Shoot Me Up

(Image créée par ChatGPT)



Auteur : Moser Charles-Henri – CID 2B

Vennes

10 Semaines

Chefs de projets :

Antoine Mveng

Aurélie Curchod

Jonathan Melly

Table des matières

[1 Spécifications 3](#_Toc181390668)

[1.1 Titre 3](#_Toc181390669)

[1.2 Description 3](#_Toc181390670)

[1.3 Matériel et logiciels à disposition 3](#_Toc181390671)

[1.4 Prérequis 3](#_Toc181390672)

[1.5 Cahier des charges 3](#_Toc181390673)

[1.5.1 Objectifs SMART 3](#_Toc181390674)

[1.5.2 Travail à réaliser par l'apprenti 4](#_Toc181390675)

[1.5.3 Si le temps le permet … 4](#_Toc181390676)

[1.6 Les points suivants seront évalués 4](#_Toc181390677)

[1.7 Validation et conditions de réussite 4](#_Toc181390678)

[2 Analyse 4](#_Toc181390679)

[2.1 Data base 4](#_Toc181390680)

[2.2 UX/UI 5](#_Toc181390681)

[2.3 Programmation/ Code 7](#_Toc181390682)

[3 Conclusion 7](#_Toc181390683)

[3.1 Bilan des fonctionnalités demandées 7](#_Toc181390684)

[3.2 Bilan personnel 7](#_Toc181390685)

[4 Divers 8](#_Toc181390686)

[4.1 Webographie 8](#_Toc181390687)

[4.2 Utilisation de ChatGPT : 8](#_Toc181390688)

[5 Annexes 8](#_Toc181390689)

# Spécifications

## Titre

*Shoot me up !*

## Description

Le projet Shoot Me Up! est un jeu de tir 2D modulaire inspiré du genre classique Space Invader. L'objectif est de créer un jeu amusant, engageant et interactif, offrant aux joueurs des niveaux de plus en plus difficiles. Avec un accent sur la programmation orientée objet (POO), la gestion de bases de données et la conception d'expérience utilisateur (UX).

Ce projet vise non seulement à développer un jeu jouable, mais aussi à démontrer des compétences en gestion de projet, codage et conception UX/UI.

## Matériel et logiciels à disposition

|  |  |
| --- | --- |
| * Un PC ETML * Accès à Internet * Visual Studio * PhpMyAdmin | * <https://etml.icescrum.com> * <https://figma.com> * Looping |

## 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

## Cahier des charges

### Objectifs SMART

L'objectif principal de Shoot Me Up! est de créer un jeu Space Invader.

Les objectifs SMART (Spécifiques, Mesurables, Atteignables, Réalistes, Temporels) pour ce projet sont :

Spécifique : Développer un jeu avec deux niveaux complets, des contrôles de personnage, des ennemis et un suivi des scores.

Mesurable : Le projet sera évalué sur l'intégralité des fonctionnalités du jeu, y compris les niveaux, les actions du joueur, les interactions ennemies et la base de données pour les scores.

Atteignable : Compte tenu de la durée de 10 semaines et des outils disponibles (PC, GitHub, Figma, MySQL, C#), l'étendue du projet a été définie de manière que les composants essentiels soient mis en œuvre tout en laissant la place à des fonctionnalités supplémentaires.

Réaliste : Le projet a été conçu avec une compréhension claire des outils de développement et des compétences requises (programmation, bases de données, conception UX/UI).

Temporel : Le projet sera achevé en 80 périodes sur 10 semaines, avec des jalons et livrables clairement définis.

### Travail à réaliser par l'apprenti

Le travail requis se trouve dans le [cahier des charges (CdC)](Shoot%20Me%20Up-CdC.docx) en annexe

### Si le temps le permet …

* Si le temps le permet, Il faudrait créer un prototype Figma
* Lier le code avec la base de données qu’on a créé.

## 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

# Analyse

## Data base

**[](DataBase/ScriptSQLCreation.sql)[](DataBase/InsertData.sql)[](DataBase/UsersRoles.sql)**

**Pourquoi certains index existent déjà ?**

Car le SGBDR créée systématiquement un index chaque fois que l’on pose une clef primaire (PRIMARY KEY) ou une contrainte d’unicité (UNIQUE) sur une table. En revanche, il n’y a pas d’index créé automatiquement par le SGBDR derrière une FOREIGN KEY (clef étrangère).

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

**Avantages :**

* Amélioration des performances de recherche : la création d’un index sur une colonne fréquemment utilisée dans les requêtes de recherche réduit considérablement le temps nécessaire à la récupération des résultats pertinents.
* Optimisation des requêtes : lorsque des index sont utilisés, les moteurs de base de données peuvent choisir des plans d’exécution plus efficaces pour exécuter les requêtes.
* Ordre et tri des données : les index permettent également de trier efficacement les données.

**Inconvénients :**

* Augmentation de la taille de la base de données : Les index prennent de l’espace de stockage, surtout s’ils sont nombreux ou sur de grandes colonnes.
* Impact sur les opérations d’écriture : Les insertions, mises à jour et suppressions deviennent plus coûteuses en termes de performance, car chaque modification des données doit également être appliquée à l’index.
* Maintenance : Les index nécessitent une maintenance (par exemple, pour assurer leur mise à jour ou en cas de changements de structure de la table), ce qui peut nécessiter des ressources supplémentaires.

**CREATE INDEX idx\_score ON t\_score (scoreResultat);**

**Justification** : Si l’application affiche souvent les scores des joueurs ou trie les scores pour afficher les classements, un index sur scoreResultat accélérerait les requêtes de classement.

**--Commande de sauvegarde**

docker exec -i db mysqldump -u root -proot db\_shoot\_em\_up > C:\Users\pv23aha\Desktop\sauvegarde\_db\_shoot\_me\_up.sql

**--Commande de restauration**

docker exec -i db mysql -u root - proot db\_shoot\_em\_up < C:\Users\pv23aha\Desktop\sauvegarde\_db\_shoot\_me\_up.sql

## UX/UI

**Personas:**

Persona 1 : [Edward Meyard](UI_UX/Personas/Persona%20Edward.pdf), 40 ans. Il est un professionnel de la finance et il est passionné par le Tennis. Il joue principalement à des jeux court ou des jeux de détente afin de pouvoir continuer à travailler.

Persona 2 : [Timothé Gartrand](UI_UX/Personas/Persona%20Timothé.pdf), 14 ans. C’est un grand amateur de jeux vidéo Il y passe beaucoup de temps avec ses amis. Il passe sa journée au cours de son école et il trouve du réconfort dans l’action dynamique des jeux vidéo.

Persona 3 avec une spécificité : [Dwight Croisier](UI_UX/Personas/Persona%20Dwight.pdf), 25 and. Il est médiamaticien. Passionné depuis toujours par le dessin et l’informatique. Il veut apprendre à jouer au jeux vidéo malgré sa main manquante à la suite d’un accident.

**Palette de couleur :**

J’ai choisi une palette de couleur à contraste élevé et un design rétro qui évoque l'univers spatial. Les couleurs principaux sont le noir, le cyan néo et les ennemis sont bleu et rouges.

**Eco-conception :**

Mode sombre de base pour limiter la consommation des écran OLED

**Accessibilité :**

Dans les options on peut modifier les touches (peut aider Dwight pour pouvoir jouer au jeu à une seule main).

**Conception :**

Définition de tous les écrans – [maquettes basse-fidélité](UI_UX/LowFidelity/Basse%20fidelité%20Space%20Invaders.pdf):

Ma maquette basse fidélité est une représentation simple et schématique de mon jeu.

Écran Principal (Menu) : Options principales avec Jouer, Multijoueur, Sélection de niveau, Score, Éditeur de niveaux, Options, et Quitter.

Écran de Score : Affiche les meilleurs scores avec le nom et la position des joueurs. Un mode jour/nuit est proposé pour s’adapter aux préférences de luminosité des utilisateurs.

Écran des Options : Paramètres ajustables tels que le volume, la configuration du clavier, la sélection de la langue (anglais ou français), et le choix du thème de fond. Des options de personnalisation des ennemis sont également disponibles.

Écran de Sélection de Niveau : Permet de choisir la difficulté (Facile, Normal, Difficile) et de sélectionner l'environnement de jeu, ainsi que le type de vaisseau spatial.

Éditeur de Niveaux : Offre la possibilité de sélectionner des objets, d'ajuster leurs points de vie (HP) et leurs dégâts. Des boutons facilitent l'ajout d'entités et permettent un retour au menu principal.

Écran Multijoueur : Indique l’état d’attente des autres joueurs avant le début de la partie, avec la possibilité de retourner au menu principal.

Écran de Jeu : Présente des indicateurs de vies et de munitions en bas de l’écran pour une gestion facile et intuitive de la progression du joueur.

Écran “Éditeur de niveau” - [maquette haute-fidélité](UI_UX/HighFidelity/Haute%20Fidelité%20Sapce%20Invaders.pdf):

Sur la maquette haute-fidélité, les utilisateurs peuvent choisir les types d’éléments à placer et les organiser librement sur l’interface de jeu. Cela leur permet de créer et de personnaliser leurs niveaux selon leurs envies, offrant une dimension créative en plus et aide à la rétention de joueurs.

**Ajout de fonctionnalités supplémentaires :**

Configuration des touches du clavier :

Une option cruciale pour les joueurs à mobilité réduite, permettant de personnaliser l’expérience de jeu selon leurs besoins.

Contraste élevé :

Pour garantir une meilleure visibilité et un confort de jeu optimal, même dans des conditions de lumière difficiles.

Mode multijoueur :

Une fonctionnalité innovante qui ajoute une touche compétitive à ce jeu Space Invaders, surprenant et engageant les joueurs modernes.

## Programmation/ Code

### **Une analyse fonctionnelle centrée utilisateur sous forme de User Stories**

En tant que joueur, je veux déplacer mon vaissseau vers la droite et la gauche afin d’éviter les tires des ennemies.

En tant que joueur, je veux pouvoir tirer des projectiles afin de détruire les envahisseirs.

En tant que joueur, je veux savoir combien j’ai de vie pour pouvoir jouer plus stratégiquement

En tant que joueur, je veux que des ennemis tire sur moi afin d’augmenter la difficulté du jeu.

En tant que joueur, je veux gagner des points pour chaque ennemi détruit afin de pouvoir me comparer au autres.

En tant que joueur, je veux voir mon score actuel afin de suivre ma performance durand la partie.

En tant que joueur, je veux avoir des obstacle afin de pouvoir me protéger contre les tires ennemies.

En tant que joueur, je veux que les obstacle se cassent au bout d’un moment pour avoir plus de difficulté.

En tant que joueur, je veux avoir un tableau avec les meilleurs résultat afin de voir mes meilleures performances

### **Schémas des classes**

### A group of rectangular signs Description automatically generated

### **Au moins un détail d’implémentation spécifique**

**Gestion du Pooling pour les Balles**

Dans le jeu **ShootWinForms**, le pooling d'objets est utilisé pour optimiser la gestion des balles tirées par les vaisseaux. Cela évite la création et la suppression constantes d'objets, ce qui pourrait entraîner des problèmes de performance.

1. **Principe du Pooling** :
   * Une liste statique bulletPool est utilisée pour stocker des balles inactives. Ces balles sont réutilisées au lieu d'en créer de nouvelles à chaque tir.
2. **Initialisation des Balles** :
   * Lorsqu'une balle est nécessaire, la méthode statique GetBullet vérifie si une balle est disponible dans le pool.
   * Si le pool contient des balles, l'une d'elles est récupérée, réinitialisée avec les nouvelles propriétés (position de départ, vitesse, image), et utilisée.
   * Si le pool est vide, une nouvelle balle est créée.
3. **Désactivation et Retour au Pool** :
   * Lorsqu'une balle quitte l'écran, la méthode Deactivate est appelée pour rendre la balle invisible et la retirer du formulaire de jeu.
   * La balle est ensuite renvoyée au pool avec ReturnBullet, ce qui permet de la réutiliser plus tard.

Cette implémentation assure une meilleure efficacité mémoire et un gameplay fluide en minimisant les opérations de création et destruction d'objets.

**Gestion du Mouvement des Envahisseurs**

L'implémentation du mouvement des envahisseurs dans le projet **ShootWinForms** est un exemple efficace de gestion des entités en jeu. Voici comment cette logique est structurée :

1. **Direction et Limites** :
   * La classe InvadersManager maintient une liste de tous les envahisseurs (InvadersList) et utilise un booléen movingRight pour déterminer la direction de leur déplacement.
   * Le mouvement est contrôlé par la méthode MoveInvaders, qui vérifie si un envahisseur atteint les bords gauche ou droit de l'écran.
     + Si un envahisseur atteint le bord droit, movingRight est réglé sur false, les envahisseurs changent de direction et descendent d'un cran.
     + Si un envahisseur atteint le bord gauche, movingRight est réglé sur true, et les envahisseurs descendent également.
2. **Vitesse de Déplacement** :
   * Une vitesse de déplacement (speed) est définie pour ajuster la rapidité du mouvement des envahisseurs. Cette vitesse peut être modifiée pour influencer la difficulté du jeu.
3. **Méthode MoveInvadersDown** :
   * Cette méthode est appelée chaque fois que les envahisseurs atteignent une limite de l'écran. Elle déplace tous les envahisseurs vers le bas d'une distance fixe.
   * Cela crée une dynamique de jeu qui devient progressivement plus difficile, car les envahisseurs se rapprochent du joueur.
4. **Gestion Efficace du Mouvement** :
   * L'approche permet un contrôle précis et efficace de l'ensemble du groupe d'envahisseurs, imitant le comportement des jeux d'arcade classiques.
   * En évitant des calculs individuels pour chaque envahisseur, le jeu reste fluide, même lorsque de nombreux ennemis sont affichés à l'écran.

# Conclusion

## Bilan des fonctionnalités demandées

**Résumé des fonctionnalités décrites dans les spécifications de départ :**

**Niveaux complets :** Les deux niveaux n'ont pas encore été entièrement réalisés. Le premier niveau est presque terminé, avec des ennemis et des obstacles bien définis, mais le second niveau demande encore du travail. Ayent les ennemis déjà créés et fonctionnels je peux facilement créer un deuxième niveau en ajoutant un 2ème forms, il y a beaucoup de moyens. Je pense qu’avec environ 8 périodes supplémentaires cette partie peut être finalisée.

**Contrôles de vaisseau :** Les commandes du Vaisseau sont fonctionnelles et réactives, offrant une expérience de jeu fluide. Toutefois, des ajustements mineurs pourraient être apportés pour améliorer encore la réactivité, notamment en cas de changements rapides de direction.

**Ennemis et interactions :** Les ennemis apparaissent correctement, mais leur comportement, notamment le tir des bullets, présente encore des problèmes. Les bullets ennemies ne fonctionnent pas comme prévu.

**Suivi des scores :** Cette fonctionnalité n'est pas encore implémentée. La structure de la base de données est en cours de conception, mais l'intégration complète pour enregistrer et afficher les scores reste à faire

**Base de données pour les scores :** La base de données n'est pas encore connectée au jeu pour le suivi des scores. Des travaux supplémentaires sont requis pour assurer le bon fonctionnement et la mise à jour des scores dans la base de données.

**Conclusion des fonctionnalités :** En conclusion, la majorité des fonctionnalités principales sont partiellement opérationnelles, mais certains aspects essentiels, comme les niveaux complets et l'intégration de la base de données pour les scores, demandent encore des ajustements et du développement supplémentaire.

## Bilan personnel

J’aime beaucoup l’idée de pouvoir créer un projet « complet » relier à une base de données un programme C# Windows Forms. Mais je pense que l’expérience requise pour ce projet étais trop élevé par rapport à notre niveau actuel.

La gestion de mon temps durant ce projet a été désastreuse car je (mon point de vue) ne possédais pas assez d’information par rapport au projet, aussi le manque d’informations dans le CdC a aussi été un Poin vicieux.

Pour commencer je faisais mon code sur Console au lieu de forms, ceci et d’autre problèmes qui sont survenu plus tard, comme les classes et tests à intégrer mon fait perdre énormément de temps. Du fait que je recommençais.

Ce projet m’a appris à créer un jeu en programmation orienté objet.

À créer des persona pour l’UX et à bien utiliser figma.

J’ai appris à créer des rôles et des utilisateurs en base de données.

Améliorations : De mon point de vue je pense que le projet aurait dû survenir après avoir appris les thèmes correspondant afin de ne pas être perdu au début.

Je pense aussi que des améliorations au niveau de la clarté des indications peuvent aider.

# Divers

## Webographie

[Stack Overflow - Where Developers Learn, Share, & Build Careers](https://stackoverflow.com/#:~:text=Stack%20Overflow%20is%20the%20largest,%20most%20trusted%20online%20community%20for)

[ChatGPT](https://chatgpt.com/)

[GitHub](https://github.com/C-HM)

## Utilisation de ChatGPT :

POO : Utilisation pour aide à la compréhension d’un code trouvé sur GitHub

UX : Pas d’utilisation de ChatGPT

DB : Utilisation de Chat GPT pour aide à l’attribution des Rôles (GRANT)

Rapport : Aide à la conjugaison et orthographe

Aide pour une base de texte modifiée par moi-même.

# Annexes

[Cahier des Charges](Shoot%20Me%20Up-CdC.docx)

[Journal de Travail et Planification](JournalC-H.xlsm)