Projet XYZ

Pour projet Shoot Me Up avec XCL

Table des matières

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.

1 Analyse préliminaire

1.1 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.

1.2 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

1.3 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.

2 Analyse / Conception

2.1 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.

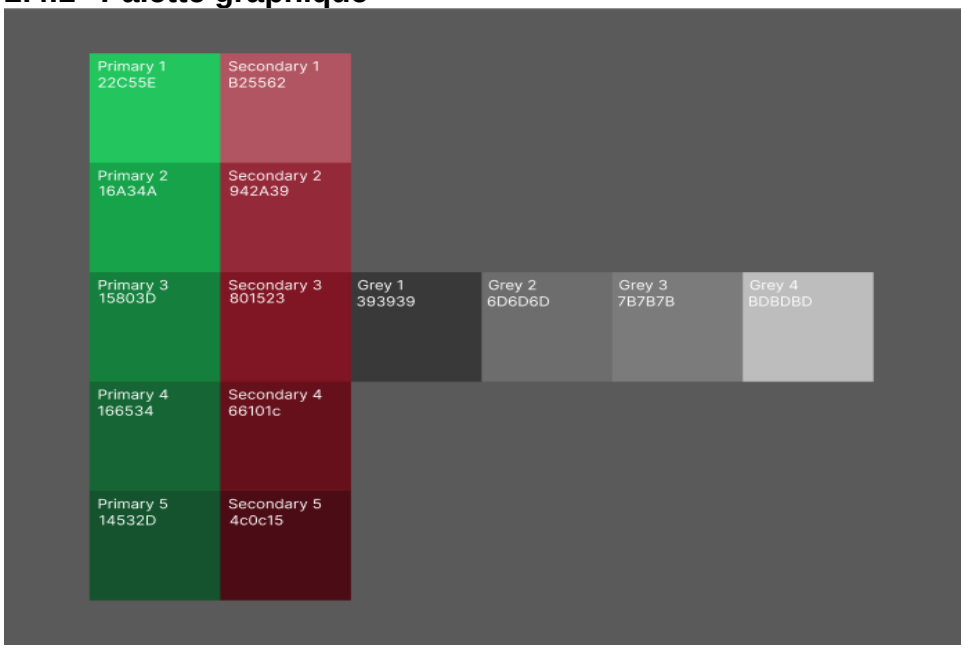
2.2 Concept

- *Diagramme de classe*
- *Diagramme(s) d'état*

2.3 Analyse fonctionelle

2.4 UX

2.4.1 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 à la force pour le soldat, et au côté mort-vivant pour le zombie.

J'ai aussi choisi un rouge foncé comme couleur secondaire, pour représenter le sang et donc le zombie. Malheureusement, ce rouge n'apparaît pas dans la maquette haute-fidélité, car c'était compliqué de l'ajouter sans réduire le contraste visuel.
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

2.4.2 Eco-Conception / Accessibilité

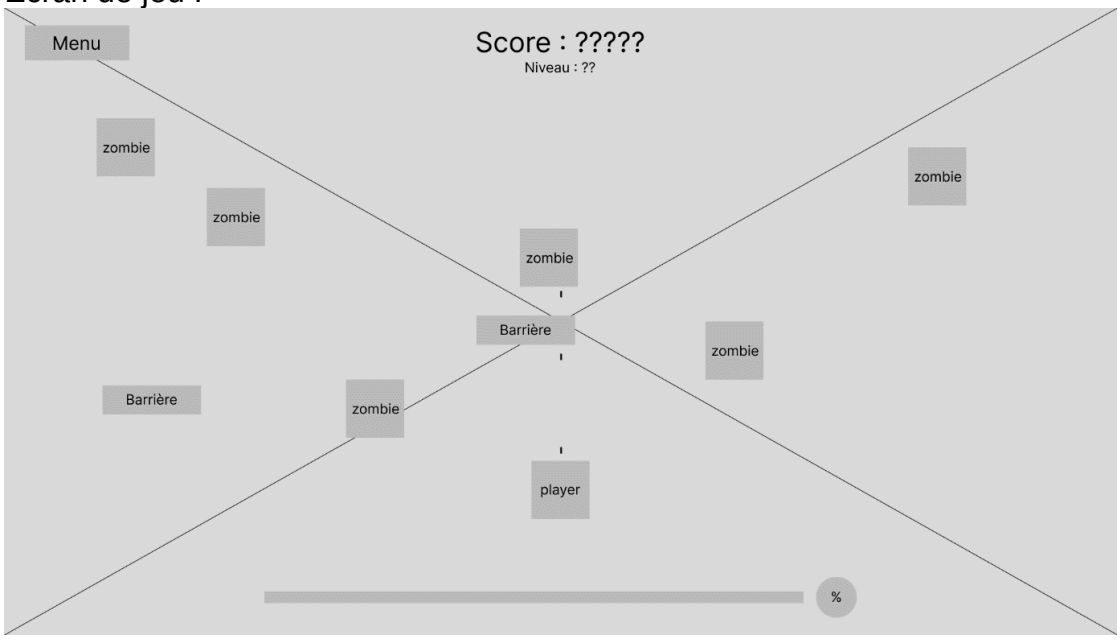
L'accessibilité en UX vise à concevoir des sites et applications utilisables par tous, y compris les personnes en situation de handicap. Ça prend en compte que les textes soient lisibles, des emplacements intuitifs et le fait que les technologies comme les lecteurs d'écrans puissent être utiliser sans 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.

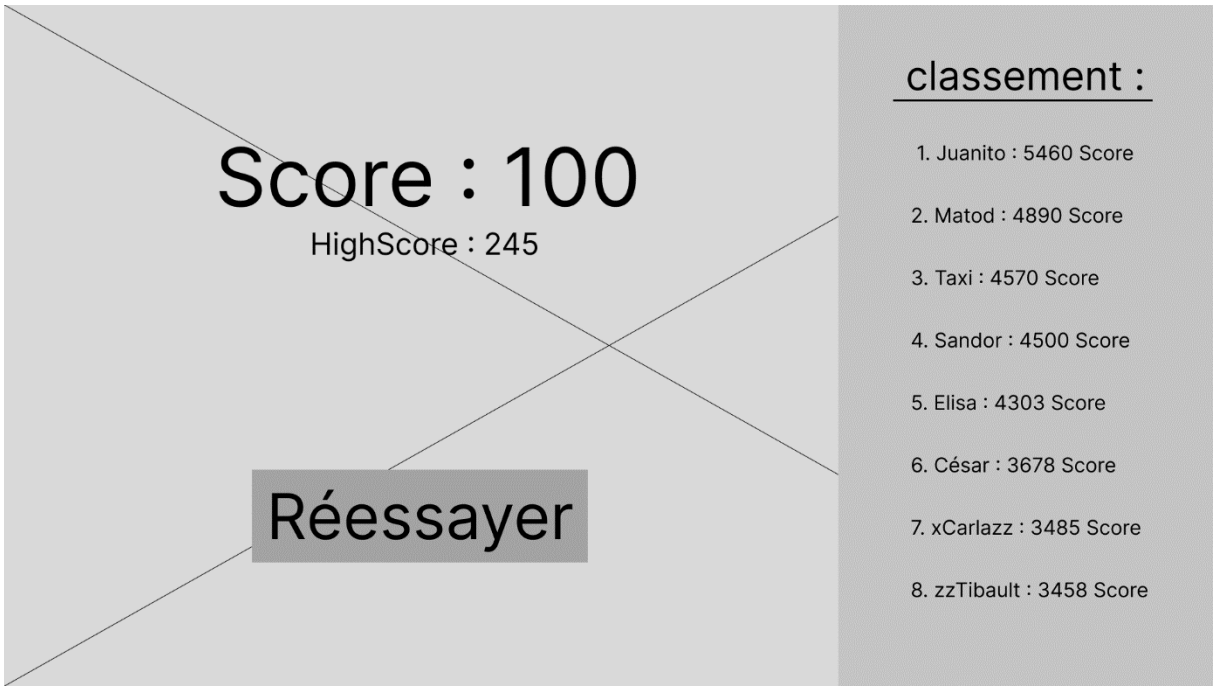
2.5 Définition des écrans wireframe

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 :

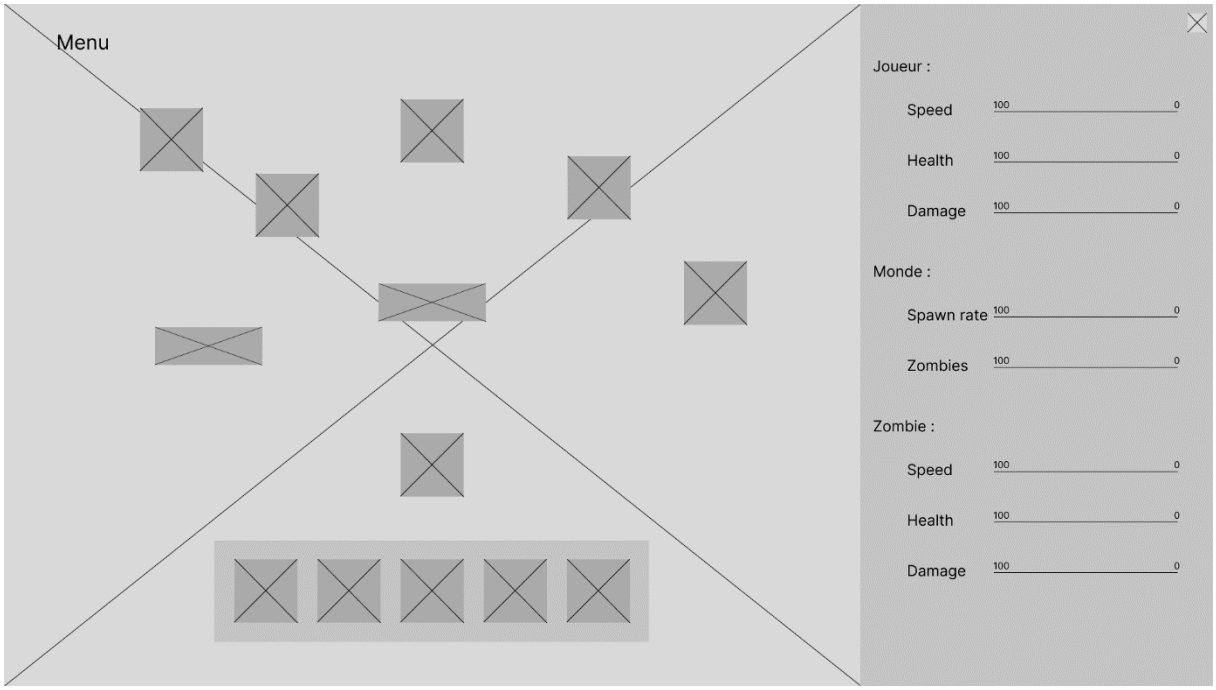
Écran de jeu :



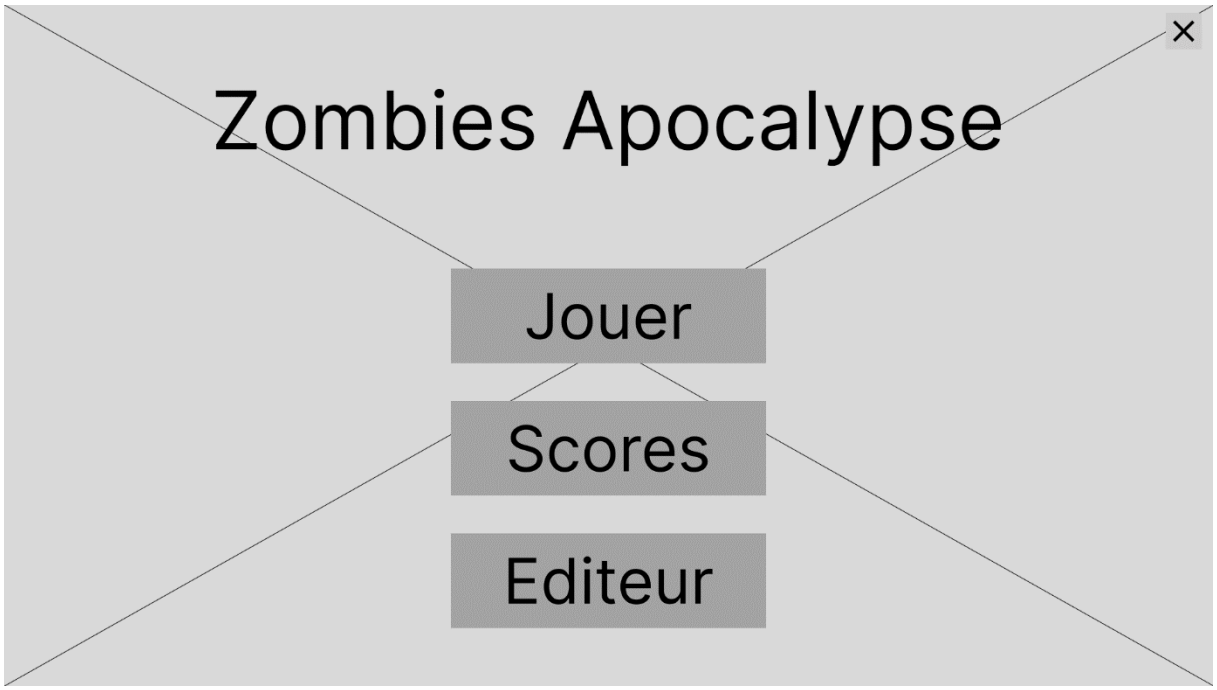
Écran de mort :



Éditeur de niveau :

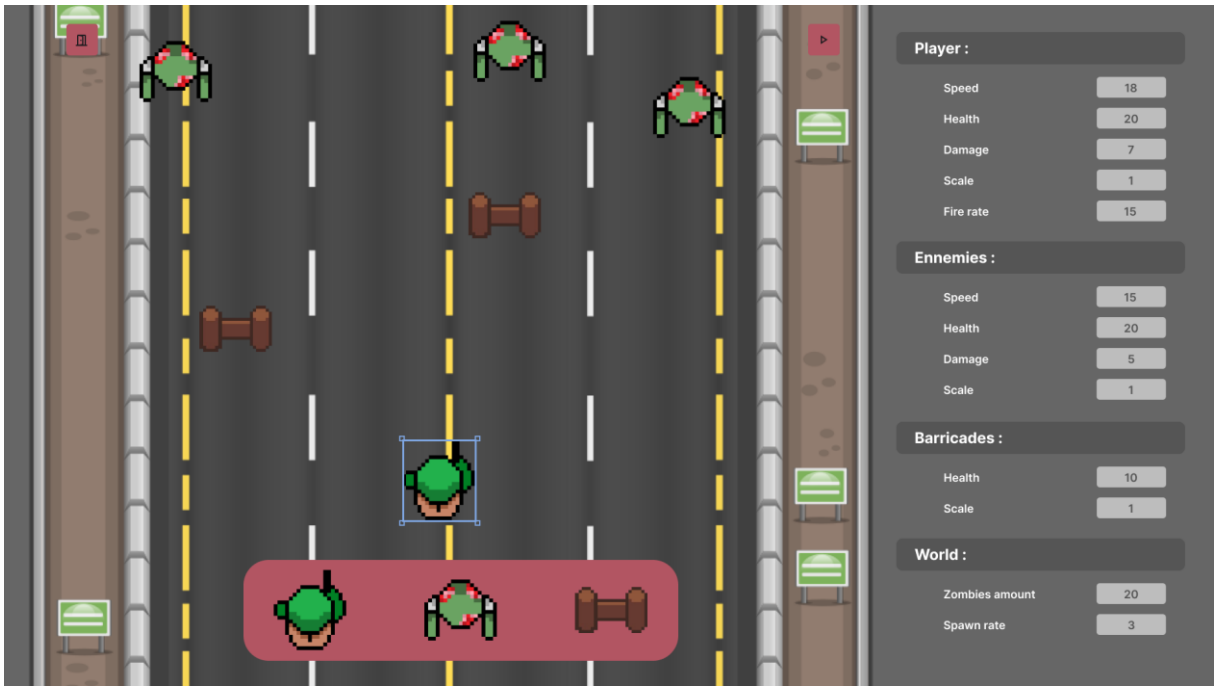


Menu Principal :



2.6 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.



2.7 Choix Effectuées sur le mockup

2.7.1 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

2.7.2 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

2.8 Stratégie de test

Décrire quels sont les MOYENS utilisés pour faire les tests, ne pas décrire les tests à effectuer !!!

Décrire l'environnement dans lequel se fait la sprint review

Décrire la stratégie globale de test :

- *types de des tests et ordre dans lequel ils seront effectués.*
- *les moyens à mettre en œuvre.*
- *couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).*
- *données de test à prévoir (données réelles ?)* ***et comment elles seront mises en place****.*
- *les testeurs extérieurs éventuels.*

3 Réalisation

3.1 Points de design spécifiques

Ce chapitre est constitué de plusieurs sous-chapitre.

Chaque sous-chapitre explique un point de design technique particulier, quelque chose que vous avez dû inventer pour répondre au besoin et qui ne peut pas s'expliquer par de simples commentaires dans le code.

Il s'agit d'explications techniques sur le fonctionnement du système. Les explications sont appuyées par des diagrammes, ou de très brefs éléments de code.

NE PAS mettre ici des pratiques usuelles que tout professionnel de la branche connaît déjà. Par exemple, n'EXPLIQUEZ PAS ICI CE QU'EST LE PATTERN MVC.

Exemple (simplifié à l'extrême) : Protection contre des formulaires mal intentionnés ou modifiés

- ***Au moment de générer le formulaire, le script php :***
  - ***Concatène les noms de tous les champs contenus dans le formulaire***
  - ***Calcule un hash SHA256 de la chaîne obtenue***
  - ***Ajoute un input nommé « CSRF » de type hidden dans le form***
- ***A la réception du POST du fromulaire***
  - ***Concatène les noms des indices de $_POST***
  - ***Calcule un hash SHA256 de la chaîne obtenue***
  - ***Vérifie que la valeur du champ CSRF correspond***

3.1.1 …

3.1.2 …

3.1.3 …

3.2 Déroulement

Résumer comment s'est passé la réalisation de chaque story, ses difficultés, les alternatives envisagées mais rejetées, ses surprises, …

3.3 Mise en place de l'environnement de travail

- ***Comment accéder au code source***
- *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
- *les versions des systèmes d'exploitation et des outils logiciels*
- *la description exacte du matériel*

Ce chapitre décrit précisément comment un employé qualifié peut recréer l'environnement dans lequel vous avez effectué ce travail

3.4 Description des tests effectués

Reprendre les tests d'acceptance d'IceScrum au moyen de la feuille ad hoc d'IceTools

3.5 Erreurs restantes

S'il reste encore des erreurs :

- *Description détaillée*
- *Conséquences sur l'utilisation du produit*
- *Actions envisagées ou possibles*

Reporter la dette technique connue. S'appuyer sur la pratique des // TODO

4 Conclusions

Développez en tous cas les points suivants :

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

5 Annexes

5.1 Manuel de référence

Issu de la génération automatique à partir des commentaires

5.2 Journal de travail