

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	INTRODUCTION	4
1.1	TITRE	4
1.2	DESCRIPTION	4
1.3	MATÉRIEL ET LOGICIELS À DISPOSITION.....	4
1.4	PRÉREQUIS.....	4
2	PLANIFICATIONS	4
3	PARTIE UX (LIEN DU DOSSIER UX - GITHUB).....	5
3.1	PERSONAS (LIEN POUR UNE MEILLEURE VISUALISATION).....	5
3.1.1	Persona 1	5
3.1.2	Persona 2	5
3.2	PALETTE GRAPHIQUE.....	6
3.2.1	Description et photo.....	6
3.3	ECO-CONCEPTION.....	7
3.3.1	Une technologie économe	7
3.3.2	Simplicité.....	7
3.4	ACCESSIBILITÉ.....	7
3.5	CONCEPTION	8
3.5.1	Maquettes base-fidélité (Lien pour une meilleure visualisation)	8
	• Jeu en cours	8
	• Menu principal	8
	• Fin de jeu	9
	• Editeur de niveau	9
	• Highscore.....	10
	• A propos.....	10
3.5.2	Maquette haute-fidélité (Lien pour une meilleure visualisation)	11
3.5.3	Fonctionnalités supplémentaires tirées des Personas plus élément original (Lien pour une meilleure visualisation).....	11
3.5.4	Choix effectués.....	13
4	PARTIE POO	14
4.1	ANALYSES FONCTIONNELLES (USERS STORIES)	14
4.2	AUTOMATIQUE DU CODE (MANUEL DE RÉFÉRENCE).....	14
4.3	SCHÉMAS DES CLASSES.....	14
4.4	IMPLÉMENTATION SPÉCIFIQUE.....	14
5	PARTIE DB.....	15
5.1	MCD	15
5.1.1	Description des tables et des entités	15
5.1.2	Description de relations et cardinalités	18
5.2	MLD.....	19
5.3	SCRIPT SQL.....	19
5.4	MISE EN PLACE DES TYPES D'UTILISATEURS.....	19
5.4.1	Explication	19
5.5	CRÉATION DES INDEX.....	20
5.5.1	Situation.....	20
5.6	BACKUP / RESTORE	21
5.6.1	Situation.....	21
6	RAPPORT DE TEST.....	22
6.1	TESTS POO	22

7	USAGE DE L'IA.....	22
7.1.1	P_UX.....	22
8	REFERENCES.....	22
8.1	LIENS UX.....	22
8.2	LIENS DB.....	22

1 INTRODUCTION

1.1 Titre

Shoot Me Up !

1.2 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.

1.3 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)

1.4 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

2 PLANIFICATIONS

La planification de chaque partie du projet, se trouve dans chaque dossier de chaque partie.


 P_DB (M. Mveng)	01.11.2024 10:26	Dossier de fichiers
 P_OO (M. Melly)	11.10.2024 16:32	Dossier de fichiers
 P_UX (Mme. Curchord)	28.10.2024 14:18	Dossier de fichiers

 JdT_Planif_GHE_PDB.xlsm	01.11.2024 10:25	Feuille de calcul ...	134 Ko
 MCB_MCB_S..._M_U_L_1	01.11.2024 10:25	Feuille de calcul ...	134 Ko

3 PARTIE UX (Lien DU DOSSIER UX - GITHUB)

3.1 Personas (Lien pour une meilleure visualisation)

3.1.1 Persona 1



Thomas Gonzalez
Professeur de Mathématiques

Sur lui :

- 30 Ans
- Genève
- Ingénieur civil
- Hétérosexuel
- Marié

Nostalgie Analytique Compétitif

Perfectionniste Curieux

Motivations :

- Explorer des jeux classiques qui ont marqué l'histoire des technologies et des loisirs numériques.
- Se reconnecter avec la culture des jeux vidéo en explorant des titres emblématiques comme un jeu de Space Invaders.
- Apprécier le design et les mécaniques de jeu rétro, avec une perspective technique et analytique.

Objectifs :

- Partager son intérêt pour les jeux vidéo avec ses collègues ou ses amis qui apprécient aussi les classiques.
- Décompresser et réduire le stress lié à son travail technique en prenant du plaisir à jouer.
- Éventuellement utiliser les jeux vidéo comme source d'inspiration pour ses propres projets en ingénierie.

Points de douleur :

- L'accessibilité des jeux (polices trop petites, menus compliqués).
- L'adaptation des touches qui changent chaque fois pour un nouveau jeu.

Personnalité

Rigoureux — Désordonné

Compétitif — Non-compétitif

Patient — Impatient

Perfectionniste — Laxiste

Organisée — Désorganisée


Compétences techniques

Connaissances en physique et mathématiques avancées

Programmation

Capacité à résoudre des problèmes

3.1.2 Persona 2



Jean-Michell Guignard
Homme retraité

Sur lui :

- 65 Ans
- Lausanne
- Informatique - Dev
- Hétérosexuel
- Célibataire

Nostalgie Simplicité Apprentissage continu

Temps de qualité Connexion intergénérationnelle

Motivations :

- Retrouver les sensations et la nostalgie des jeux d'arcade de son enfance.
- Occuper son temps libre de manière divertissante et stimulante.
- Se connecter avec ses petits-enfants ou ses amis en partageant des expériences de jeux vidéo.
- Se détendre après des années de travail en jouant à des jeux qui ne demandent pas trop de stress.

Objectifs :

- Jouer à des jeux qui rappellent ses jeunes années.
- Améliorer ses réflexes et sa rapidité malgré l'âge.
- Partager cette passion pour les jeux rétro avec ses proches.

Points de douleur :

- L'accessibilité des jeux (polices trop petites, menus compliqués).
- Les interfaces utilisateurs non adaptées aux seniors.

Personnalité

Nostalgique — Déconnecté

Social — Solitaire

Patient — Impatient

Résiliente — Fragile

Organisée — Désorganisée

Compétences techniques

Compétences en informatique infrastructure

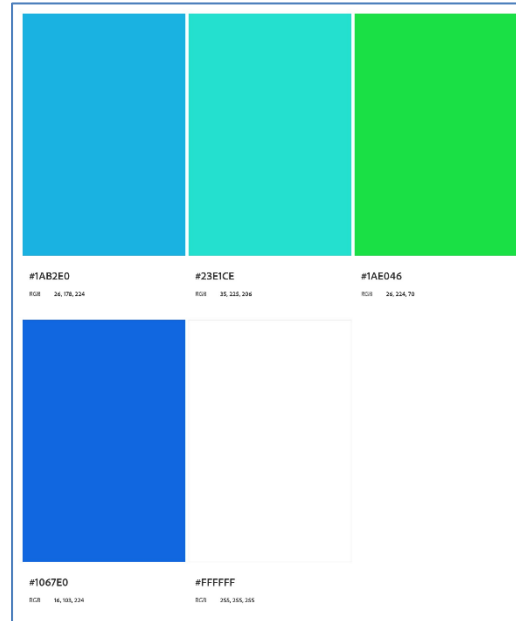
Réseaux Sociaux

Achats en ligne

3.2 Palette graphique

3.2.1 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 composée de couleurs plutôt vifs grâce à l'outil « **Adobe Color** » pour bien remarquer les composants de mon programme. L'harmonie chromatique est du type Semblable, et le mode colorimétrique est RVB.



3.3 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.

3.3.1 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.

3.3.2 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.

3.4 Accessibilité

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

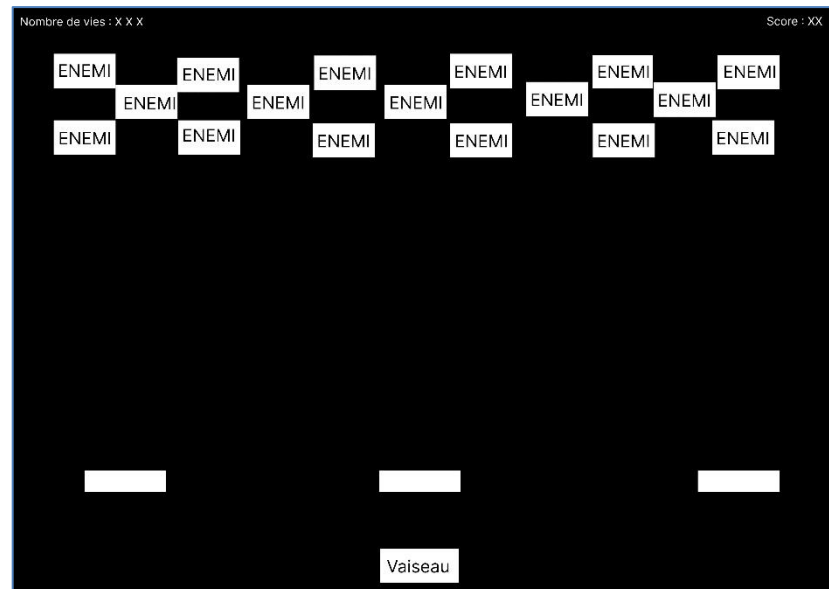
Thomas Gonzalez est un professeur qui aimerait plutôt se détendre et se reconnecter avec des jeux qu'il jouait quand il était jeune. Cependant, il a des difficultés avec les polices trop petites et aussi, il a la peine avec les nouvelles dispositions des touches qui chaque fois sont implémentées dans 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.

3.5 Conception

3.5.1 Maquettes base-fidélités ([Lien](#) pour une meilleure visualisation)

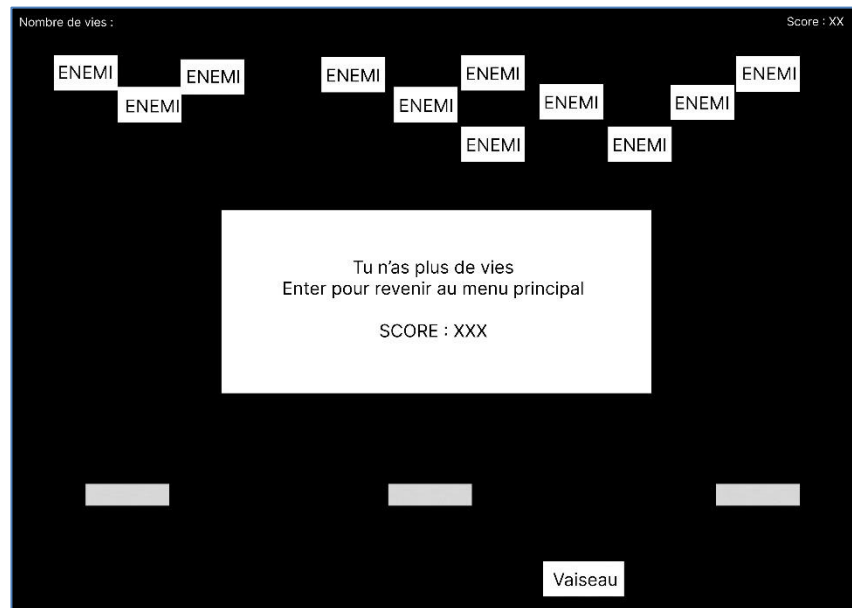
- *Jeu en cours*



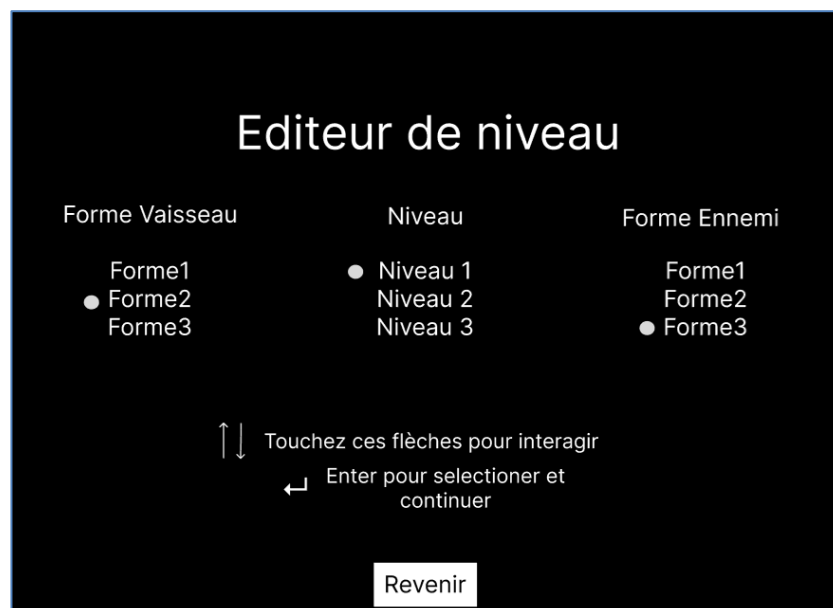
- *Menu principal*



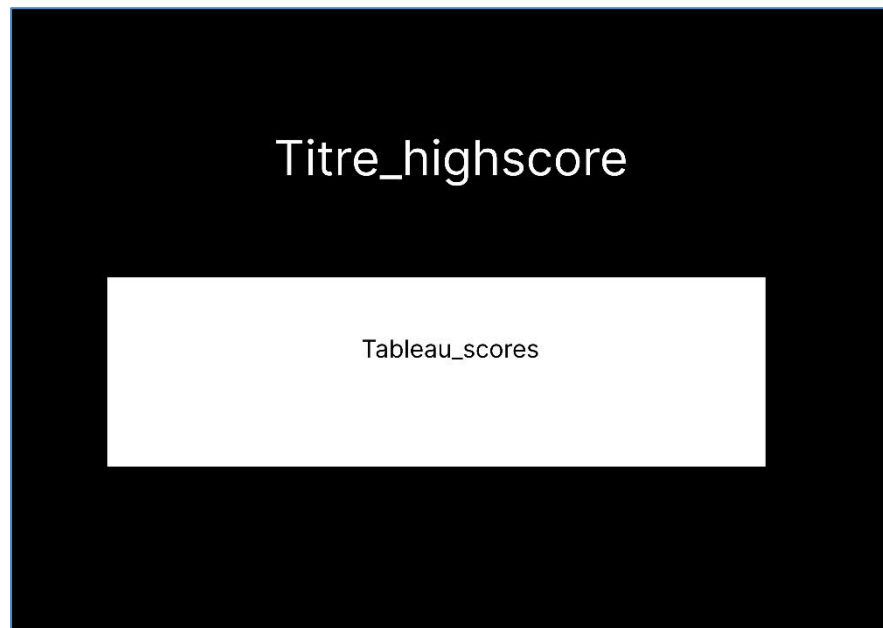
- *Fin de jeu*



- *Editeur de niveau*



- *Highscore*

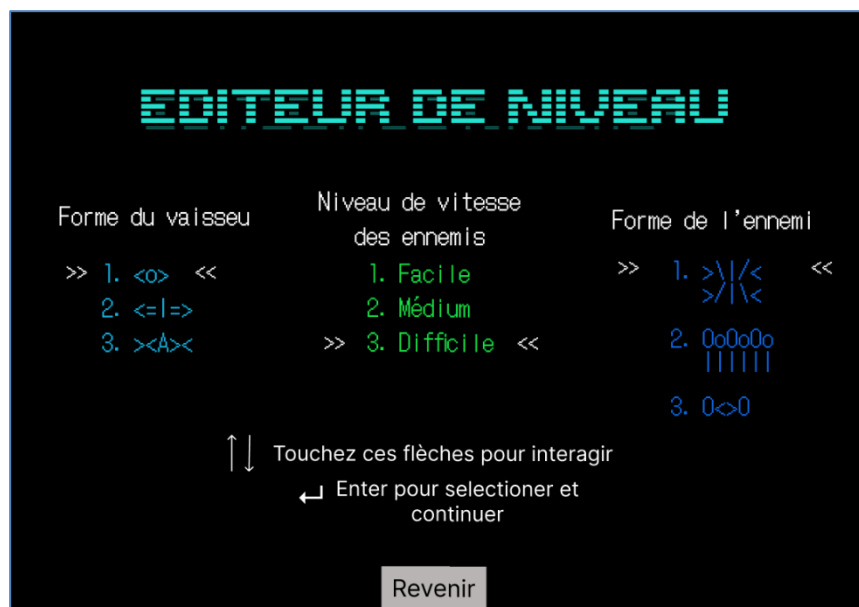


- *A propos*



3.5.2 Maquette haute-fidélité ([Lien](#) pour une meilleure visualisation)

Editeur de niveau



3.5.3 Fonctionnalités supplémentaires tirées des Personas plus élément original ([Lien](#) pour une meilleure visualisation)

Menu principal : Bouton pour l'augmentation des tailles de polices + nouvelle option pour le changement des touches pour interagir avec le jeu + option pour changer la langue du jeu.



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.



Langue : Cette option, c'est une option personnelle que me facilite beaucoup la vie, car il y a des gens qui ne sont pas forcément de langue française comme mon persona Jean-Michel Guignard, donc j'ai implémenté cette option pour les gens comment Thomas Gonzalez, n'a pas forcément la langue française comme langue maternelle et aussi comme lui, il pourrait avoir d'autres situations similaires.



3.5.4 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 d'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.
- **Fonctionnalités attirées aux personas** : Pour cette partie, j'ai juste pris les points faibles/douleur de chaque persona que j'ai faite puis j'ai modifié mes maquettes de façons que les points de douleurs des personas soient plus prises en compte pour eux de façons que le jeu est plus adapté à eux. Aussi j'ai choisi l'option de la langue, car de fois ça m'arrive qu'il y a des jeux qui ne sont pas dans une langue que je parle et j'aimerais bien que cette tpe de idéologie soient implémenté en tous les jeux/programmes.

4 PARTIE POO

4.1 Analyses fonctionnelles (Users Stories)

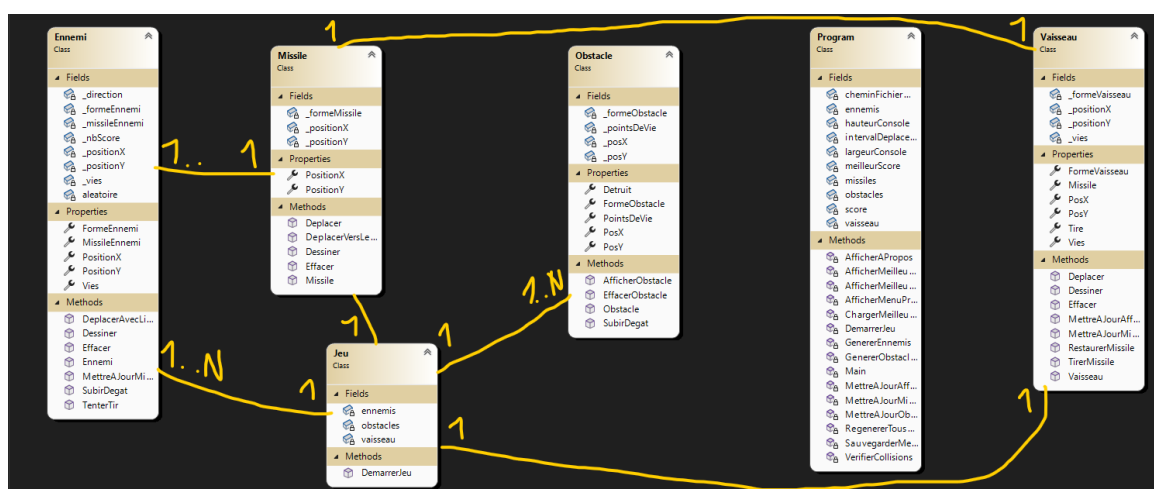
Voir le fichier Excel

4.2 Automatique du code (manuel de référence)

J'ai tenté de faire le Doc FX mais, ça ne montre pas toutes les classes. Je n'ai pas pu trouver pourquoi il ne montre pas toutes les classes.

Le Doc FX se trouve dans le répertoire avec le projet, P_POO.

4.3 Schémas des classes



4.4 Implémentation spécifique

Implémentation : Génération et Régénération des Ennemis

Dans mon code, cette fonctionnalité est répartie dans deux méthodes principales : `GenererEnnemis` et `RegenererTousLesEnnemis`. Ces méthodes sont appelées au début du jeu et après que tous les ennemis soient morts. Voici leur fonctionnement :

- **GenererEnnemis** : Cette méthode crée une première vague d'ennemis avec des positions initiales organisées en lignes et en colonnes.
- **RegenererTousLesEnnemis** : Cette méthode est appelée chaque fois que la liste des ennemis est vide, c'est-à-dire que le joueur a détruit tous les ennemis. Elle génère un nouvel ensemble d'ennemis de manière similaire à la première vague.

```
static void RegenererTousLesEnnemis()
{
    int nombreLignes = 3;           // Nombre de lignes d'ennemis
    int nombreEnnemisParLigne = 10; // Nombre d'ennemis par ligne
    int debutY = 2;                 // Position Y de départ des ennemis
    int largeurEnnemi = 5;          // Largeur d'un ennemi

    string[] formesEnnemis = { "[ ]-[ ]", "<<<*>>>", "00+00" }; // Formes possibles des ennemis

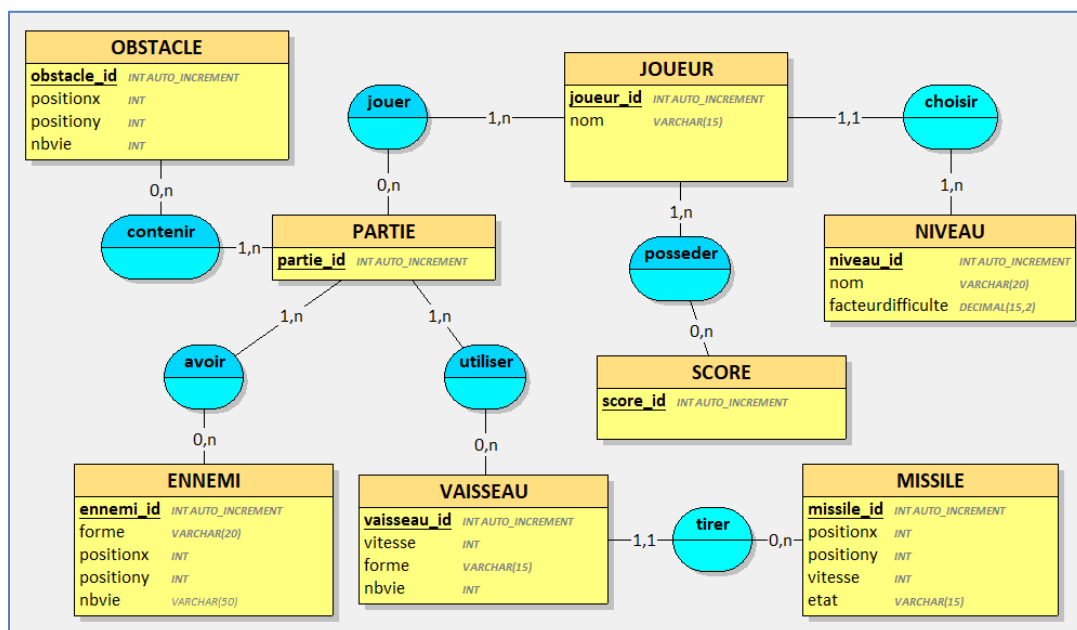
    int largeurTotaleEnnemis = nombreEnnemisParLigne * largeurEnnemi; // Largeur totale d'une ligne d'ennemis
    int espaceRestant = Console.WindowWidth - largeurTotaleEnnemis; // Espace restant sur la console
    int espacementXEnnemi = espaceRestant / (nombreEnnemisParLigne + 1); // Espacement entre chaque ennemi

    // Génération des lignes et colonnes d'ennemis
    for (int ligne = 0; ligne < nombreLignes; ligne++)
    {
        for (int col = 0; col < nombreEnnemisParLigne; col++)
        {
            int positionX = (col * largeurEnnemi) + ((col + 1) * espacementXEnnemi);
            int positionY = debutY + ligne * 2;

            if (positionX + largeurEnnemi <= Console.WindowWidth)
            {
                string forme = formesEnnemis[ligne % formesEnnemis.Length];
                Ennemi ennemi = new Ennemi(forme, 1, positionX, positionY, 10);
                ennemis.Add(ennemi); // Ajoute l'ennemi à la liste des ennemis
                ennemi.Dessiner();   // Dessine l'ennemi sur la console
            }
        }
    }
}
```

5 PARTIE DB

5.1 MCD



5.1.1 Description des tables et des entités

- **Table t_joueur**

Description : Représente un joueur dans le jeu.

Colonnes :

joueur_id : Identifiant unique du joueur.

nom : Nom du joueur.

niveau_id : Référence vers le niveau du joueur dans le jeu.

- **Table t_ennemi**

Description : Représente un ennemi dans le jeu.

Colonnes :

ennemi_id : Identifiant unique de l'ennemi.

Forme : Forme ou type visuel de l'ennemi.

positionx : Position de l'ennemi sur l'axe X.

positiony : Position de l'ennemi sur l'axe Y.

nbvie : Nombre de vies de l'ennemi.

- **Table t_missile**

Description : Représente les missiles lancés par le vaisseau ou les ennemis.

Colonnes :

missile_id : Identifiant unique du missile.

positionx : Position du missile sur l'axe X.

positiony : Position du missile sur l'axe Y.

vitesse : Vitesse du missile.

etat : État du missile (actif, inactif).

- **Table t_obstacle**

Description : Représente les obstacles dans le jeu.

Colonnes :

obstacle_id : Identifiant unique de l'obstacle.

positionx : Position de l'obstacle sur l'axe X.

positiony : Position de l'obstacle sur l'axe Y.

nbvie : Nombre de vies de l'obstacle.

- **Table t_partie**

Description : Représente une session de jeu.

Colonnes :

partie_id : Identifiant unique de la partie.

- **Table t_score**

Description : Représente le score d'un joueur dans une partie.

Colonnes :

score_id : Identifiant unique du score.

- **Table t_vaisseau**

Description : Représente le vaisseau contrôlé par le joueur.

Colonnes :

vaisseau_id : Identifiant unique du vaisseau.

vitesse : Vitesse du vaisseau.

forme : Forme ou type visuel du vaisseau.

nbvie : Nombre de vies restantes du vaisseau.

missile_id: Référence au type de missile utilisé par le vaisseau.

- **Table t_niveau**

Description : Représente les niveaux de difficulté dans le jeu.

Colonnes :

niveau_id : Identifiant unique du niveau.

nivNom : Nom du niveau.

nivFacteurdifficulte : Facteur de difficulté pour le niveau.

5.1.2 Description de relations et cardinalités

- **t_joueur - t_partie (N-N)**

Relation : Plusieurs joueurs par partie et vice-versa

Table d'association : jouer

- **t_partie - t_ennemi (N-N)**

Relation : Plusieurs ennemis par partie et vice-versa

Table d'association : avoir

- **t_partie - t_obstacle (N-N)**

Relation : Plusieurs obstacles par partie et vice-versa

Table d'association : contenir

- **t_joueur - t_score (1-N)**

Relation : Un joueur peut avoir plusieurs scores

Table d'association : posséder

- **t_joueur - t_niveau (N-1)**

Relation : Plusieurs joueurs peuvent être dans un même niveau

Table d'association : FK dans t_joueur

- **t_vaisseau - t_missile (N-1)**

Relation : Plusieurs vaisseaux peuvent utiliser le même missile

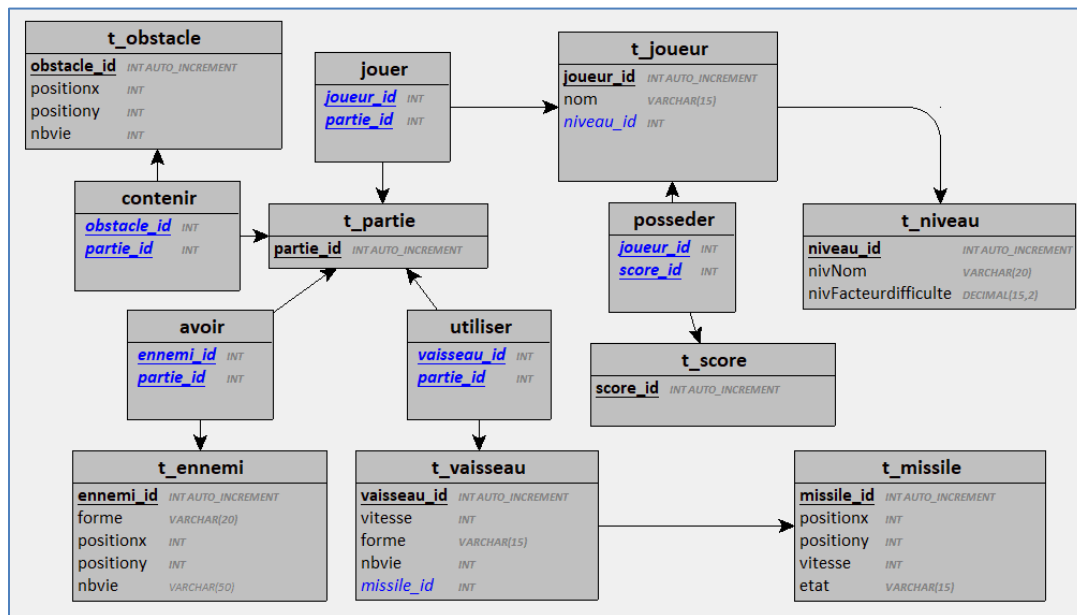
Table d'association : FK dans t_vaisseau

- **t_vaisseau - t_partie (N-N)**

Relation : Plusieurs vaisseaux par partie et vice-versa

Table d'association : utiliser

5.2 MLD



5.3 Script SQL

Le script SQL se trouve dans le même répertoire de ce rapport sous le nom de **db_shootmeup.sql**.

5.4 Mise en place des types d'utilisateurs

5.4.1 Explication

Pour gérer différents types d'utilisateurs de manière simple, on peut créer des rôles pour chaque type (Administrateur, Joueur) avec des permissions spécifiques. Chaque utilisateur est ensuite assigné à un rôle, ce qui facilite la gestion des droits d'accès sans avoir à définir des permissions pour chaque utilisateur individuellement.

Voici un exemple de comment on pourrait mettre en place cela :

```
1 CREATE ROLE administrateur;
2 CREATE ROLE joueur;
3
4
5 GRANT SELECT, INSERT, UPDATE, DELETE ON `db_shootmeup`.* TO administrateur;
6 GRANT SELECT ON `db_shootmeup`.* TO joueur;
7
8
9 CREATE USER 'AdminUser'@'localhost' IDENTIFIED BY 'admin_password';
10 GRANT administrateur TO 'AdminUser'@'localhost';
11
```

5.5 Création des index

5.5.1 Situation

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 ?**

Des index sont définis à travers les contraintes de clé primaire et les clés étrangères. Les clés primaires créent automatiquement un index unique pour garantir l'unicité des valeurs de ces colonnes de façons ça facilite l'accès rapide à la base de données.

- **Quels sont les avantages et les inconvénients des index ?**

Avantages :

- Réponses de requêtes plus rapides avec des SELECT, WHERE et ORDER BY, cela nous permet un accès plus rapide aux données.
- Améliore la performance lors des jointures entre tables en réduisant le nombre de requêtes nécessaires pour accéder aux données liées.

Inconvénients :

- La consommation d'espace de stockage est augmentée chaque fois qu'un index est ajouté.
- S'il y a trop d'index redondants ou inutiles, alors il se pourrait mener à une fragmentation de la base de données.

- **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.**

Ajouter un index sur le champ nom de la table t_joueur serait utile parce que, dans une application en C#, on pourrait souvent vouloir chercher les informations d'un joueur en utilisant son nom.

Par exemple, imaginons qu'on veuille afficher les scores ou les statistiques d'un joueur. Si l'application cherche régulièrement un joueur spécifique par son nom, l'index aiderait à trouver rapidement ses données, sans avoir à fouiller toute la table à chaque fois.

Un index sur « nom » rendrait aussi d'autres fonctions plus rapides, comme afficher une liste des meilleurs joueurs ou chercher des joueurs avec des noms similaires.

5.6 Backup / Restore

5.6.1 Situation

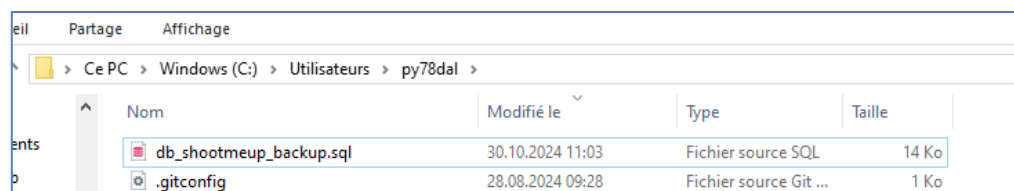
Nous souhaitons réaliser une sauvegarde (Backup) de votre base de données. Ensuite, nous souhaitons nous assurer que cette sauvegarde est correcte en la rechargeant dans MySQL (opération de restauration). Donner la commande permettant de faire :

- **Un backup de votre base de données**

Pour le faire, il faut d'abord identifier si on travaille sur un environnement Docker ou Uwamp, dans ce cas, on travaille sur l'environnement docker et on fait la suivant commande sur une console.

```
1 docker exec -i db mysqldump -u root -proot --databases db_shootmeup > db_shootmeup_backup.sql
```

Ensuite, nous verrons le fichier backup dans notre répertoire personnel sur Windows.



Nom	Modifié le	Type	Taille
db_shootmeup_backup.sql	30.10.2024 11:03	Fichier source SQL	14 Ko
.gitconfig	28.08.2024 09:28	Fichier source Git ...	1 Ko

- **Un restore de votre base de données En expliquant en détail chaque commande utilisée.**

Avant de faire un restore de base de données, il faut aller sur notre base de données, soit avec MySQL ou phpMyAdmin et supprimer la base de données de base, pour qu'il n'ait pas des soucis au moment d'importer le backup.

```
mysql>
mysql> DROP DATABASE db_shootmeup;
Query OK, 13 rows affected (0.22 sec)

mysql>
```

Ensuite, on ouvre une console, et on exécute la suivant commande :

Cette commande, prendra le fichier sauvegarde .sql et le mettra dans notre base de données.

```
C:\Users\py78dal>docker exec -i db mysql -u root -proot db_restauracion > db_shootmeup_backup.sql;
mysql: [Warning] Using a password on the command line interface can be insecure.
C:\Users\py78dal>
```

ATTENTION : Il ne faut pas oublier, de créer la base de données avant de restaurer la base de données.

```
mysql> CREATE DATABASE db_restauracion;
Query OK, 1 row affected (0.02 sec)
```

6 RAPPORT DE TEST

6.1 Tests POO

J'ai fait trois tests, donc un pour vérifier que la vie du joueur se charge correctement, qu'il y ait bien 3 vies.

Le deuxième test est pour vérifier que l'ennemi perd une vie dès qu'il est touché par un missile

Le troisième test est pour vérifier les collisions des missiles avec les obstacles.

7 USAGE DE L'IA

7.1.1 P_UX

Pour cette partie des projets, je n'ai pas beaucoup utilisé l'aide d'une intelligence artificiel puisque pour la conception des maquettes, c'est l'imagination qui compte le plus, cependant, je me suis aidé un peu pour les données de mes personas, mais à la fin j'ai orienté plutôt mes personas vers un côté personnel, ce sont des gens que je connais et son proche à moi.

8 REFERENCES

8.1 Liens UX

- [Support de cours du module et annexes I322](#)
- [Cahier de charges de ce projet](#)
- [ChatGPT](#)
- [Adote Color Palette](#)
- [ASCII Art Archive](#)
- [115 Bonnes pratiques – Eco-Conception](#)

8.2 Liens DB

- [ChatGPT](#)
- [Cahier de charges de ce projet](#)
- [Looping](#)
- [MySQL](#)
- [Docker Desktop](#)