

Projet OceanInvader

Table des matières

1	Analyse préliminaire	3
1.1	Introduction	3
1.2	Objectifs.....	3
1.3	Gestion de projet	5
2	Analyse / Conception.....	5
2.1	Gameplay	5
2.2	Concept	5
2.3	Analyse fonctionnelle.....	7
2.3.1	Mouvement (joueur).....	7
2.3.2	Point de vie (joueur).....	7
2.3.3	Attaque (joueur).....	7
2.3.4	Régénération de point de vie (joueur).....	7
2.3.5	Obstacle	8
2.3.6	Perdre.....	8
2.3.7	Ennemi comportement.....	8
2.3.8	Menu.....	8
2.4	UX.....	9
2.5	Stratégie de test.....	15
3	Réalisation.....	16
3.1	Déroulement	16
3.2	Mise en place de l'environnement de travail.....	17
3.3	Description des tests effectués.....	17
3.3.1	Sprint 1	17
3.4	Erreurs restantes	18
4	Conclusions	18
5	Annexes.....	20
5.1	Journal de travail	20

1 Analyse préliminaire

1.1 Introduction

Ce projet a été réalisé dans le cadre d'une formation d'informaticien CFC à l'ETML. Le but est de réaliser individuellement un jeu en deux dimensions en se basant sur la programmation orienter objet, sujet enseigner parallèlement au projet pendant le module ****. Les apprentis ont le choix sur le style et la forme du jeu à réaliser. Le jeu est fait en langage C# à l'aide de l'IDE Visual Studio 2022. La gestion du projet est faite avec IceScrum, et la gestion des fichiers avec GitHub. Ce projet permet à l'apprenti de d'apprendre ou de se perfectionner dans la programmation orienter objet, et d'appliquer la gestion de projet « agile » via IceScrum.

1.2 Objectifs

Le but de ce projet est de réaliser un jeu en 2D en windows forms qui sera composé des caractéristiques suivantes :

- a. Maquettes
 - i. Menu principal
 - ii. Ecran de jeu (niveau)
 - iii. Éditeur de niveau (voir détails ci-dessous)
 - iv. High scores
- b. Contraintes de réalisation
 - i. Un concept de niveaux décrivant
 - 1. Le numéro du niveau (Level 1, Level 2, ...)
 - 2. Le joueur
 - a. Déplacements
 - b. Nombre de vies
 - c. Capacités de tir : direction, rafale, cooldown, décompte munitions, recharge, ...
 - d. Un sprite
 - 3. Les ennemis du niveau avec (pour chaque type)
 - a. Nombre de vies
 - b. Minutage d'apparition
 - c. Tir (oui / non)
 - d. Un sprite
 - 4. Les obstacles avec (pour chaque type)

- a. Une taille
 - b. Une position X,Y
 - c. Un sprite
 - d. Le comportement en cas de dégâts (tir, collision)
- ii. Structure et données des niveaux décrits et stockés dans une base de données relationnelle
- c. Fonctionnalités
 - i. Au moins 2 niveaux implémentés avec
 - 1. Joueur
 - 2. Ennemis
 - 3. Obstacles

Gestion des highscores (en base de données)

Il est important de noter que certains de ces objectifs ont été écartés du projet ou modifiés, du à l'absence de base de données.

1.3 Gestion de projet

Pour la gestion de ce projet nous avons utilisé IceScrum, un outil qui permet de planifier et de gérer des projets de façon agile. Nous avons fait ce projet en utilisant un seul sprint sur toute la durée du projet.

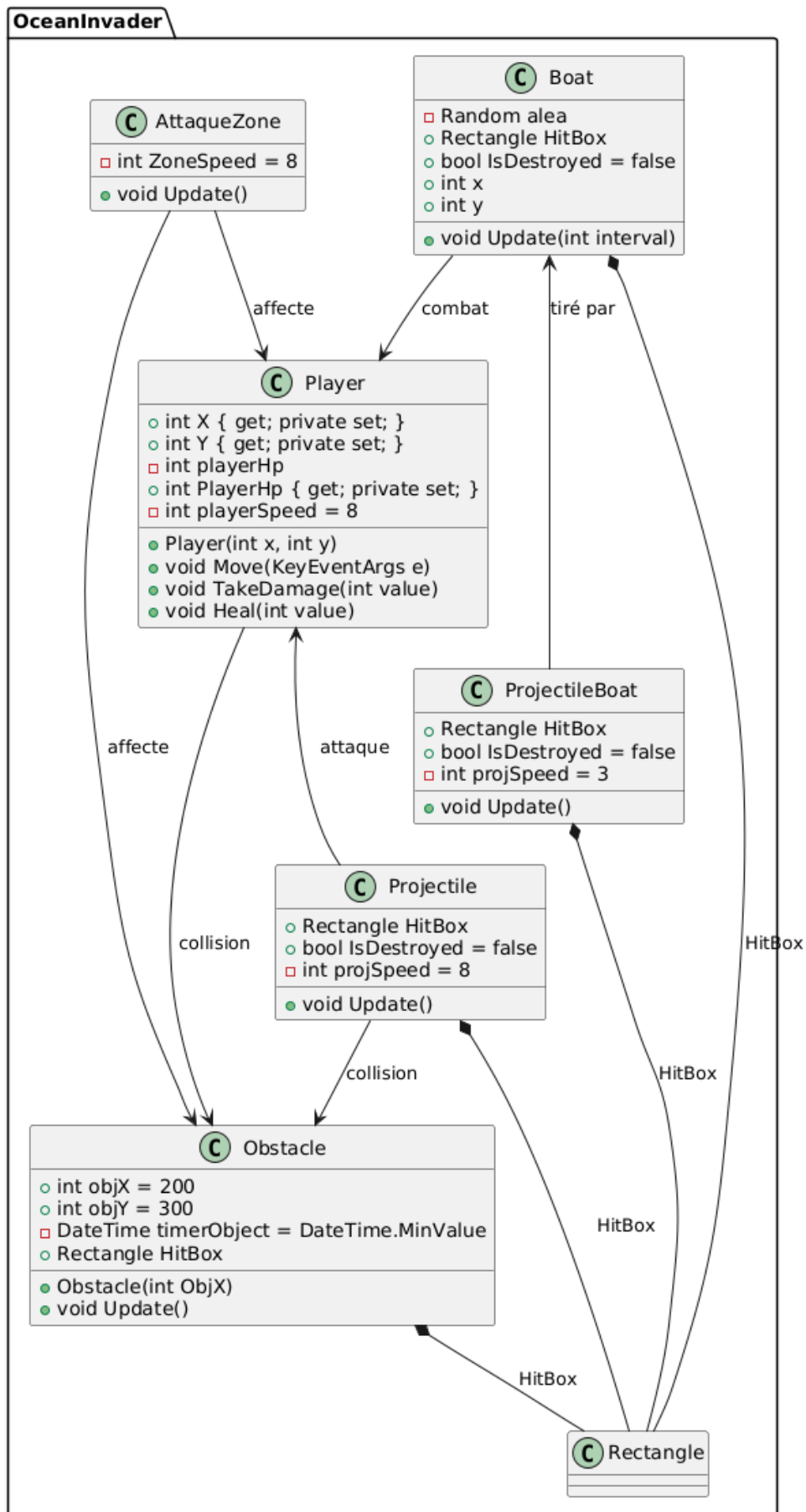
2 Analyse / Conception

2.1 Gameplay

- Le joueur se trouve dans une zone vers le bas de l'écran et doit empêcher les ennemis qui arrive du haut vers le bas d'entrer dans sa zone.
- Le joueur peut se déplacer comme bon lui semble dans la sa zone mais ne peux pas en sortir.
- Le joueur a deux types d'attaque, un tir normal qui fera de légers dégâts aux ennemis, et un tir faisant des dégâts de zone, infligeant de lourds dégâts aux ennemis sur une grande zone. L'attaque de zone possède un cooldown de 10 secondes entre chaque nouvelle attaque.
- Les ennemis tir devant eux pour attaquer le joueur.
- Le but du jeu est de survivre le plus longtemps possible afin de faire le score le plus haut.
- La partie se termine lorsque le joueur n'a plus de points de vie, ou si un ennemi est entré dans la zone du joueur.
- Il y a des obstacles en mouvement devant la zone du joueur qui lui permettent de se protéger contre les tirs des bateau ennemis.

L'utilisateur du jeu a le choix entre plusieurs niveaux de difficulté différentes. Entre le niveau le plus bas et le niveau le plus haut, la vitesse d'apparition des ennemis, la cadence de tir et les dégâts des projectiles ennemis augmentent, pour rendre les niveaux de plus en plus difficiles.

2.2 Concept



2.3 Analyse fonctionnelle

2.3.1 Mouvement (joueur)

(Auteur: Nicola Golaz)

En tant que joueur, Je veux pouvoir me déplacer de gauche à droite, en avant et en arrière, afin de naviguer à travers la mer et éviter ou attaquer les ennemis.

Tests d'acceptance:	
direction	Quand je presse sur les touches de direction (wasd/flèche directionnel), L'avatar se déplace en haut, en bas et de droite a gauche en fonction de la touche pressée.
zone de mouvement	Quand je me déplace avec mon avatar, Une zone de mouvement est délimiter par la zone verte (image ci joint).

2.3.2 Point de vie (joueur)

(Auteur: Nicola Golaz)

En tant que joueur, je veux commencer chaque partie avec 10 points de vie, afin de pouvoir résister aux attaques des ennemis pendant un certain temps.

Tests d'acceptance:	
Recevoir des dégâts 1	Quand je me fait toucher par les tir d'un ennemi de niveau 1, Je perd un point de vie.
Recevoir des dégâts 2	Quand je me fait toucher par les tir d'un ennemi de niveau 2, Je perd deux point de vie.
Recevoir des dégâts 3	Quand je me fait toucher par les tir d'un ennemi de niveau 3, Je perd trois point de vie.

2.3.3 Attaque (joueur)

(Auteur: Nicola Golaz)

En tant que joueur, je veux pouvoir tirer des attaques, afin de pouvoir éliminer des ennemis plus faibles d'un seul coup.

Tests d'acceptance:	
Attaque normal	Quand j'appuie sur le click gauche de ma souris, J'effectue une attaque normal qui inflige 1 de dégât.
Attaque de zone 1	Quand j'appuie sur le click droit de ma souris, J'effectue une attaque qui s'étend sur une zone défini et qui inflige de gros dégât.
Attaque de zone 2	Après avoir utilisé l'attaque de zone, J'ai un cooldown de 10 seconde.

2.3.4 Régénération de point de vie (joueur)

(Auteur: Nicola Golaz)

En tant que joueur, je veux récupérer des points de vie après avoir coulé un certain nombre de bateaux ennemis, afin de prolonger ma durée de survie et continuer à combattre.

Tests d'acceptance:	
Régénération bateau niveau 2	Pour chaque bateau de niveau 2 coulé, Je récupère 1 point de vie.

Régénération bateau de niveau 3	Quand je coule un bateau de niveau 3, Je récupère 2 point de vie.
Limite de point de vie	Quand mes point de vie sont au maximum (10/10), Je ne peux pas récupérer de point de vie supplémentaire.

2.3.5 Obstacle

(Auteur: Nicola Golaz)

En tant que joueur, Je veux que des obstacle se trouve sur la map, afin de pouvoir se protéger des ennemie.

Tests d'acceptance:
 collision Quand les obstacles reçoivent des tir du joueur ou des ennemie, Ils les bloquent.
 position Dans le jeu, Les obstacles sont juste devant la zone verte (voir draw.io).

2.3.6 Perdre

(Auteur: Nicola Golaz)

En tant que joueur, Je veux que la partie se termine quand les ennemie passe derrière le joueur, afin de définir une fin a une partie.

Tests d'acceptance:
 mort 1 Quand des ennemies passe derrière une certaine limite (voir image), Le joueur a perdu et la partie s'arrête.
 mort 2 Quand l'avatar du joueur n'a plus de point de vie, Le joueur a perdu et la partie s'arrête.

2.3.7 Ennemi comportement

(Auteur: Nicola Golaz)

En tant que joueur, Je veux que les bateau ennemis ai un comportement (déplacement/attaque), Pour que le jeu ai une certaine difficulté.

Tests d'acceptance:
 Déplacement 1 Quand je lance une partie, les bateau ennemie arrive depuis le haut de l'écran et se dirige vers le bas.
 Déplacement 2 Quand je lance une partie, Les bateau ennemie se déplacent de gauche a droite de façon aléatoire.
 Mur gauche et droite Quand je suis dans une partie, Dès que les ennemis se heurtent a une parois (gauche/droite) ils repartent dans la direction opposé.

2.3.8 Menu

(Auteur: Nicola Golaz)

En tant que joueur, Je veux un menu, Pour pouvoir naviguer entre les différentes fonctionnalité du jeu.

Tests d'acceptance:
 Editeur de niveau Quand je suis dans le menu, Je peux cliquer sur des boutons pour choisir entre trois niveau différent.
 Jouer Quand je suis dans le menu, Il y a un bouton qui me permet de quitter le jeu.

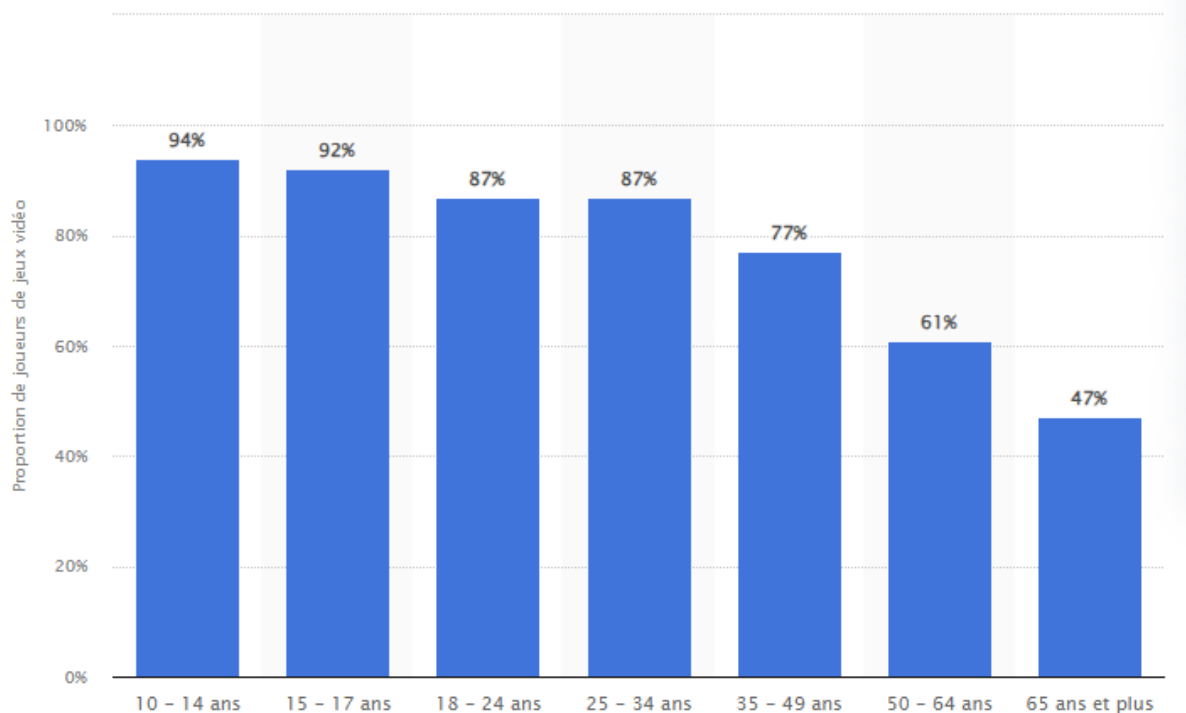
2.4 UX

Conception centrée utilisateur

Création des personnas :

Après avoir effectué une recherche sur internet, j'ai constaté que la tranche d'âge qui passe le plus de temps sur les jeux vidéo est celle entre 10 et 14 ans. Le premier persona que je vais créer aura donc 13 ans.

Pour mon deuxième persona, j'ai choisi une femme de 27 ans pour couvrir une large tranche d'âge.



[Part de Français jouant aux jeux vidéo par âge 2023 | Statista](#)

J'ai choisi l'âge de ces deux persona pour pouvoir couvrir la tranche d'âge où les proportions de joueurs sont les plus élevées.

Les deux persona ont des caractéristiques différentes, pour pouvoir couvrir des types de population différents. Par exemple, l'un a des moyens financiers faibles, et l'autre nettement plus élevés.

Persona 1



BIO

- Prénom/Nom : Pierre Lemoine
- Age : 13 ans
- Métier : Etudiant
- Famille : Célibataire

Aisance numérique ● ● ● ● ●

Fréquence d'usage ● ● ● ● ●

Moyen financier ● ● ● ● ●

Histoire

Pierre a 13 ans et passe la plupart de ses après-midis à jouer aux jeux vidéo, s'évadant dans des mondes fantastiques où il se sent libre et compétent. Ses parents, inquiets de le voir autant devant l'écran, lui demande souvent de réduire son temps de jeu. Pierre tentait de trouver un équilibre entre sa passion pour les jeux et les attentes de sa famille. Malgré cela, il continuait à rêver de devenir un joueur de jeux vidéo professionnel un jour.

But

- S'amuser
- Faire le score le plus élevé
- Personnaliser les niveaux en fonction pour pouvoir s'entraîner.

Personnalité

Impatient

Attentes fonctionnelles

Pouvoir se déplacer facilement dans les différents menus et avoir accès à un jeu fluide.

Personna 2



BIO

- Prénom/Nom : Elise Moreau
- Age : 27 ans
- Métier : Serveuse
- Famille : Célibataire

Aisance numérique ● ● ● ● ●

Fréquence d'usage ● ● ● ● ●

Moyen financier ● ● ● ● ●

Histoire

Élise, 27 ans, travaille comme serveuse dans un café animé, où elle aime l'ambiance vivante et les échanges avec les clients. Le soir, elle plonge dans son univers de jeux vidéo, où elle se passionne pour les quêtes épiques et les stratégies complexes. Cette double vie lui permet de mêler interaction sociale et immersion virtuelle. Son équilibre entre le monde réel et le monde numérique fait de ses journées une aventure captivante.

But

- Se détendre et s'amuser
- Améliorer son attention
- Devenir meilleure stratégiquement

Personnalité

Intelligente, Calme, Extravertie

Attentes fonctionnelles

Avoir accès plusieurs niveaux pour s'entraîner sur différentes difficultés. Pouvoir modifier les niveaux à sa guise.

Choix de la palette graphique

Palette de couleur :

Yale Blue: 1B3B5F

Moonstone: 53A2BE

Aquamarine: 9FFFCB



J'ai choisi ces couleurs car représente bien le thème du jeu (L'océan), et quelle créer un bon contraste ce qui permet une meilleurs visibilité des éléments.

Palette de gris :

Gunmetal: 31393C

Ash gray: AFBFC0

Cadet gray: 9FA4A9



Eco-conception

La palette de couleur est composée de peu de couleur, ce qui permet de d'alléger les textures et les éléments visuel, réduisant ainsi la consommation d'énergie lié au rendu graphique.

L'affichage des High-Score et la sélection des niveaux se fait sur la même page, ce qui permet de charger une page en moins et ainsi d'économiser de l'énergie.

Accessibilité

Chaque bouton et chaque section est étiquetée avec des labels explicites, qui permettent à l'utilisateur de comprendre facilement et rapidement l'utilité du bouton ou de la section en question.

Les boutons relativement grands, permettent aux utilisateurs comme des personnes âgées de pouvoir utiliser les différentes fonctionnalités aisément.

Conception

Définition de tous les écrans (WireFrame) :

- Ecran d'accueil
- Gestionnaire de niveau/high score
- Ecran de jeu (dans un niveau)
- Ecran éditeur de niveau

High-fidelity :

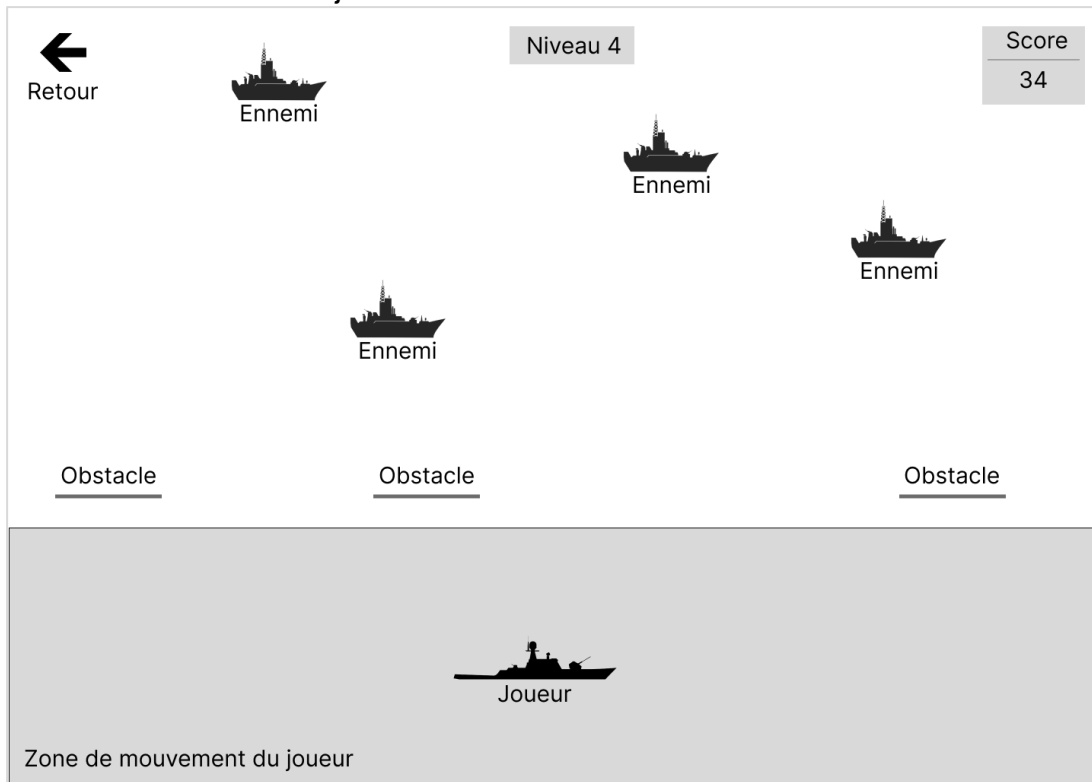
- Ecran éditeur de niveau

WireFrame – Menu Principal



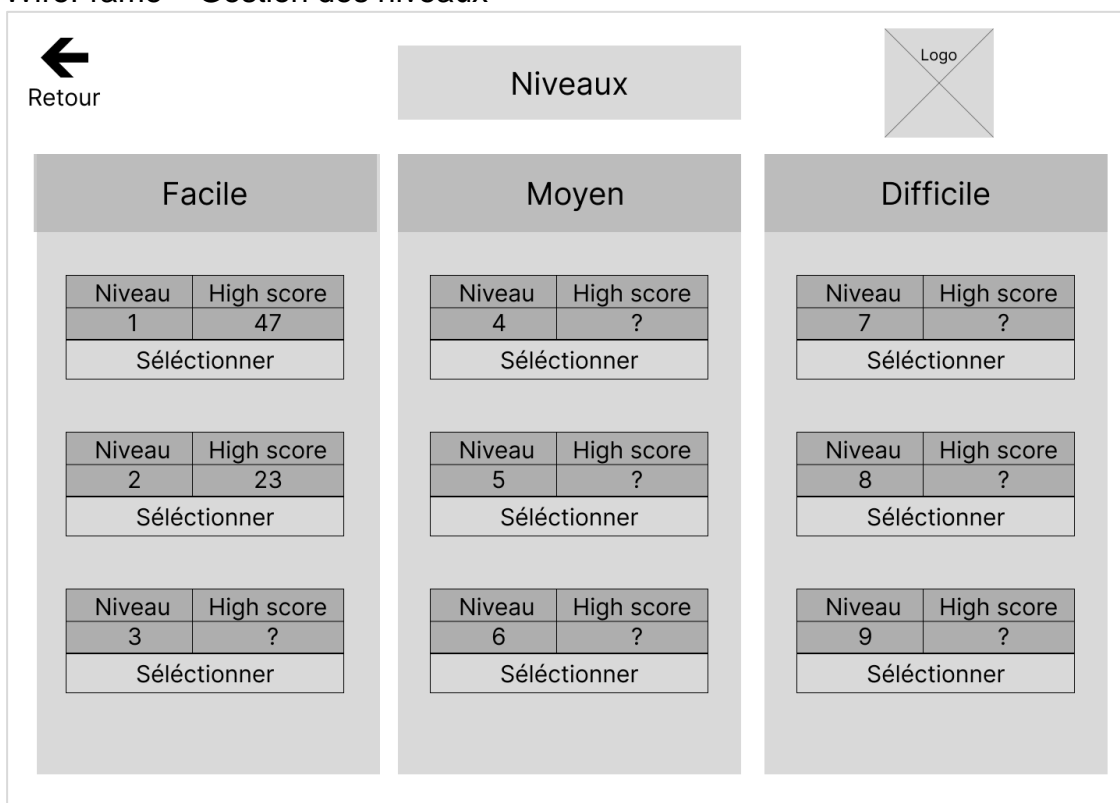
Cet écran permet de choisir entre les différentes fonctionnalités du jeu, comme lancer une partie ou choisir le niveau.

WireFrame – Ecran de jeu



Cet écran représente ce que l'utilisateur verra après avoir lancé une partie.

WireFrame – Gestion des niveaux



Sur cet écran, l'utilisateur peut choisir entre les niveaux, et voir les HighScore.

WireFrame – Editeur de niveau

The wireframe illustrates the 'Editeur de niveau' (Level Editor) interface. It features a top navigation bar with a back arrow, the title 'Editeur de niveau', and a 'Sauvegarder' (Save) button. The main area is divided into three vertical panels: 'Joueur' (Player), 'Ennemis' (Enemies), and 'Obstacle'.

Joueur (Player) Panel:

- Point de vie: A slider control.
- Vitesse de déplacement: A slider control.
- Dégâts attaque 1, 2, 3: Three separate slider controls.
- Taille de la zone de mouvement: A slider control with a value of 'Y' and a maximum of '400'.
- Reinitialiser: A button at the bottom.

Ennemis (Enemies) Panel:

- Point de vie: A slider control.
- Vitesse de déplacement: A slider control.
- Mouvement aléatoire (gauche/droite): Two buttons labeled 'Non' and 'Oui'.
- Dégâts par attaque: A slider control.
- Vitesse d'apparition: Two buttons labeled 'Basse' and 'Elevée'.
- Reinitialiser: A button at the bottom.

Obstacle Panel:

- Nombre d'obstacle: A slider control with a value of '5'.
- Longueur de l'obstacle: A slider control.
- Image: A placeholder box with a diagonal cross and the text 'Image'.
- Mouvement (gauche/droite): Two buttons labeled 'Non' and 'Oui'.
- Reinitialiser: A button at the bottom.

HighFidelity – Editeur de niveau

The screenshot shows the 'Editeur de niveau' (Level Editor) interface. It features a dark blue header with a back arrow, the title 'Editeur de niveau', and a green 'Sauvegarder' (Save) button. The main area is divided into three columns: 'Joueur' (Player), 'Ennemis' (Enemies), and 'Obstacle' (Obstacle). Each column contains several adjustable parameters with sliders, checkboxes, and buttons. The 'Joueur' column includes 'Point de vie' (Health), 'Vitesse de déplacement' (Movement speed), 'Dégâts attaque 1, 2, 3' (Attack damage), and 'zone de mouvement' (Movement zone). The 'Ennemis' column includes 'Point de vie', 'Vitesse de déplacement', 'Mouvement aléatoire (gauche/droite)' (Random movement), 'Dégâts par attaque' (Damage per attack), and 'Vitesse d'apparition' (Appearance speed). The 'Obstacle' column includes 'Nombre d'obstacle' (Number of obstacles), 'Longueur de l'obstacle' (Obstacle length), an 'Aperçu' (Preview) of the obstacle, and 'Mouvement (gauche/droite)' (Movement). Each column has a 'Reinitialiser' (Reset) button at the bottom.

L'écran 'Editeur de niveau', permet à l'utilisateur de modifier tous les niveaux du jeu en fonction de ses envies.

Choix effectué (High-Fidelity) :

Le joueur peut choisir entre trois catégories : Joueur, Ennemis et Obstacle.

J'ai gardé la police de base (Inter) pour faire mon high-fidelity, car c'est une police sans-serif ce qui la rend facile à lire. Cela conviendra aux joueurs les plus âgés, mais aussi au plus jeune.

J'ai utilisé une police d'écriture plutôt grande, pour proposer une bonne visibilité des paramètres au joueur.

J'ai choisi d'afficher un aperçu de l'obstacle, pour permettre au joueur de mieux se représenter la longueur.

2.5 Stratégie de test

Tests fonctionnels : vérifie que toutes les fonctionnalités principales du jeu répondent aux exigences (Gameplay).

Tests unitaires : Vérifie séparément que chaque composant du jeu (mouvement, tir, dégâts) fonctionne comme prévu.

Moyen à mettre en œuvre : outil de test automatisé, comme MStest. Des personnes externes ou interne au projet pour tester les fonctionnalités du programme.

3 Réalisation

- Cooldown de l'attaque de zone :
Le cooldown utilise DateTime.Now pour déterminer si le temps entre la dernière attaque et le temps présent est plus petit ou plus grand que 10. Si le temps est plus petit, le temps restant avant la prochaine attaque est affiché, si il est plus grand ou égal a 10, l'attaque peut être exécutée.
- Gestion des listes :
Lorsque je dois parcourir mes listes d'objet pour pouvoir ensuite les modifier, je ne peux pas supprimer un objet de la liste pendant que je la parcours avec une boucle 'foreach', car cela crée une erreur. C'est pourquoi, pour régler ce problème, j'ai décidé d'utiliser des copies des listes, de marquer les éléments à supprimer, et de les supprimer une fois sortie de la boucle.

Exemple :

```
foreach (ProjectileBoat projectileBoat in new List<ProjectileBoat>(projectileBoats))
{
    projectileBoat.Render(airspace);
    if (projectileBoat.projBoatY > 600)
    {
        projectileBoat.IsDestroyed = true;
    }

    foreach (Player player in players)
    {
        // Condition pour voir si le projectile doit être détruit
        if (projectileBoat.HitBox.Intersects(player.HitBox))
        {
            player.TakeDamage(Program.niveau);
            projectileBoat.IsDestroyed = true;
        }
    }
}
projectileBoats.RemoveAll(b => b.IsDestroyed); // Destruction des projectiles
```

3.1 Déroulement

Je n'ai pas eu de problème pour permettre au joueur de se déplacer, j'ai essayé de trouver comment faire pour que le joueur ne se déplace pas de façon saccadé (par accoup) mais je n'ai pas réussi.

Certaines story comme le fait de tirer un projectile au click m'ont pris beaucoup de temps à réaliser car je n'avais pas les connaissances requises pour réaliser cette fonction.

Le reste des story c'est déroulé sans problème.

3.2 Mise en place de l'environnement de travail

Vous pouvez trouver le code source de mon jeu dans mon repos github :

[NicolaGolaz/P_OceanInvader: db](#)

Vous y trouverez le dossier « OceanInvader », qui contient tous les fichiers qui composent mon jeu.

Le projet a été réalisé sur Windows 11 et Windows 10. La version de Windows 11 est 23H2. L'IDE utilisé est Visual Studio 2022.

Les seuls matériels utilisés sont deux PC (clavier, souris, etc.).

Le programme ne fonctionne pas bien sur Windows 11.

3.3 Description des tests effectués

3.3.1 Sprint 1

3.3.1.1 Menu

Editeur de niveau	Quand je suis dans le menu, Je peux cliquer sur des boutons pour choisir entre trois niveaux différents.	OK 30 Oct
Jouer	Quand je suis dans le menu, Il y a un bouton qui me permet de quitter le jeu.	OK 30 Oct

3.3.1.2 Point de vie (joueur)

Recevoir des dégâts 1	Quand je me fais toucher par le tir d'un ennemi de niveau 1, Je perds un point de vie.	OK 30 Oct
Recevoir des dégâts 2	Quand je me fais toucher par le tir d'un ennemi de niveau 2, Je perds deux points de vie.	OK 30 Oct
Recevoir des dégâts 3	Quand je me fais toucher par le tir d'un ennemi de niveau 3, Je perds trois points de vie.	OK 30 Oct

3.3.1.3 Perdre

mort 1	Quand des ennemis passent derrière une certaine limite (voir image), Le joueur a perdu et la partie s'arrête.	OK 30 Oct
mort 2	Quand l'avatar du joueur n'a plus de points de vie, Le joueur a perdu et la partie s'arrête.	OK

		30 Oct
3.3.1.4 Régénération de point de vie (joueur)		
Régénération bateau niveau 2	Pour chaque bateau de niveau 2 coulé, Je récupère 1 point de vie.	OK 30 Oct
Régénération bateau de niveau 3	Quand je coule un bateau de niveau 3, Je récupère 2 point de vie.	OK 30 Oct
Limite de point de vie	Quand mes point de vie sont au maximum (10/10), Je ne peux pas récupérer de point de vie supplémentaire.	OK 30 Oct
3.3.1.5 Obstacle		
collision	Quand les obstacles reçoivent des tir du joueur ou des ennemie, Ils les bloquent.	OK 30 Oct
position	Dans le jeu, Les obstacles sont juste devant la zone verte (voir draw.io).	OK 30 Oct
3.3.1.6 Attaque (joueur)		
Attaque normal	Quand j'appuie sur le click gauche de ma souris, J'effectue une attaque normal qui inflige 1 de dégât.	OK 9 Oct
Attaque de zone 1	Quand j'appuie sur le click droit de ma souris, J'effectue une attaque qui s'étend sur une zone défini et qui inflige de gros dégât.	OK 9 Oct
Attaque de zone 2	Après avoir utilisé l'attaque de zone, J'ai un cooldown de 10 seconde.	OK 9 Oct
3.3.1.7 Mouvement (joueur)		
direction	Quand je presse sur les touches de direction (wasd/flèche directionnel), L'avatar se déplace en haut, en bas et de droite a gauche en fonction de la touche pressée.	OK 25 Sep
zone de mouvement	Quand je me déplace avec mon avatar, Une zone de mouvement est délimiter par la zone verte (image ci joint).	OK 25 Sep

3.4 Erreurs restantes

Il ne reste plus d'erreur dans le code, il se compile et l'on peut jouer au jeu.

4 Conclusions

J'ai réussi à atteindre la majeure partie de mes objectifs que j'avais défini au début du projet. J'ai eu des difficultés à comprendre certain concept comme les 'hitbox', mais une fois le concept acquis j'ai pu aisément l'appliquer au reste de mon code. La création des story et des test d'acceptances m'a pris beaucoup de temps, j'aurais préféré avoir plus de temps pour coder et faire le rapport. Si je pouvais améliorer ce projet, j'optimiserais la gestion de la mémoire, car le jeu demande de plus en plus de mémoire au fur et a mesure de la partie.

5 Annexes

5.1 Journal de travail

Vous trouverez mon JDT dans mon repository github.