

ZombiesApocalypse

Pour projet Shoot Me Up avec XCL

Table des matières

<u>1</u>	<u>Analyse préliminaire</u>	3
1.1	<u>Introduction</u>	3
1.2	<u>Objectifs</u>	3
1.3	<u>Planification initiale</u>	4
<u>2</u>	<u>Analyse / Conception</u>	4
2.1	<u>Concept</u>	4
2.2	<u>Stratégie de test</u>	4
2.3	<u>Risques techniques</u>	4
2.4	<u>Planification</u>	4
2.5	<u>Dossier de conception</u>	5
<u>3</u>	<u>Réalisation</u>	5
3.1	<u>Dossier de réalisation</u>	5
3.2	<u>Description des tests effectués</u>	6
3.3	<u>Erreurs restantes</u>	6
3.4	<u>Liste des documents fournis</u>	6
<u>4</u>	<u>Conclusions</u>	6
<u>5</u>	<u>Annexes</u>	7
5.1	<u>Résumé du rapport du TPI / version succincte de la documentation</u>	7
5.2	<u>Sources – Bibliographie</u>	7
5.3	<u>Journal de travail</u>	7
5.4	<u>Manuel d'Installation</u>	7
5.5	<u>Manuel d'Utilisation</u>	7
5.6	<u>Archives du projet</u>	7

NOTE L'INTENTION DES UTILISATEURS DE CE CANEVAS :

Toutes les parties en italiques sont là pour aider à comprendre ce qu'il faut mettre dans cette partie du document. Elles n'ont donc aucune raison d'être dans le document final.

De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n'aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l'alourdir inutilement.

- **Analyse préliminaire**

- **Introduction**

Ce projet est basé sur le module 320 (Programmation orienté objet). Le but est de faire un jeu vidéo du type SpaceInvaders mais personnalisé. Un jeu vidéo est une très bonne manière d'apprendre à programmer orienté objet car dans le monde du jeu vidéo c'est plus facile l'implémenter. C'est aussi une bonne façon pour que les élèves prennent un peu plus de plaisir que de faire des applications qui n'ont pas d'interaction.

- **Objectifs**

- 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

- ii. Gestion des highscores (en base de données)

-

- **Gestion de projet**

Les outils employés incluent IceScrum, qui est utilisé pour appliquer la méthode de gestion de projet Scrum, et GitHub, qui sert à mettre en place un système de sauvegardes afin de prévenir toute perte de fichiers.

- **Analyse / Conception**

- **Analyse fonctionnel**

tir(normal)

(Auteur: Eithan Sanchez Filipe)

en tant que joueur je veux pouvoir tirer avec un pistolet sur les zombies afin de les tuer	
Tests d'acceptance:	
modèle	dans la partie un model de balle est présent a chaque tir
déplacement	dans une partie quand mon personnage tir la balle se déplace tout droit
collision	quand ma balle touche un zombie ou la bordure elle disparaît
cadence de tir	quand je tir une balle la deuxième mets ~1s a partir (dépend des améliorations)
dégât	quand la balle touche un zombie elle inflige 7 point de dégâts

tir(sécial)

(Auteur: Eithan Sanchez Filipe)

en tant que joueur je veux pouvoir tirer avec une mitraillette sur les zombies afin de les tuer	
Tests d'acceptance:	
modèle	dans la partie un model de balle est présent a chaque tir
déplacement	dans une partie quand mon personnage tir la balle se déplace tout droit
collision	quand ma balle touche un zombie ou la bordure elle disparaît
cadence de tir	quand je tir une balle la deuxième mets ~0.4s a partir (dépend des améliorations)
dégâts	quand la balle touche un zombie elle inflige 10 point de dégâts

zombies

(Auteur: Eithan Sanchez Filipe)

en tant que joueur je veux que des zombies apparaissent sur la map aléatoirement	
Tests d'acceptance:	
apparition	quand le jeu se lance les zombies apparaissent de manière aléatoire
but	quand je lance le jeu leurs but est d'arriver tout en bas de la fenêtre
modèle	quand je lance le jeu le zombie a un modèle

Vie(personnage)

(Auteur: Eithan Sanchez Filipe)

en tant que développeur je veux que le personnage perde de la vie

Tests d'acceptance:	
barre de vie	quand je lance le jeu la barre de vie de notre personnage apparait
Dégât	quand le personnage subit des dégâts il perd de la vie(la barre de vie diminue)
position	quand je regarde le haut de la fenêtre j'y vois la barre de vie qui est rouge
vie	le personnage a 25 points de vie

Déplacement joueur

(Auteur: Eithan Sanchez Filipe)

En tant que joueur je veux pouvoir me déplacé	
Tests d'acceptance:	
déplacements	dans une partie quand je presse sur A,D le personnage bouge dans la direction de la touche pressé
bordure	dans une partie quand je vais sur le bord de la fenêtre avec le personnage le personnage se bloque donc il ne peut pas déplacer la fenêtre

attaque zombie

(Auteur: Eithan Sanchez Filipe)

en tant que joueur je veux que les zombies tapent les obstacles et le bord de la map	
Tests d'acceptance:	
bord	quand le zombie atteint le bord bas de la map il enleve de la vie et disparaît
obstacle	quand le zombie atteint un obstacle il le tape jusqu'à ce qu'il se casse
dégât	quand il tape quelque chose il lui enlève 5 PV

vie(zombie)

(Auteur: Eithan Sanchez Filipe)

en tant que joueur je veux pouvoir infliger des dégâts aux zombies	
Tests d'acceptance:	
perte de vie	quand un balle touche le zombie il perd de la vie

vie de base	quand je lance le jeu les zombies ont 20 PV
-------------	---

Personnage

(Auteur: Eithan Sanchez Filipe)

en tant que joueur je veux un personnage afin de pouvoir le contrôler et jouer	
Tests d'acceptance:	
apparition	quand le jeu se lance le personnage apparait en bas au milieu de l'écran
modèle	quand le personnage apparait il a un modèle d'un soldat il change en fonction de son type d'attaque

niveau

(Auteur: Eithan Sanchez Filipe)

en tant que joueur je veux pouvoir changer de niveau afin de continuer de jouer	
Tests d'acceptance:	
changement de niveau	quand il ne reste plus de zombies dans le niveau le niveau change
nombre de zombies	quand le jeu change de niveau le nombre de zombies grandit
Texte	quand le jeu change de niveau un texte en disant le nouveau niveau apparait

Barrière

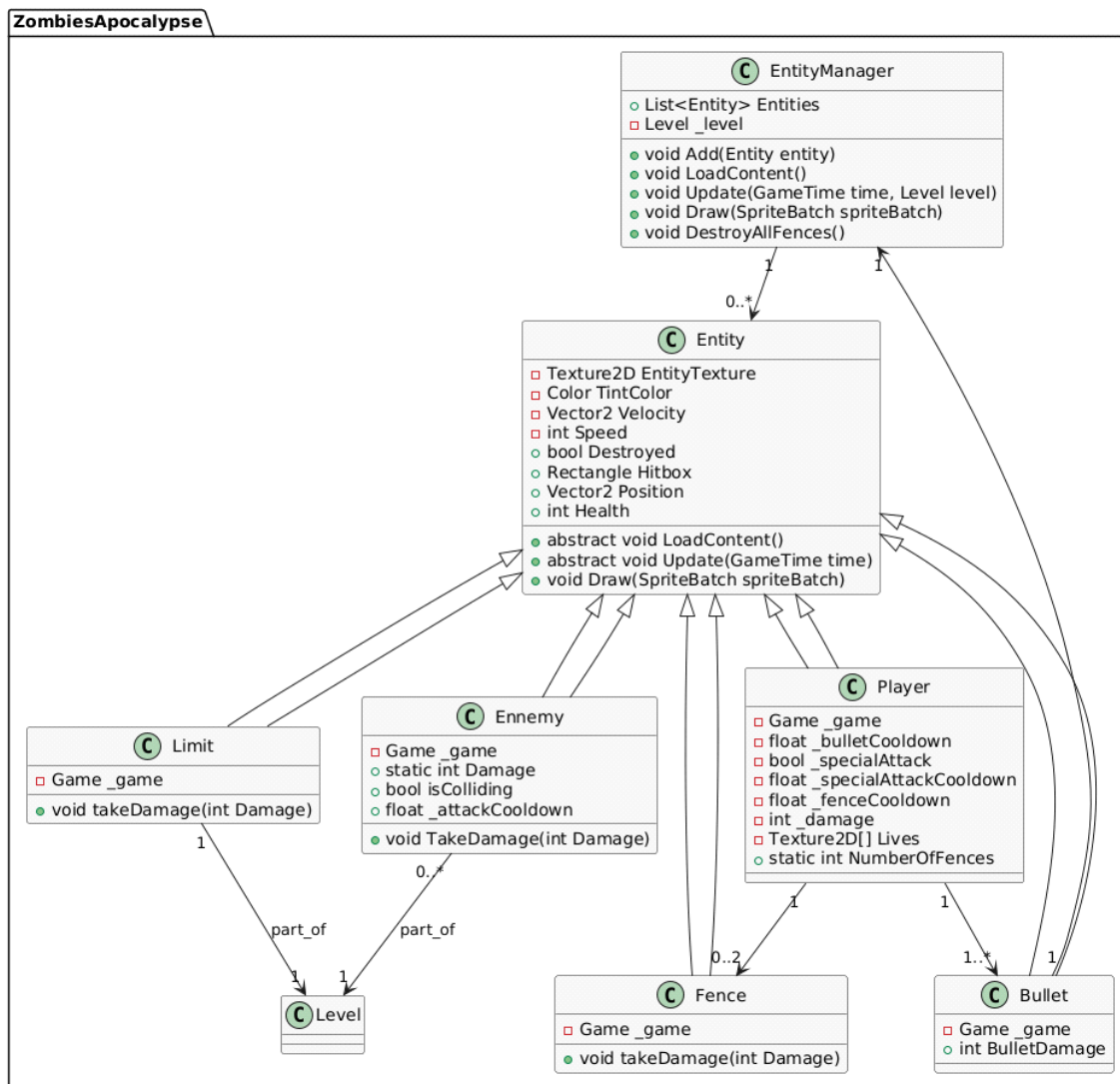
(Auteur: Eithan Sanchez Filipe)

en tant que joueur je veux pouvoir placer des barrières	
Tests d'acceptance:	
placement	quand le joueur presse F une barrière apparait
vie	les barrières ont 10 points de vie
collision	les zombies ne peuvent pas les traverser mais les balles oui

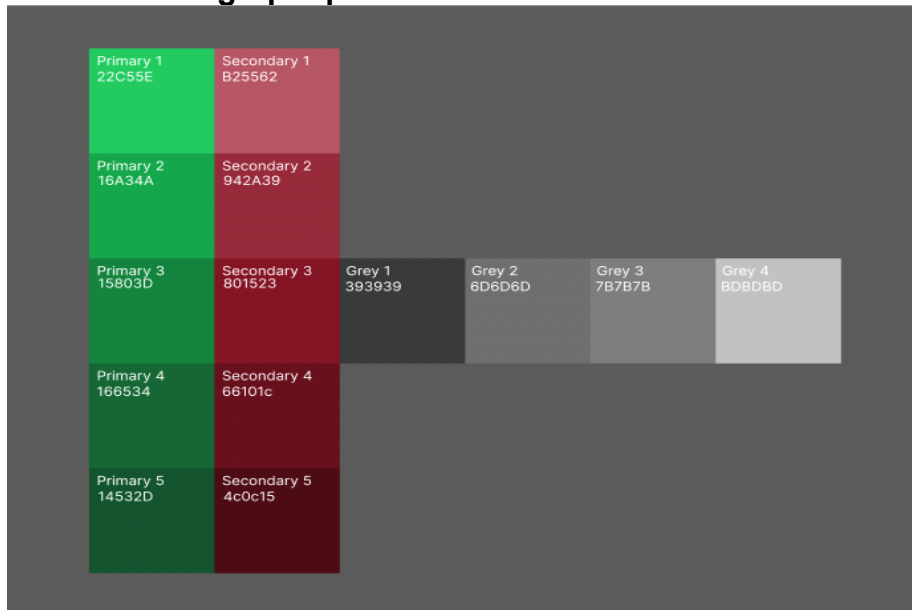
- **Gameplay**

Dans le jeu *Zombies Apocalypse* le joueur incarne un soldat qui doit se défendre contre des vagues successives de zombies tout en protégeant des fortifications précieuses. En posant des barrières de protection et en utilisant plusieurs types d'attaques, le joueur doit éliminer autant d'ennemis que possible pour survivre le plus longtemps possible. Le système de gestion d'entités du jeu contrôle la création la mise à jour et l'interaction de tous les éléments (le joueur, les zombies, les obstacles). Cette approche assure une expérience fluide et immersive qui augmente en intensité au fur et à mesure que le joueur progresse dans les niveaux.

- **Concept**



- **Analyse fonctionnelle**
- **UX**
- **Palette graphique**



Dans cette palette, le vert est la couleur principale de mes maquettes parce qu'il représente à la fois le soldat et le zombie. C'est une couleur qui marche bien pour ces deux personnages. Le vert renvoie la force pour le soldat, et le côté mort-vivant pour le zombie. Malheureusement, ce vert n'apparaît pas dans la maquette haute-fidélité, car c'était compliqué de l'ajouter sans réduire le contraste visuel.

J'ai aussi choisi un rouge foncé comme couleur secondaire, pour représenter le sang et donc le zombie.

Il est important que les couleurs se démarquent les unes des autres afin d'éviter que les personnes daltoniennes ou les personnes âgées puissent différencier les couleurs.

- **Eco-Conception / Accessibilité**

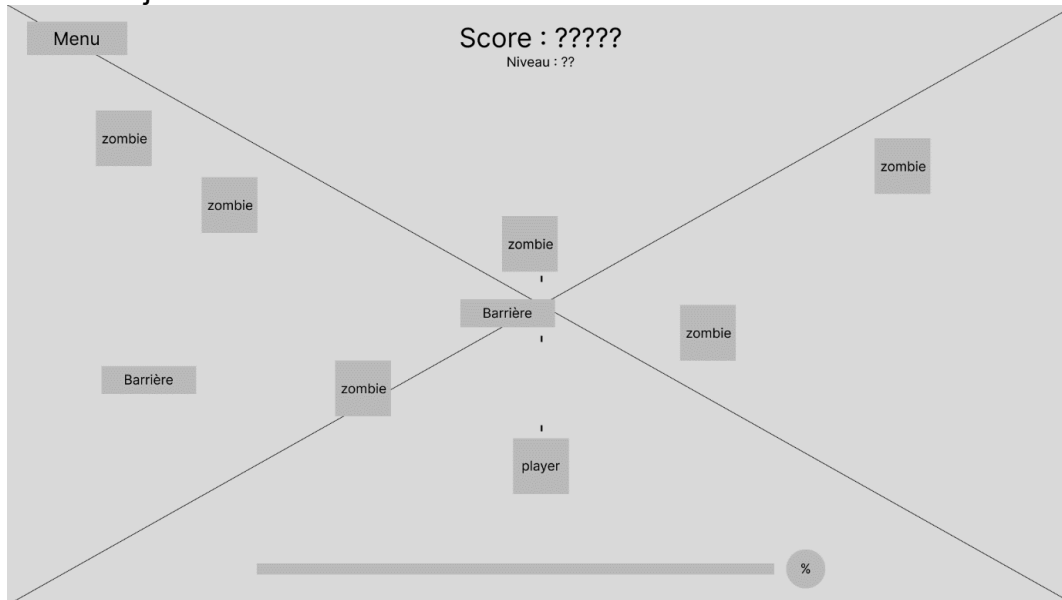
L'accessibilité en UX vise à concevoir des sites et applications utilisables par tous, y compris les personnes en situation de handicap. Ça prend en compte que les textes soient lisibles, des emplacements intuitifs et le fait que les technologies comme les lecteurs d'écrans puissent être utilisées sans problèmes. Dans mes maquettes l'accessibilité est présente dans les couleurs qui sont faciles à différencier et dans les emplacements des boutons.

L'éco-conception UX consiste à concevoir des expériences utilisateur en tenant compte de l'impact environnemental des produits numériques. Cela implique de réduire la consommation d'énergie, d'optimiser les performances des sites et applications, et de minimiser les ressources utilisées. Dans mon cas j'ai choisi des couleurs qui ne sont pas trop claires pour réduire la consommation de l'écran.

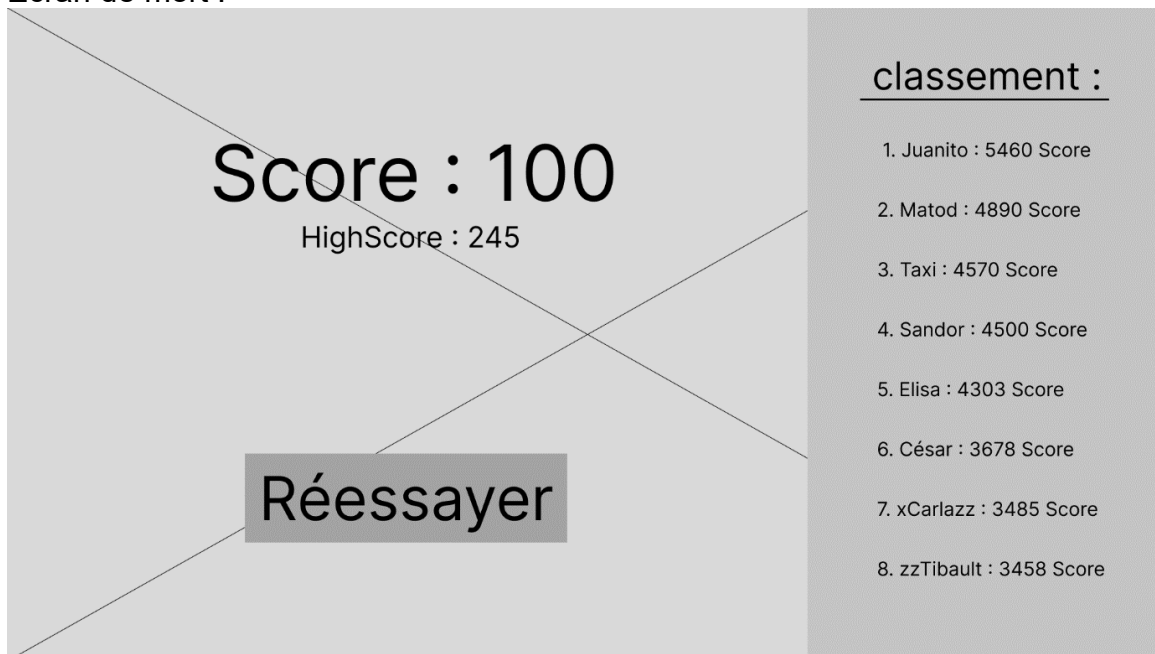
- **Définition des écrans wireframe**

Ce sont des esquisses simples qui montrent la structure générale des écrans. Le visuel ne rentre pas en compte (les couleurs, la typographie, etc..), le wireframe reprend les emplacements et les fonctionnalités principales. Dans mon projet j'ai les wireframes suivants :

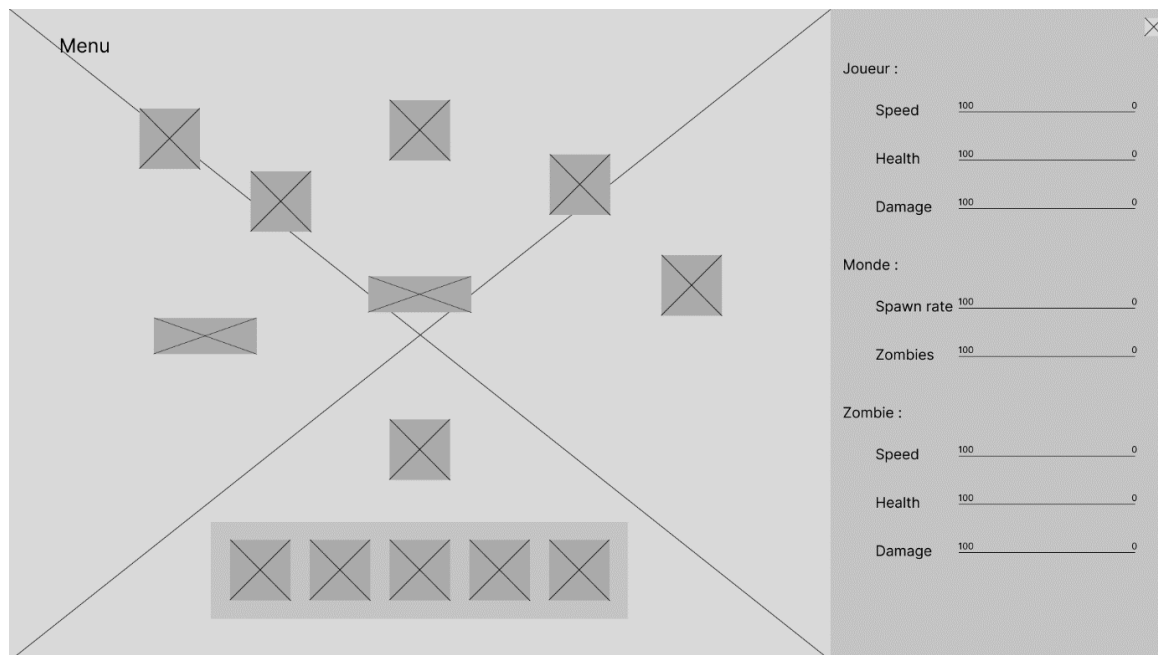
Écran de jeu :



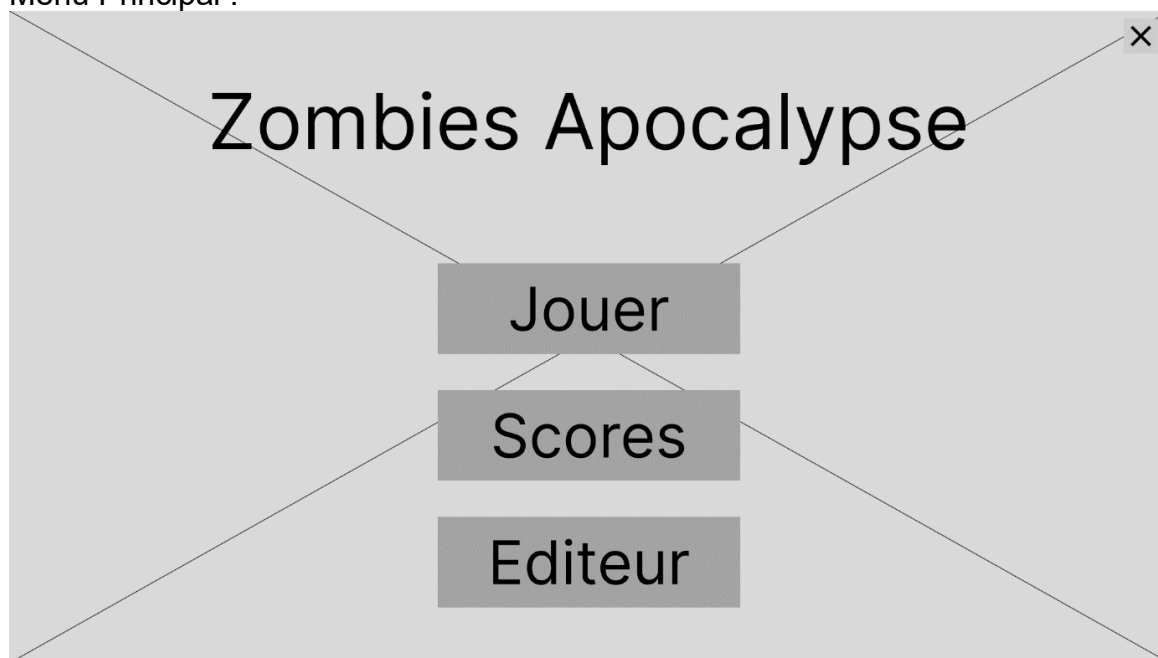
Écran de mort :



Éditeur de niveau :

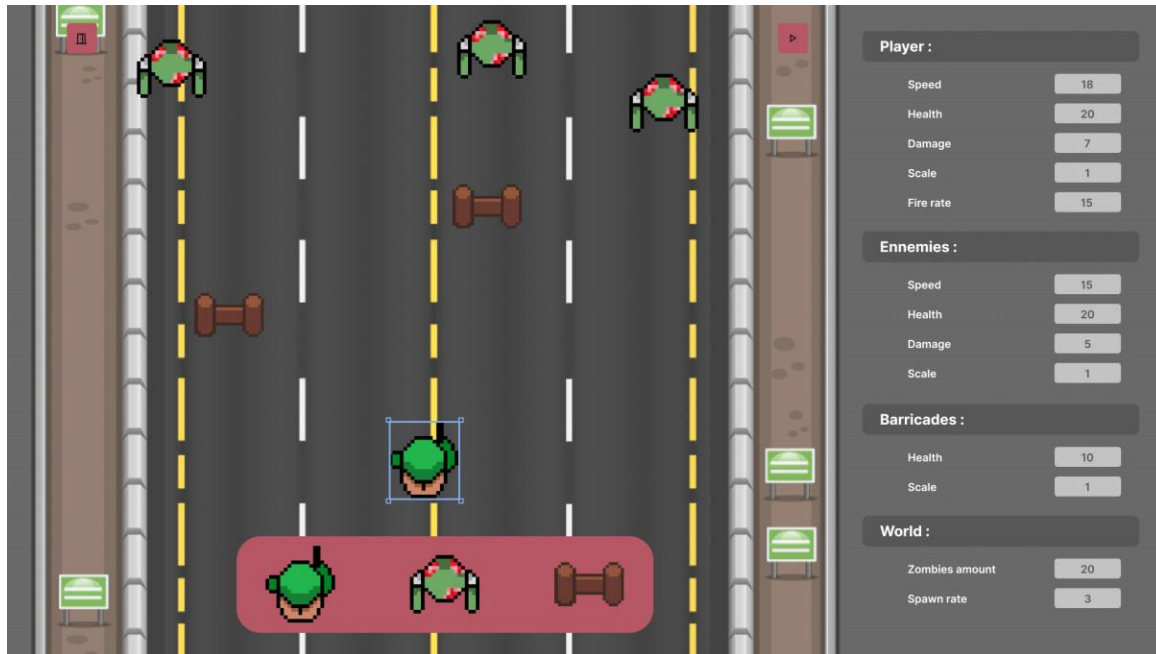


Menu Principal :



- **Définition du mockup**

Un mockup est une maquette qui représente fortement le produit final, comparé à un wireframe le mockup comporte plus de détails. Les emplacements restent les mêmes cependant les couleurs les typographies sont exactement celles qu'on va ajouter à notre produit final. Dans le projet il est demandé d'effectuer le mockup d'un écran, celui de l'éditeur de niveau.



- **Choix Effectuées sur le mockup**

- **Typographie et style visuel**

Pour la typographie j'ai opté pour une police sans-serif pour facilité la lecture et la rendre plus rapide. Le style visuel est très sobre avec peu de couleurs différentes afin de permettre la compréhension et ne pas encombrer l'écran

- **Organisation des éléments UX**

Dans le mockup on y voit 2 gros éléments l'un est la barre d'éléments en bas de l'écran et l'autre est les propriétés des éléments à droite. La barre d'éléments contient des objets qu'on peut placer sur le jeu. Les propriétés contiennent des catégories qui sont le joueur, ennemies, barricades et monde. Pour modifier ces propriétés il y a une zone de texte dans laquelle on peut modifier les valeurs afin de varier les différentes options

- **Stratégie de test**

Les tests seront réalisés dans un environnement de développement configuré sur Visual Studio 2022, version 17.9.1. Cette configuration permet d'exécuter et de déboguer le jeu de manière efficace.

Ces tests visent à vérifier le bon fonctionnement des composants individuels, tels que les classes et les méthodes du jeu (gestion des entités, collisions, contrôles du joueur, etc.).

Les tests seront faits sur:

- Un PC équipé de Visual Studio 2022 et de l'environnement MonoGame et MSTest

Les tests couvriront les fonctionnalités principales de l'application, mais ne seront pas exhaustifs pour les éléments mineurs ou peu impactants.

Pour tester le jeu deux personnes m'ont aidé, ces personnes sont Mathis Olaya et Elliott Scherrer

- **Réalisation**

Pour la réalisation je vais expliquer quelques points que j'ai trouvé intéressants.

Le premier est la gestion des entités car je ne pouvais pas supprimer des entités pendant que la liste où elles étaient se faisait parcourir. Exemple :

```
foreach (Entity entity in Entities.ToArray())
{
    if (entity.Destroyed)
    {
        Entities.Remove(entity);
        if (entity is Enemy zombie)
            _level.NumberOfZombies--;
    }
}
```

Le ToArray permet de transformer la liste en tableau et de parcourir le tableau ce qui fait donc qu'on ne modifie pas la liste pendant qu'elle est parcourue

Le deuxième est le fonctionnement pour restart le jeu. Pour ce faire j'ai créé un enum avec des gamestates

```
public enum GameState
{
    Playing,
    GameOver
}
```

Avec cela quand le joueur mourait le gamestate passait à gameover ce qui permettait au jeu de se relancer grâce à une méthode qui remettait tous les objets à 0 :

```
private void RestartGame()
{
    // Remets les Variables à 0
    _player = new Player(this);
    _level = new Level(this);
    _limit = new Limit(this);
    _gameState = GameState.Playing;
}
```

- **Déroulement**

Globalement chaque story s'est bien déroulée. Les fonctionnalités ont été ajoutées comme prévu et les tests se sont passés sans problème. En revanche lors de la refactorisation j'ai eu des problèmes à cause des sprites qui ne voulaient pas s'ajouter

- **Mise en place de l'environnement de travail**

- *Le code source est disponible sur GitHub. Pour le récupérer, clonez le dépôt et ouvrez la solution (ZombiesApocalypse.sln) dans Visual Studio 2022*
- *Le développement se fait sous Windows 10, avec Visual Studio 2022 version 17.9.1 et le moteur MonoGame. MSTest est utilisé pour les tests unitaires*

- **Erreurs restantes**

Dans mon code il y a quelques erreurs de style:

- *La barre de vie ne baisse pas adéquatement*
- *Pas de background*

- **Conclusions**

Pour ce projet je pense avoir réussi car la plupart des fonctionnalités n'ont aucun souci cependant la barre de vie n'a pas été développée parfaitement ce qui fait donc qu'elle ne marche pas très bien.

Je suis assez content de mon projet car le jeu est fonctionnel et je pense pouvoir y jouer. Comme dans un projet précédent je n'ai pas du tout aimé le système de IceScrum surtout dans ce projet-là, car on était des équipes de 1 personne.

Après pour la refactorisation je n'ai pas rencontré plus de soucis que ça.

Je pense que par la suite j'essaierai d'ajouter une base de données avec des améliorations du personnage et d'autres armes.

- **Annexes**

- **Journal de travail**

Le journal de travail se trouve dans la racine du projet dans le dossier doc