*Fenêtre de jeu*

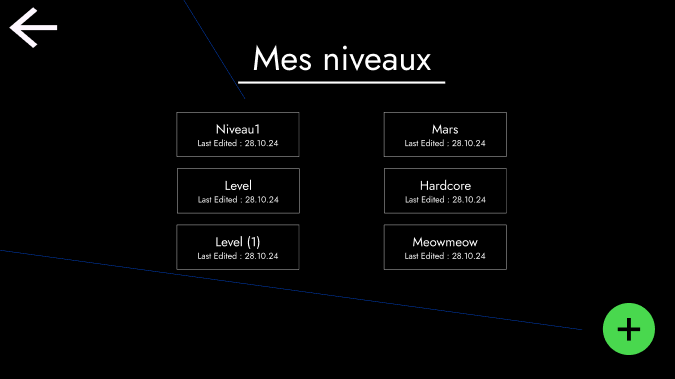
**

*Menu principal*

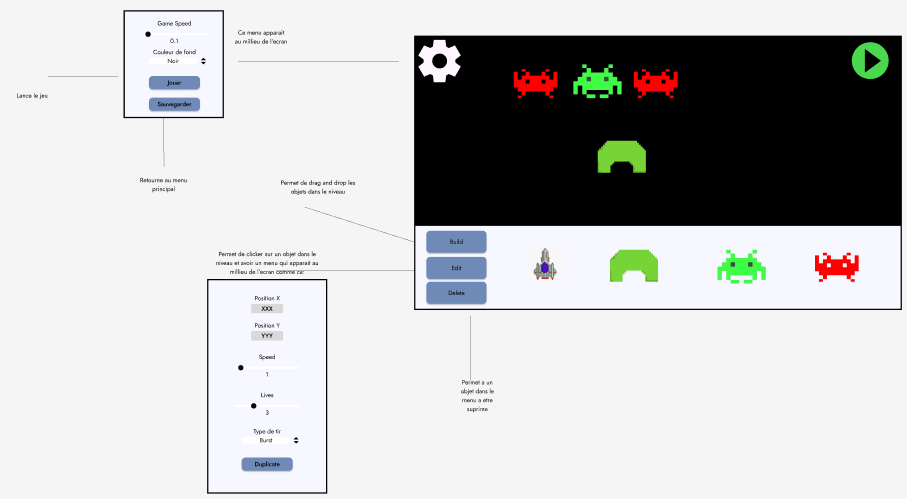
**

### Ecran “Éditeur de niveau” - maquette haute-fidélité / high-fidelity wireframe

*Ecran sélection de niveau*



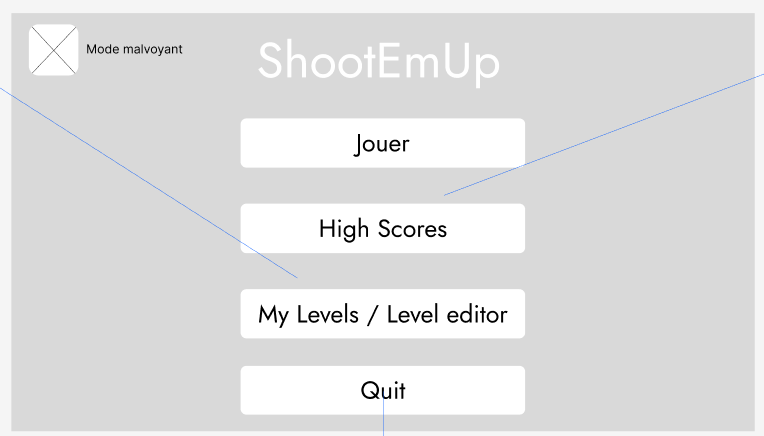
*Ecran d’éditeur de niveau*



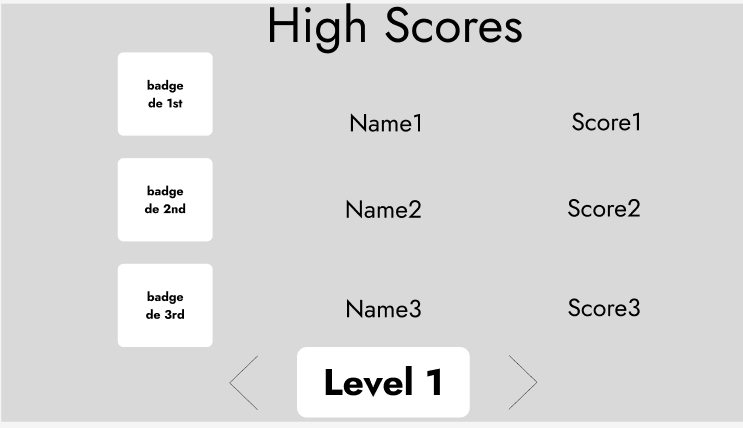
### Ajout de fonctionnalités supplémentaires tirées des Personas un élément original

Ma persona est malvoyante, donc j’ai fait des maquettes en mode malvoyant

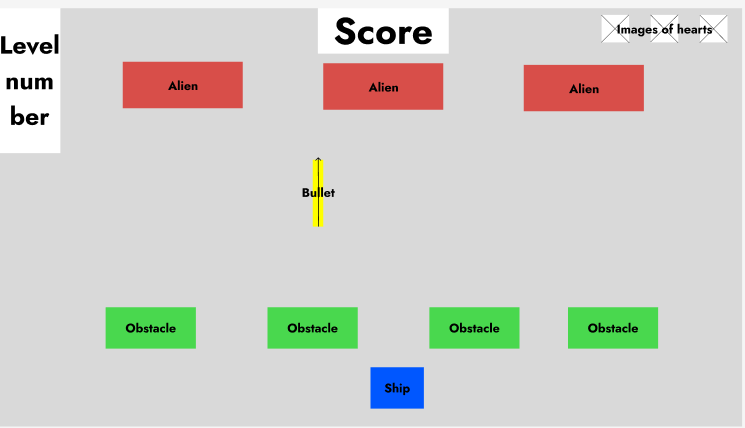
Menu principal



Fenêtre High Scores



Fenêtre de jeu



*(Le jeu est color codé)*

### Choix effectués

Dans le menu principal, j’ai choisi de faire des petits boutons car j’avais envie de montrer l’atmosphère du jeu avec un grand fond d’écran qui prend l’attention du joueur. J’ai aussi mis le bouton de mode malvoyant en grand en haut a gauche. C’est par ce que la première chose que le joueur verra est ce bouton par ce que c’est hyper important de jouer avec le bon mode. Je l’ai aussi fait plus grand pour mieux le voire.

Dans la page des high scores, j’ai choisi de mettre les scores en format «tableau», j’ai fait ceci par ce que j’aime quand les choses sont alignées. J’ai aussi mis un bouton en bas pour changer de niveau, car avoir tous les scores mélangés serait pas très utilisables.

Dans la page de jeu, j’ai choisi de mettre des images de cœurs pour représenter les vies. J’ai fait ceci car je pense que ça reste avec le thème retro, et j’imaginais des cœurs pixelisées, comme dans Minecraft. J’ai aussi choisi de mettre le nombre du niveau, pour clarté.

Dans ma page de niveaux, j’ai choisi de pouvoir jouer directement à des niveaux crées, j’ai fait ça par ce que je pense que ça aurait été énervant de devoir rentrer dans l’éditeur niveau pour clicker sur jouer.

Dans ma page éditeur de niveau, j’ai décidé d’utiliser des menus pop-ups, pour garder une simplicité au menu de base pour ne pas faire peur aux utilisateurs. J’ai aussi préféré faire un système de drag and drop pour la même raison.

# Utilisation AI

# Au cours de ce projet, j'ai utilisé l'IA pour accélérer le processus de certaines tâches : il s'agissait de :

# UX : créer des personas car le fait de le faire faire par un humain n'a aucune valeur ajoutée. Je pense que c'est une utilisation équitable car cela accélère mon projet et me permet de faire plus de choses.

# Pendant POO, j'ai utilisé de l'IA pour clarifier le fonctionnement des choses, car mon projet est assez unique et je ne pouvais pas demander à mes camarades de classe.

# Je n'ai pas utilisé d'IA pendant la partie base de données de ce projet.

# Conclusion

## Bilan des fonctionnalités demandées

Je pense bien avoir réussi les 3 projets.

En POO, je pense que j’ai fait bien plus que demandé.

En DB, je pense que j’ai tout fait à 100%.

En UX, je pense que j’aurais pu faire plus de maquettes hautes fidélité mais j’ai fait tout le CdC.

## Bilan personnel

Si je devais refaire ce projet, j’aurais plus poussé la partie UX. Je pense que j’aurais pu faire un design plus ambitieux, et faire plus de maquettes haute fidélités. Je n’ai pas pris assez de risques pendant ce projet.

En POO, j’aurais bien aimé intégrer la base de donnes dans mon programme. Je pense que j’aurais appris beaucoup de choses utiles. Je pense que j’aurais pu rendre mon code plus efficace aussi.

Jai bien aime la partie DB et je n’ai aucuns regrets par ce que je pense que j’ai tout fait au max. Même si j’aurais aimé implémenter la base de donne dans mon code C#.

En conclusion, j'ai vraiment apprécié ce projet. Je pense que c'était une représentation réaliste de ce que le monde réel demandera. La seule chose que je n'ai pas aimée était la clarté du projet. Je pense qu'il serait préférable de le développer davantage avant l'année prochaine. Ce que nous devions faire et comment chaque module interagissait entre eux n'était pas clair.

# Livrables

 1 release sur GitHub contenant :

a. README à la racine contient votre Nom, prénom et classe

b. Une archive ZIP avec les éléments UX

c. Une archive ZIP avec les éléments DB

d. Une archive ZIP avec les éléments POO

e. [Code source inclus automatiquement]

f. Journal de travail + Rapport

# Divers

## Journal de travail

Le journal de travail peut être trouve dans le repository dans SHOOTMEUP\Documention\

# Annexes

Repo GitHub: <https://github.com/romaindenis1/SHOOTMEUP>

Personas: <https://www.figma.com/design/CHN5kP4u6NHTtOFWdjPSxG/Free-User-Personas-(Community)?t=cpWfUPjqWxzQGqqj-0>

Maquette Figma: <https://www.figma.com/proto/gik7GHelnRbdcryRW7borb/Maquette-Projet-UX?t=cpWfUPjqWxzQGqqj-1>