Shoot Me Up



Antoine Fabre – CID2B

ETML-Vennes

Durée

Mveng - Melly - Curchod

Table des matières

[1 Spécifications 3](#_Toc308526316)

[1.1 Titre 3](#_Toc308526317)

[1.2 Description 3](#_Toc308526318)

[1.3 Matériel et logiciels à disposition 3](#_Toc308526319)

[1.4 Prérequis 3](#_Toc308526320)

[1.5 Cahier des charges 3](#_Toc308526321)

[1.5.1 Objectifs et portée du projet (objectifs SMART) 3](#_Toc308526322)

[1.5.2 Caractéristiques des utilisateurs et impacts 3](#_Toc308526323)

[1.5.3 Fonctionnalités requises (du point de vue de l’utilisateur) 3](#_Toc308526324)

[1.5.4 Contraintes 3](#_Toc308526325)

[1.5.5 Travail à réaliser par l'apprenti 4](#_Toc308526326)

[1.5.6 Si le temps le permet … 4](#_Toc308526327)

[1.5.7 Méthodes de validation des solutions 4](#_Toc308526328)

[1.6 Les points suivants seront évalués 4](#_Toc308526329)

[1.7 Validation et conditions de réussite 4](#_Toc308526330)

[2 Planification Initiale 4](#_Toc308526331)

[3 Analyse 4](#_Toc308526332)

[3.1 Opportunités 4](#_Toc308526333)

[3.2 Document d’analyse et conception 4](#_Toc308526334)

[3.3 Conception des tests 5](#_Toc308526335)

[3.4 Planification détaillée 5](#_Toc308526336)

[4 Réalisation 5](#_Toc308526337)

[4.1 Dossier de Réalisation 5](#_Toc308526338)

[4.2 Modifications 5](#_Toc308526339)

[5 Tests 5](#_Toc308526340)

[5.1 Dossier des tests 5](#_Toc308526341)

[6 Conclusion 5](#_Toc308526342)

[6.1 Bilan des fonctionnalités demandées 5](#_Toc308526343)

[6.2 Bilan de la planification 5](#_Toc308526344)

[6.3 Bilan personnel 5](#_Toc308526345)

[7 Divers 6](#_Toc308526346)

[7.1 Journal de travail 6](#_Toc308526347)

[7.2 Bibliographie 6](#_Toc308526348)

[7.3 Webographie 6](#_Toc308526349)

[8 Annexes 6](#_Toc308526350)

# Spécifications

## Titre

***Shoot me up !***

## Description

Ce projet a pour but de mettre en pratique les connaissances acquises dans les cours de programmation orienté objet, bases de données et expérience utilisateur.

## Matériel et logiciels à disposition

Un ordinateur avec Windows 10,

Figma,

Looping et

Visual studio 2022

## Prérequis

Pour réaliser ce projet correctement il faut avoir les connaissances du cours I322, I106 et I320.

A compléter par une description des compétences, des connaissances et de la formation minimum pour être à même de réaliser le projet …

## Cahier des charges

### Gestion de projet

1. La planification est à faire selon les instructions spécifiques de votre chef de projet.
2. Un journal de travail devra être rendu. L’outil que vous utilisez est libre, mais les caractéristiques suivantes doivent être respectées :
   * La structure et la présentation sont claires et soignées.
   * Les sources, les fichiers, les répertoires, les commits, et autres sources d'informations concernées par le journal sont référencés.
   * L'état et les durées des tâches mentionnées sont précisés.
   * Toutes les activités planifiées, les aides extérieures, ainsi que les imprévus et les heures supplémentaires y sont mentionnés.
   * Les succès et les échecs sont mentionnés.
   * Le travail journalier et son appréciation critique, ainsi que les réflexions y figurent.

### Qualité

1. Réaliser un programme informatique de qualité
   * Organisé (namespace, classes, commit log,…)
   * Compacté (pas de copié/collé,…)
   * Optimisé (utilisation de structures adaptées)
   * Testé (tests unitaires)
   * Commenté
   * Complet (code, script DB, maquettes PDF, éxécutable, …)
2. Prouver que vous êtes digne de confiance lorsqu’on vous confie un projet
   * Journal de travail à jour
   * Pro-activité
     + **Poser des questions** au client
     + Faire des démonstrations
     + Utiliser un système de versioning de code (GIT)

### Fonctionnalités requises (du point de vue client)

### Replica d’un « shoot’em up 2D », comme Space Invader

* 1. Maquettes
     1. Menu principal
     2. Ecran de jeu (niveau)
     3. Éditeur de niveau (voir détails ci-dessous)
     4. High scores
  2. Contraintes de réalisation
     1. Un concept de niveaux décrivant
        1. Le numéro du niveau (Level 1, Level 2, …)
        2. Le joueur
           1. Déplacements
           2. Nombre de vies
           3. Capacités de tir : direction, rafale, cooldown, décompte munitions, recharge, …
           4. Un sprite
        3. Les ennemis du niveau avec (pour chaque type)
           1. Nombre de vies
           2. Minutage d’apparition
           3. Tir (oui / non)
           4. Un sprite
        4. Les obstacles avec (pour chaque type)
           1. Une taille
           2. Une position X,Y
           3. Un sprite
           4. Le comportement en cas de dégâts (tir, collision)
     2. Structure et données des niveaux décrits et stockés dans une base de données relationnelle
  3. Fonctionnalités
     1. Au moins 2 niveaux implémentés avec
        1. Joueur
        2. Ennemis
        3. Obstacles
     2. Gestion des highscores (en base de données)

### Spécificités UX

La documentation contenue dans livraison finale du projet (Github) comprend :

* Un chapitre d’analyse de l’UX:
  + Conception centrée utilisateur
    - Création de deux profils de joueurs sous forme de “Personas”
  + Choix de la palette graphique
  + Eco-conception
  + Accessibilité
* Un chapitre de conception
  + Définition de tous les écrans - maquettes base-fidélité / low-fidelity wireframes
  + Ecran “Éditeur de niveau” - maquette haute-fidélité / high-fidelity wireframe
  + Choix effectués
* Un chapitre d’évaluation
  + Tests (A/B tests, test d’utilisabilité)

### Spécificités POO

La livraison finale du projet (Github) comprend :

1. Le code
   * Programmation orientée objet
   * Tests unitaires
2. Documentation (chapitres du rapport)
   * Une analyse fonctionnelle centrée utilisateur sous forme de User Stories
   * Automatique du code (manuel de référence)
   * Schémas des classes
   * Au moins un détail d’implémentation spécifique

### Spécificités DB

1. MCD
2. MLD
3. Script SQL de création

## 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

# Planification Initiale

m-planification-jnltrav1

# Analyse

## Opportunités

Afin de réussir ce projet il faut mettre en œuvre les connaissances des cours

I322 (Concevoir et Implémenter des interfaces utilisateur), I320 (Programmation Orienté Objet) et I106 (Interroger, traiter et assurer la maintenance des bases de données).

Le matériel utilisé pour mener à bien ce projet sont

L'application web Figma

Le logiciel de modélisation de donnée Looping

Le logiciel de programmation Visual Studio

Ce paragraphe énumère la liste des difficultés potentielles de tout ordre :

Liste des compétences à acquérir ou approfondir

Liste du matériel à exploiter

Recherche d’informations particulières

Gestion du travail en équipe & collaboration

Ainsi que les solutions possibles

Si les spécifications de départ ne laissent pas de doutes sur la manière de réaliser un projet, ce chapitre ne fera que renvoyer le lecteur aux spécifications.

## POO

### Analyse fonctionnelle

En lançant ce programme la fenêtre du menu principale va se lancer en premier,

Sur cette fenêtre il y a un bouton quitter en haut à gauche qui va fermer l’application en entier si on clique dessus. Ensuite il y a un bouton jouer en bas au milieu qui va ouvrir la fenêtre de jeu et cacher la fenêtre du menu principale.

Quand la fenêtre de jeu se lance elle est en pleine écran et le jeu commence tout de suite. Le joueur joue un vaisseau spatial qui apparait en bas de l’écran.

Le vaisseau tir automatiquement, c’est un timer qui va dire tous les combiens de temps le vaisseau peut tirer donc dans le timer on va instancier l’objet balle avec les coordonnées du vaisseau comme ça la balle va apparaitre sous le vaisseau. Ensuite pour déplacer la balle on va diminuer la valeur Y de sa location grâce à la méthode MoveBullet qui va être utilisé dans le timer. Pour faire bouger toutes les balles on utilise une boucle foreach qui va chercher toutes les balles dans liste bullets et les faire avancer.

Pour faire bouger le vaisseau on utilise une méthode Form1\_KeyDown qui va détecter les touches presser par le clavier, on va retenir seulement les touches W, A, S, D et E. Les touches W, A, S et D sont des touches de mouvement. Cela veut dire que si on appuie dessus on va changer la Picture location x et y. ensuite il y a la touche E, quand on appuie dessus la capacité spéciale du joueur (ultimate) va se lancer. Après quand la grosse balle va se déplacer ça sera le même principe que pour le tir de base. Mais avant de tirer cette balle on va vérifier que le cooldown de cette capacité est à zéro sinon ça ne va pas tirer.

Après il y a les ennemis, ce sont des objets qui utilise la méthode Move qui va être rythmé avec le timer “MoveEnnemi”. Mais dons cette méthode on ne va pas juste déplacer l’ennemi, quand l’ennemi va toucher ou depacer le pixel 1920 de la fenêtre moins sa largeur, l’ennemi va descendre de quelque pixel et ensuite repartir dans l’autre sens. Il se passe la même chose de l’autre cote. Ensuite pour créer des vagues d’ennemi on va instancier d’abord dans une méthode tous les ennemis et ensuite dans une autre méthode si le nombre d’ennemi est à zéro et que le nombre de la vague est inferieur a 5 ça appelle la méthode qui créer les ennemis mais les ennemis seront plus rapides.

La méthode “CheckCollisionsBullet” débute avec une boucle for qui se répète en fonction du nombre de balle sur le terrain, ensuite toutes les images des balles vont être considéré comme un rectangle et la même chose avec les ennemis.

Après si le rectangle de la balle touche le rectangle de l’ennemi on va enlever du jeu cet ennemi et cette balle. On va ajouter de 1 le score, on va mettre à jour le score et ajouter un point d’expérience. Cette méthode va être appelé dans le timer 1. Ce timer est le plus rapide donc il n’y aura pas de moment où la balle chevauchera l’ennemi. Il y a aussi les balles que les ennemis tir c’est le même principe que les balles que le joueur tir juste au lieu d’apparaitre sous le vaisseau, elles apparaissent en haut de la fenêtre et toujours au-dessus du vaisseau. Le système de collision reste le même. Cependant le système de collision de la capacité spécial avec les ennemis est un peu différent car quand la balle va toucher un ennemi la balle ne va disparaitre car c’est un tir puissant.

Quand on tue un ennemi on récupère un point d’expérience. La méthode LevelUP va voir si la variable des points d’expérience est égale à six, si c’est le cas ça va ajouter un niveau en plus, on reset les points d’expérience et des boutons au-dessus des deux habilités vont s’afficher avec marquer dessus améliorer. Quand on va appuyer sur un des boutons, par exemple celui au-dessus de l’attaque de base cela va améliorer l’attaque de base en tirant plus vite et aussi les deux boutons vont disparaitre. Donc ça a diminué le temps du timer qui gère la création des balles.

Quand le joueur va perdre pour une quel qu'onques raison la fenêtre de jeu va se cacher et la fenêtre de fin de jeu va s’ouvrir. Il va être écrit sur un label en haut Game Over et en dessous la cause de la mort sera écrite aussi dans un label. Deux boutons seront disposés en dessous avec écrit dessus rejouer et continuer. Si on appuie sur rejouer cette fenêtre va se fermer et la fenêtre du jeu s’ouvre.

Si on appuie sur le bouton continuer cette fenêtre va se fermer et la fenêtre du menu principale s’ouvre.

Quand le joueur va gagner en tuant tous les ennemis de toutes les vagues, cette fenêtre se ferme, la même fenêtre que si on perd s’ouvre mais a la place de Game over il y a écrit Game Win et en dessous des félicitations sont écrite.

Les boutons sont les mêmes que quand on perd.

Ce paragraphe décrit le fonctionnement de manière détaillée.

Autant que possible de manière graphique, imagée, tableaux, etc.

Tous les cas particuliers devraient y être spécifiés…

Il s’agit d’y présenter les fonctionnalités à développer :

Découpage en étapes, en modules, en fonctionnalités, etc.

Formulaires, interfaces graphiques, pages web, etc.

Schémas de navigation, schémas événementiels, structogramme, pseudocode, etc.

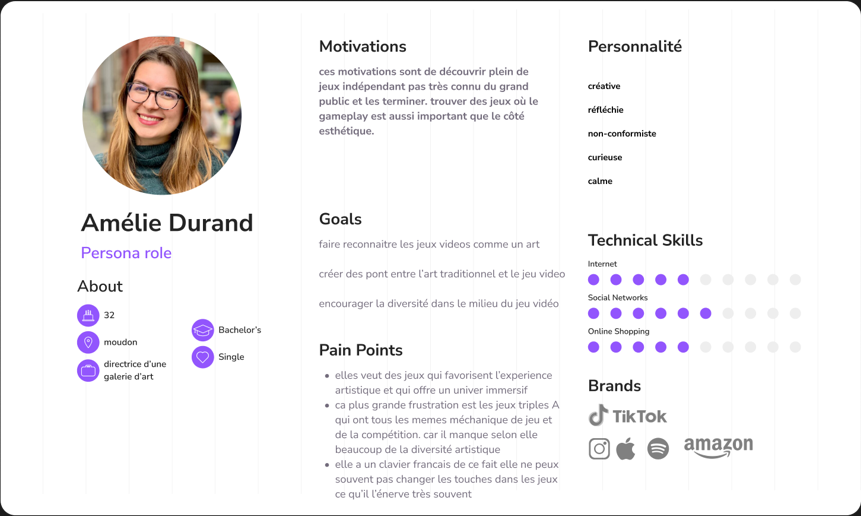
Si le projet inclut une base de données :

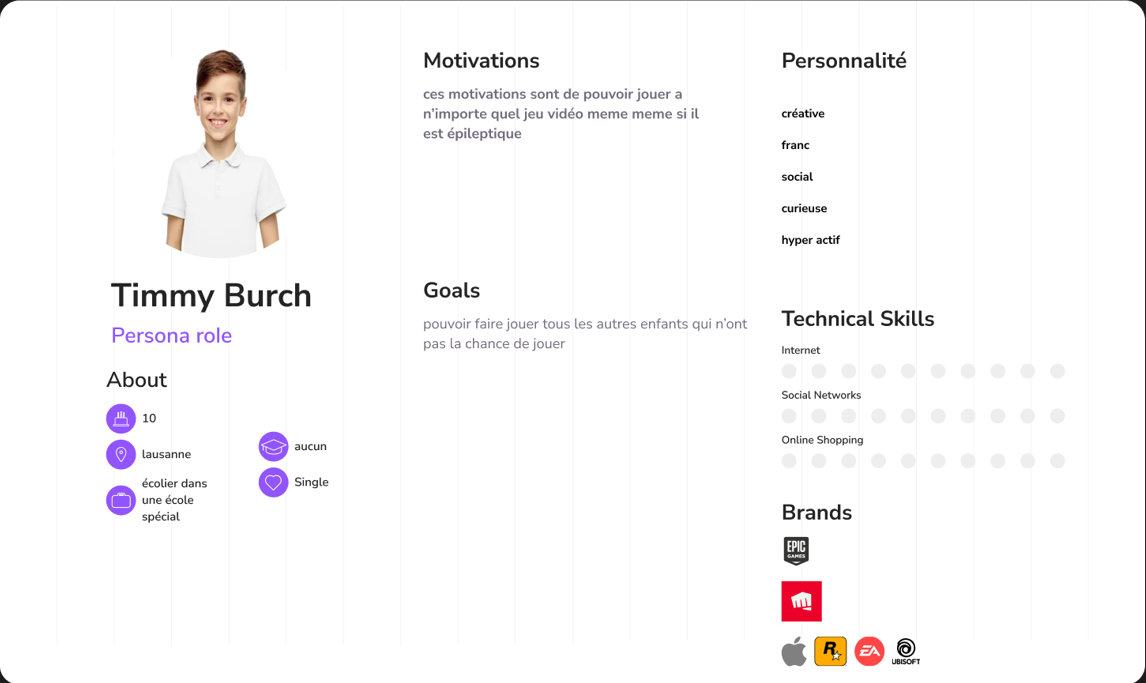
Dictionnaire des données

Modèle conceptuel des données, modèles logique des données.

## UX

### Conception centrée utilisateur



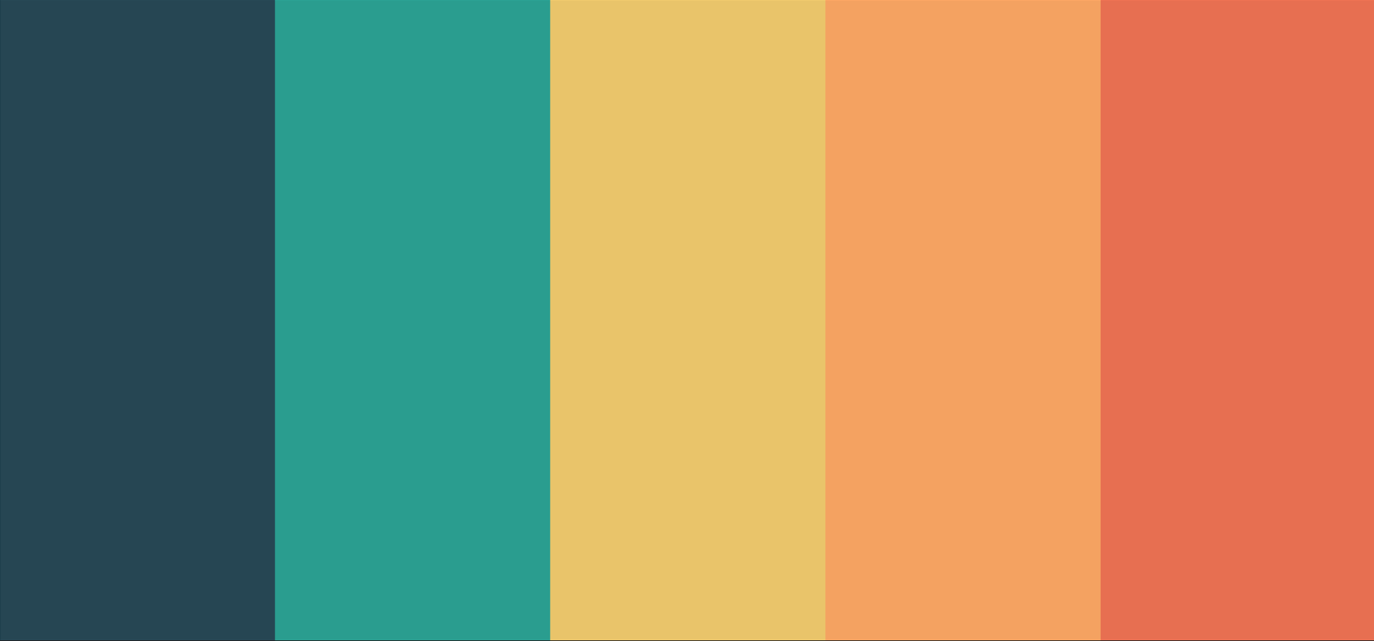


Voici les deux personas, en premier on a Amélie Durant une grande fan du côté artistique dans les jeux. Mais sa plus grande peine c’est d’avoir un clavier français, dans énormément de jeu on ne peut changer les touches ce qu’il l’énerve beaucoup. En remédiant a son problème j’ai tout de suite pensé à un endroit dans les paramètres ou on peut changer les touches de jeu. 

Ensuite il y a le petit Timmy butch qui est épileptique mais adore les jeux,

Les medecins ne conceil vraiment pas de jouer au jeu pour éviter au plus les crises d’épilepsie. Cependant il y a une option a cocher pour activer le mode épilepsie. Ce mode va desactiver toutes les animation flash du jeu et beaucoup ralentir le jeu ce qui peut grandement éviter des crises d’épilepsie a des personnes épileptique.

### Palette graphique



Voici la palette graphique de cette application.

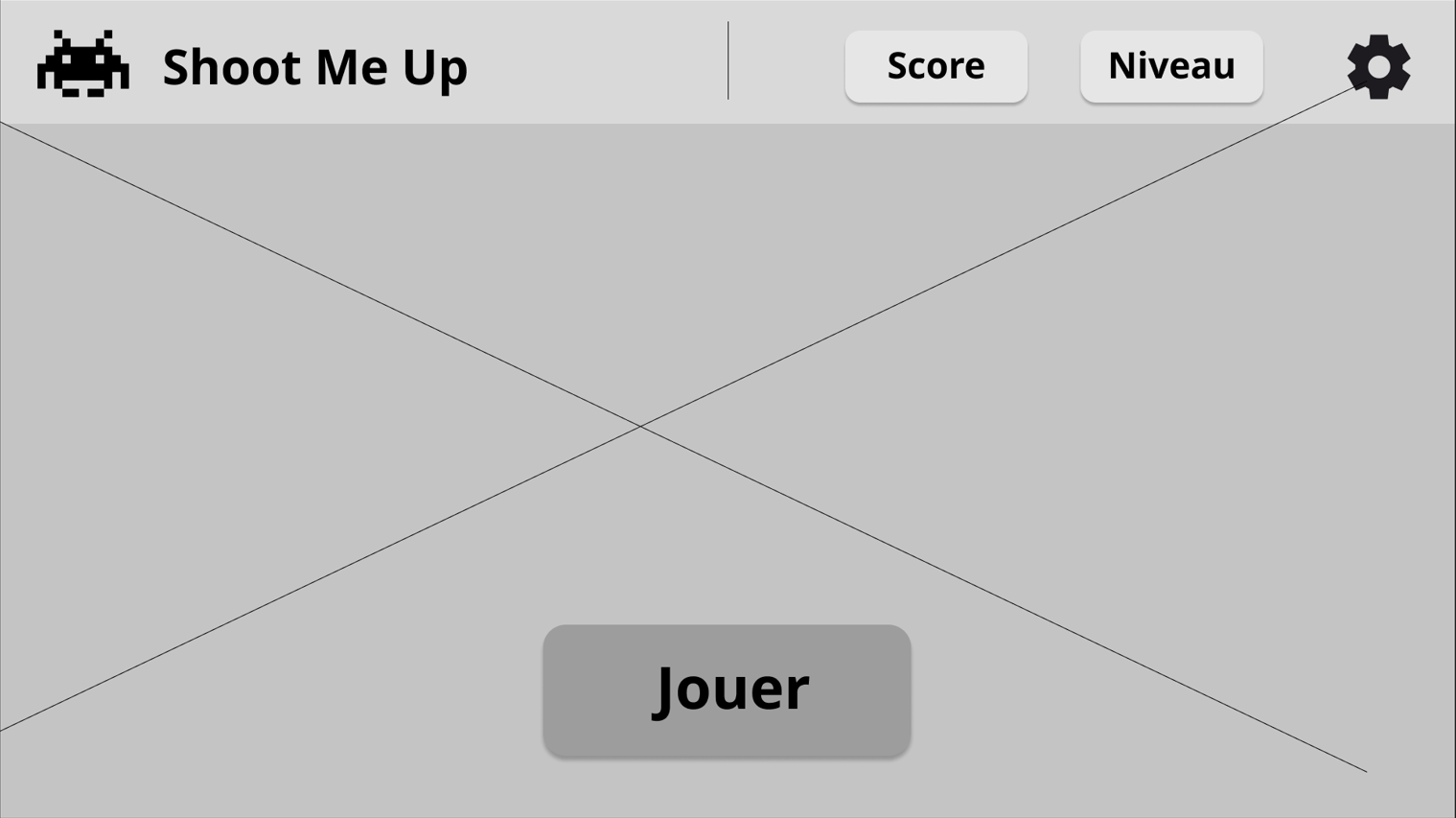
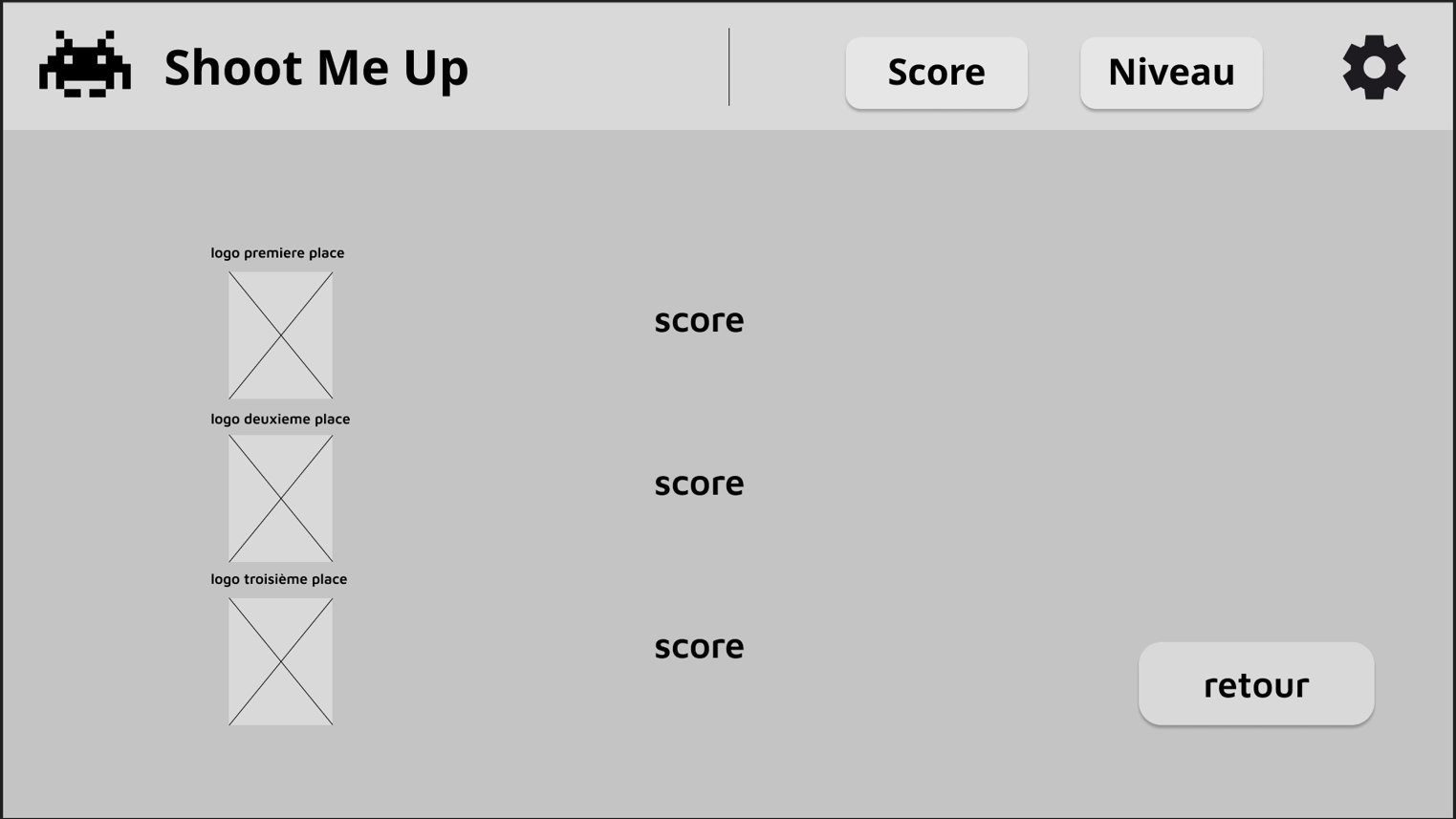
J’ai choisi cette palette car l’ensemble de ces couleurs faisait un style plutôt retro futuriste ce qui est parfait pour ce jeu.

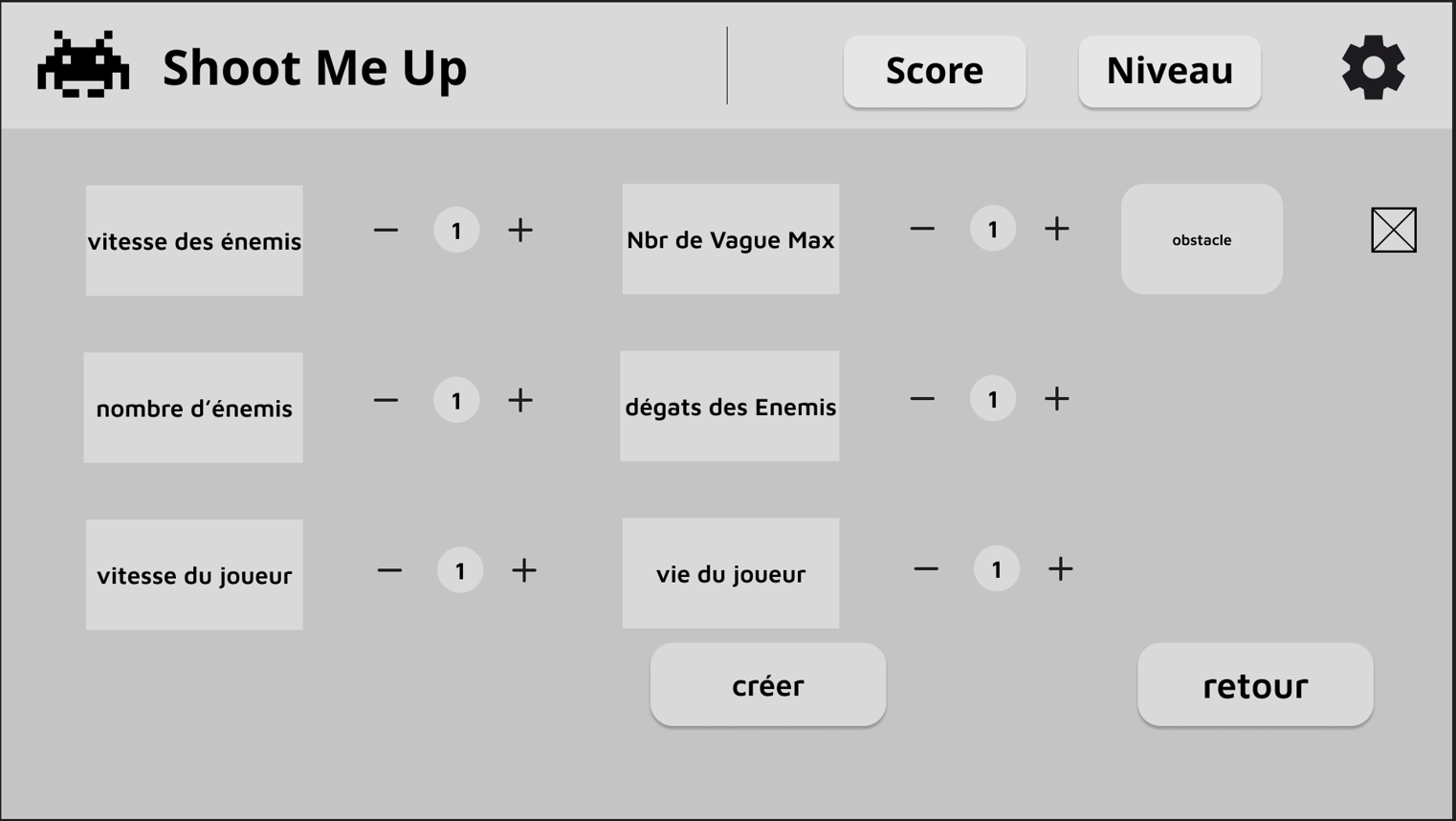
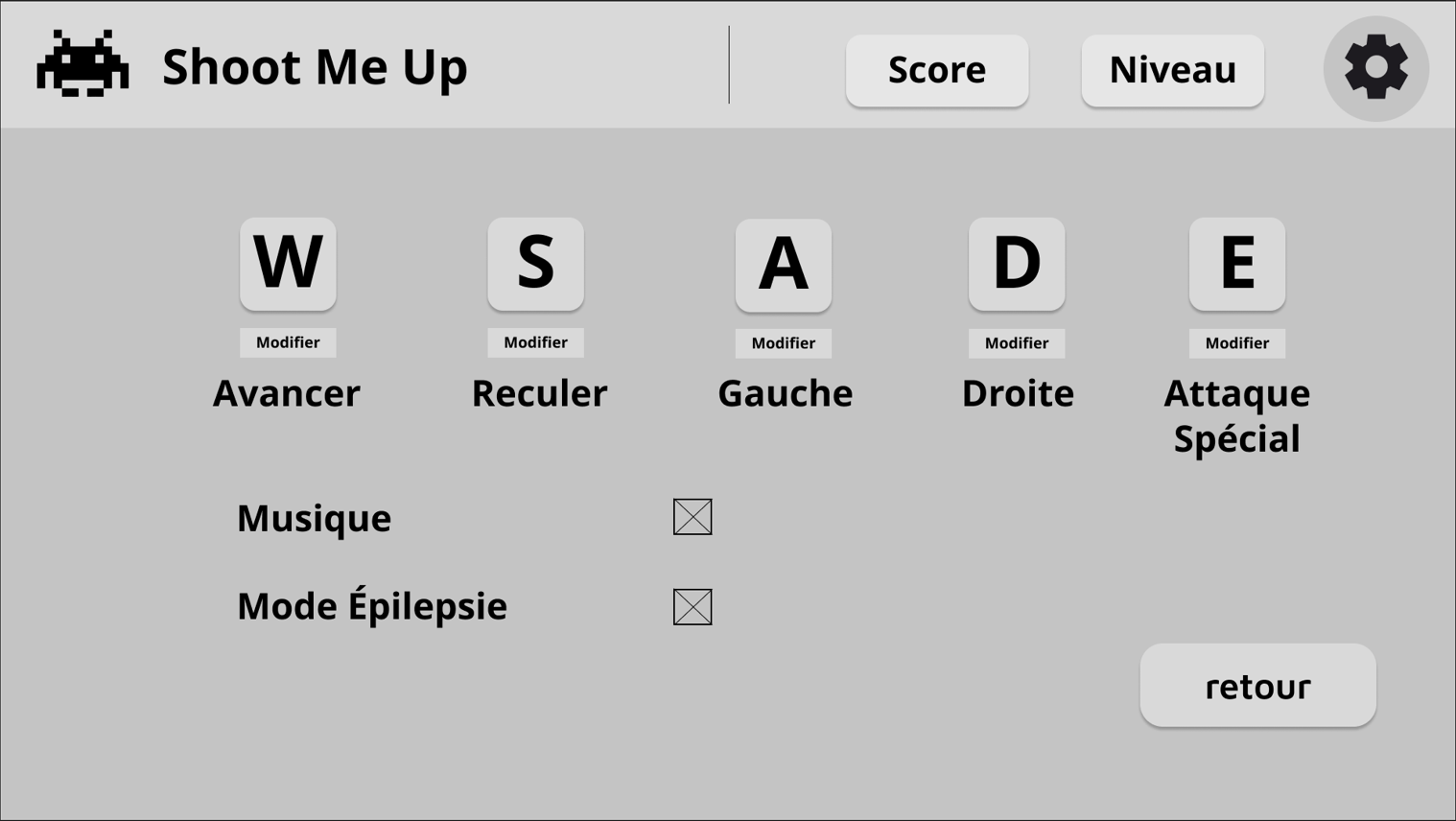
### Eco-Conception

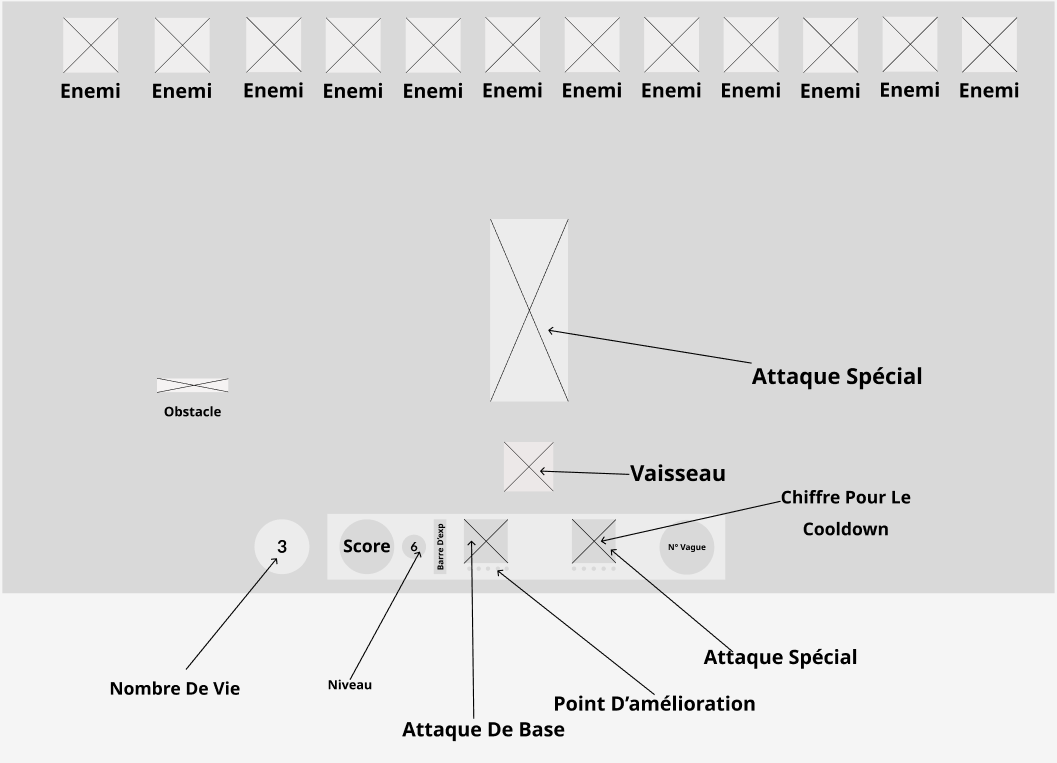
Pour l’éco-conception dans ce jeu j’ai plutôt choisi une approche de sensibilisation en remplaçant les aliens par des déchets qu’il faut détruire. Et en plus de ça toutes les images sont de qualité ultra réduite.

## Conception UX

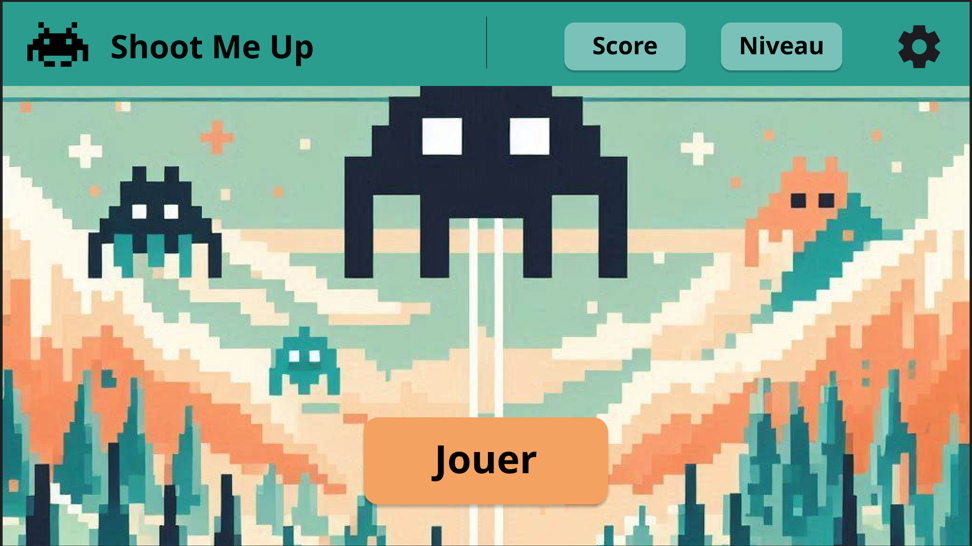
Voici la maquette basse fidélité

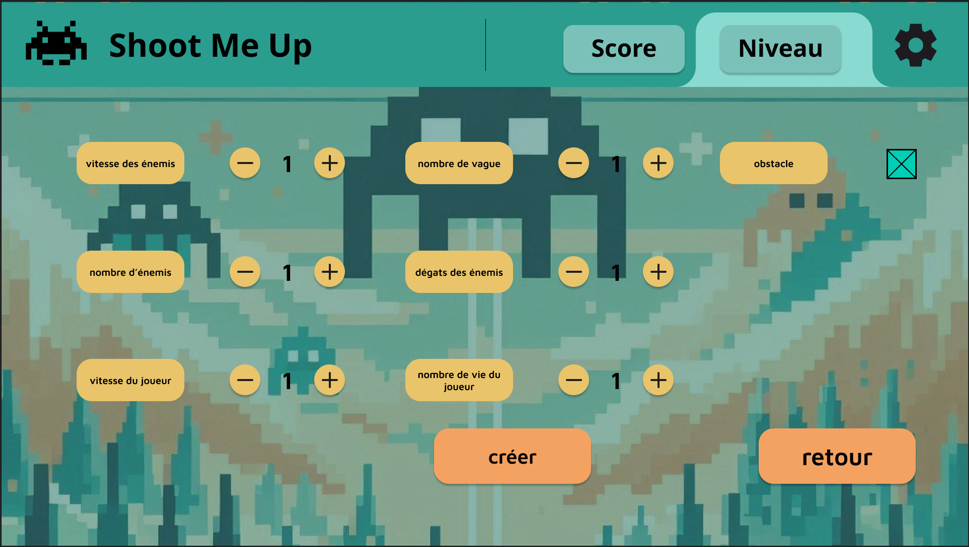
 Menu de base  Menu des scores

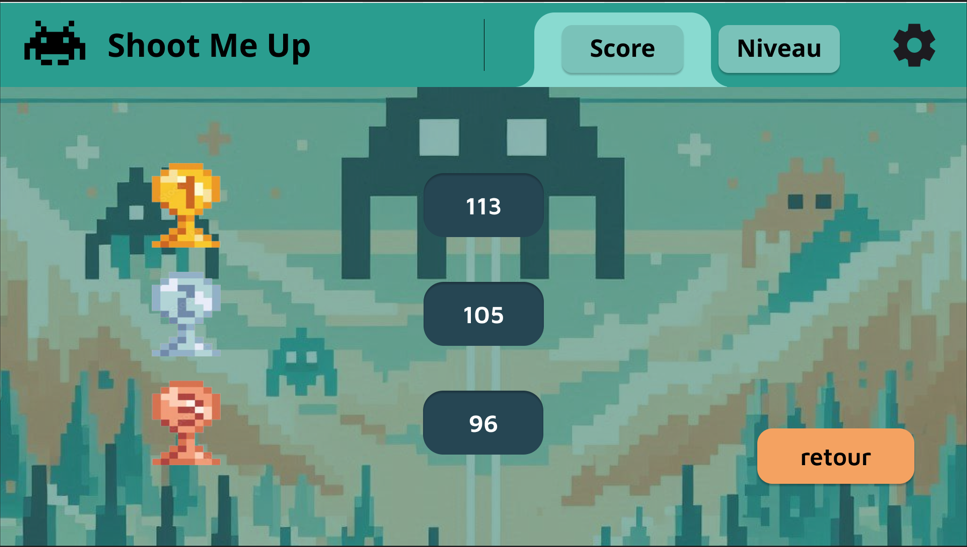
 Editeur de niveau Paramètre

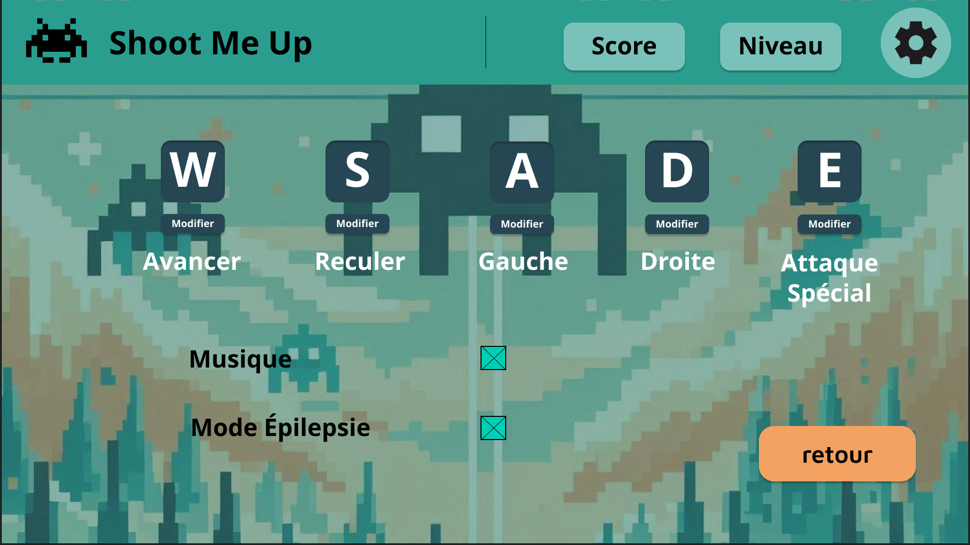
 Jeu

Voici la maquette haute-fidélité











## DB

### Schémas

## Journal de travail

Le journal de travail se trouve dans ce répertoire :

A ce stade, après l’analyse complète du projet, un planning détaillé et complet (avec tâches, sous-tâches, dépendances, durée, …) peut être finalisé.

Le planning détaillé doit s’inscrire dans le planning initial. Il faut que l’on puisse situer cette planification détaillée par rapport à la planification initiale.

# Réalisation

## Dossier de Réalisation

Cette partie permet de reproduire ou reprendre le projet par un tiers.

Pour chaque étape, il faut décrire sa mise en œuvre. Typiquement :

Versions des outils logiciels utilisés (OS, applications, pilotes, librairies, etc.)

Configurations spéciales des outils (Equipements, PC, machines, outillage, etc.)

Code source commenté des éléments logiciels développés.

Modèle physique d’une base de données.

Arborescences des documents produits.

Il faut décrire le parcours de réalisation et justifier les choix.

# Conclusion

## Bilan des fonctionnalités demandées

1. poo

Dans ce programme il y a un menu principal, un écran de jeu et une page pour les high scores. On peut se déplacer, le tir est automatique à part pour la capacité spécial et le joueur à un Sprite. Ensuite pour les ennemis tous les ennemis ont qu’une vie, il y a un système de vague d’ennemi, les ennemis tirent et ils ont tous un Sprite.

Obstacle à faire

2.UX

Il y a deux Personnas, une palette graphique, une éco-conception (centré plus dans la sensibilisation) et de l’accessibilité pour les personnes qui ont de l’épilepsie

Ensuite il y a tous les écrans du logiciel en basse fidélité, l’éditeur de niveau en haute-fidélité et en plus j’ai ajouté le menu principale, l’écran de jeu, les high-scores en haute-fidélité.