

Demon Butcher

Pour projet Shoot Me Up avec XCL

Table des matières

1	Analyse préliminaire	4
1.1	Introduction	4
1.2	Objectifs.....	4
1.3	Gestion de projet	5
2	Analyse / Conception.....	6
2.1	Gameplay	6
2.2	Concept	6
2.3	Analyse fonctionnelle.....	6
2.4	Stratégie de test.....	6
3	Analyse UX.....	7
3.1	Conception centrée utilisateur	7
3.2	Choix de la palette graphique	9
3.3	Eco-conception	10
3.4	Accessibilité	11
4	Conception UX	12
4.1	Définition des wireframes	12
4.2	Editeur de niveau (High fidelity Mockup)	14
4.3	Choix effectués.....	15
5	Réalisation.....	16
5.1	Points de design spécifiques	16
5.1.1	16
5.1.2	16
5.1.3	16
5.2	Déroulement	16
5.3	Mise en place de l'environnement de travail	16
5.4	Description des tests effectués	17
5.5	Erreurs restantes	17
6	Conclusions	18
7	Annexes.....	19
7.1	Manuel de référence.....	19
7.2	Journal de travail	19

1 Analyse préliminaire

1.1 Introduction

Ce projet consiste à créer un jeu en 2D de type *Shoot 'Em Up* réalisé dans le cadre de différents modules (C106 : Base de données, 320 : Programmation orienté objet et 322 : Expérience utilisateur) réunis en un gros projet. Le choix de rassembler ces trois sujets en un projet ludique permet aux élèves d'en apprendre plus sur chacun de ces domaines et d'acquérir de l'expérience en étant immergés dans un projet concret.

Ce chapitre décrit brièvement le projet, le cadre dans lequel il est réalisé, les raisons de ce choix et ce qu'il peut apporter à l'élève ou à l'école. Il n'est pas nécessaire de rentrer dans les détails (ceux-ci seront abordés plus loin) mais cela doit être aussi clair et complet que possible (idées de solutions). Ce chapitre contient également l'inventaire et la description des travaux qui auraient déjà été effectués pour ce projet.

Ces éléments peuvent être repris des spécifications de départ.

1.2 Objectifs

Ce chapitre énumère les objectifs du projet. L'atteinte ou non de ceux-ci devra pouvoir être contrôlée à la fin du projet. Les objectifs pourront éventuellement être revus après l'analyse.

Ces éléments peuvent être repris des spécifications de départ.

1.3 Gestion de projet

Pour gérer ce projet, nous utilisons IceScrum pour la gestion des tâches, du journal de travail, ainsi que pour la création d'user-stories et de tests d'acceptances. D'un autre côté, nous utilisons GitHub pour la gestion du versionnement du code et pour le suivi des modifications sur la durée.

Ce chapitre décrit la méthode de gestion de projet utilisée, ainsi que les éventuelles particularités requises par le contexte et/ou le chef de projet

2 Analyse / Conception

2.1 Gameplay

- *Le joueur*
- *Les ennemis*
- *Les déplacements*
- *Les niveaux*
- *Le tir*
- *La gestion des vies*
- *...*

2.2 Concept

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

2.3 Analyse fonctionnelle

Reprendre le contenu des User Stories d'IceScrum : Story + tests d'acceptance (avec IceTools) + maquettes

2.4 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.*

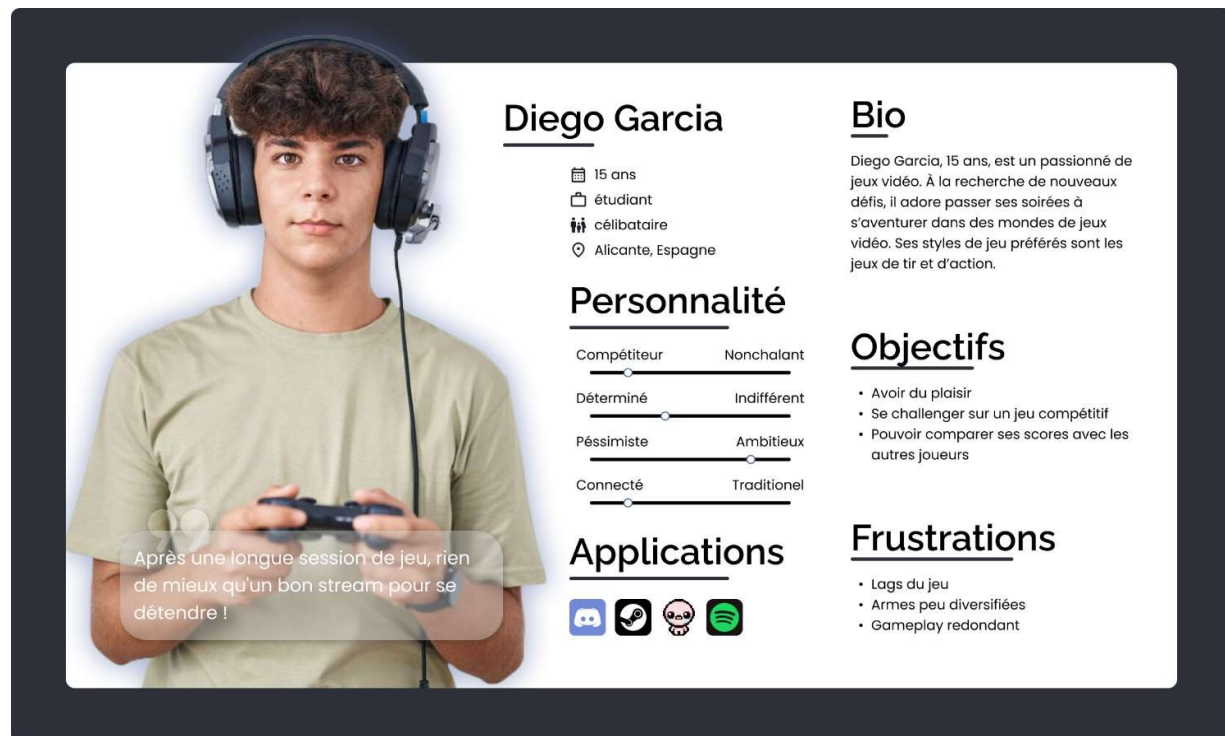
3 Analyse UX

3.1 Conception centrée utilisateur

Dans la création de jeux vidéo, l'approche de conception centrée utilisateur met l'accent sur les besoins et préférences des joueurs, ce qui favorise une expérience de jeu optimale et engageante.

Pour *Demon Butcher*, nous avons identifié deux profils d'utilisateurs types, ou « personas ». Ces personas représentent les motivations et attentes principales de groupes de joueurs distincts, ce qui permet alors d'orienter les choix de conception vers des expériences axés sur les utilisateurs.

1) Diego Garcia



Diego Garcia

15 ans
étudiant
célibataire
Alicante, Espagne

Bio

Diego Garcia, 15 ans, est un passionné de jeux vidéo. À la recherche de nouveaux défis, il adore passer ses soirées à s'aventurer dans des mondes de jeux vidéo. Ses styles de jeu préférés sont les jeux de tir et d'action.

Personnalité

Compétiteur — Nonchalant
Déterminé — Indifférent
Pessimiste — Ambitieux
Connecté — Traditionnel

Applications

Discord, Twitch, YouTube, Spotify

Objectifs

- Avoir du plaisir
- Se challenger sur un jeu compétitif
- Pouvoir comparer ses scores avec les autres joueurs

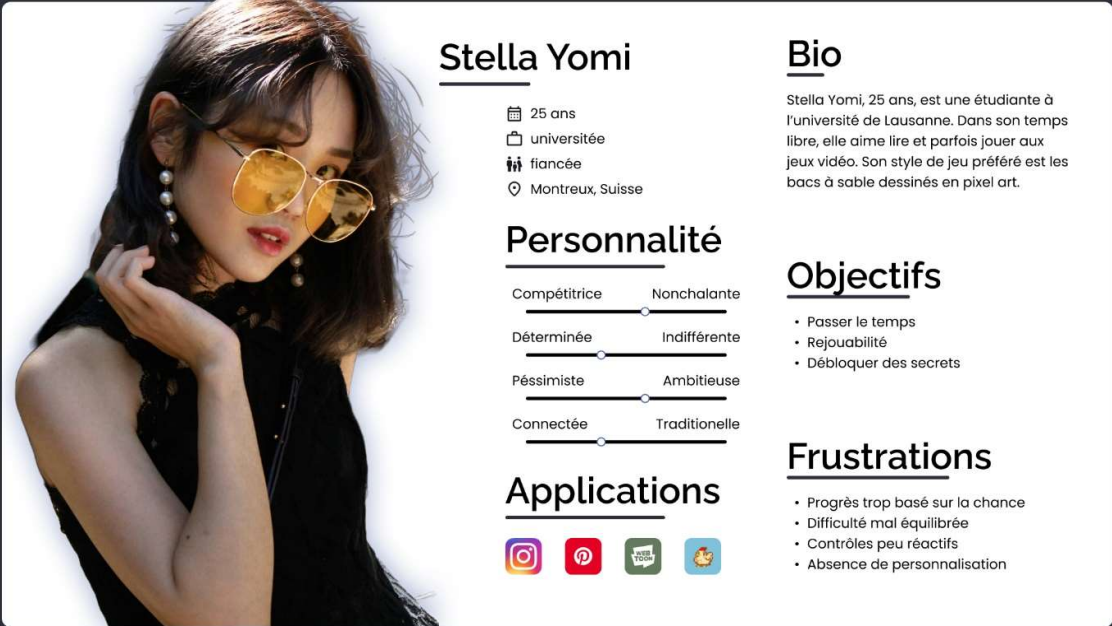
Frustrations

- Lags du jeu
- Armes peu diversifiées
- Gameplay redondant

Après une longue session de jeu, rien de mieux qu'un bon stream pour se détendre !

Ce persona représente un joueur jeune et passionné par les défis et la compétition. Il recherche une expérience intense où il peut repousser ses limites et comparer ses scores à ceux des autres joueurs. Il apprécie particulièrement les jeux de tir et d'action, où chaque partie est l'occasion de se surpasser. La diversité des défis, le rythme rapide, et une fluidité dans le gameplay sont essentiels pour lui.

2) Stella Yomi



Stella Yomi

📅 25 ans
🏠 université
💍 fiancée
📍 Montreux, Suisse

Bio

Stella Yomi, 25 ans, est une étudiante à l'université de Lausanne. Dans son temps libre, elle aime lire et parfois jouer aux jeux vidéo. Son style de jeu préféré est les bacs à sable dessinés en pixel art.

Personnalité

Compétitrice — Nonchalante
Déterminée — Indifférente
Pessimiste — Ambitieuse
Connectée — Traditionnelle

Applications

📷 📌 🗨️ 🐣

Objectifs

- Passer le temps
- Rejouabilité
- Débloquer des secrets

Frustrations

- Progrès trop basé sur la chance
- Difficulté mal équilibrée
- Contrôles peu réactifs
- Absence de personnalisation

Ce persona incarne une joueuse plus détendue, qui se tourne vers les jeux pour se divertir sans pression. Cette joueuse apprécie les environnements immersifs, avec une préférence pour des jeux qui permettent d'explorer et de prendre le temps de découvrir des éléments cachés ou uniques. Les joueurs « casual » cherchent avant tout un jeu accessible, intuitif et plaisant, où le gameplay peut être aussi simple que relaxant.

3.2 Choix de la palette graphique

Pour Demon Butcher, le choix des couleurs est important pour créer l'ambiance du jeu. La palette de couleur est divisée en deux parties : des tons neutres pour les éléments de fond et des couleurs d'accent pour attirer l'attention sur certains détails.



1) Tons Neutres :

Les gris clairs au noir profond servent de fond au jeu. Ces couleurs neutres posent les bases visuelles sans être trop présentes, ce qui laisse les éléments principaux ressortir.

2) Couleurs d'Accent

Les couleurs d'accent vont des tons rose clair jusqu'aux rouges très foncés. Ces couleurs sont parfaites pour les éléments clés, comme les ennemis, les effets d'attaque ou les éléments d'UI importants. Elles rappellent les thèmes « démon » et « sanguinaire » du jeu, ce qui aide à renforcer l'immersion dans l'univers du jeu.

3.3 Eco-conception

L'éco-conception dans Demon Butcher vise à réduire l'empreinte environnementale du jeu en optimisant les ressources utilisées et en minimisant l'impact des éléments graphiques et techniques. Pour accomplir cela, plusieurs principes ont été appliqués dans la conception de l'interface et des menus :

1) Optimisation des images

Les éléments visuels (les icônes, les arrière-plans etc.) sont conçus pour être légers. En limitant la taille et la complexité des images dans les menus, le jeu devient non seulement moins lourd à télécharger mais utilise aussi moins de mémoire et de puissance de traitement, ce qui vise à réduire la consommation d'énergie.

2) Palette de couleurs limitée

En restreignant la palette de couleurs aux tons neutres et à quelques accents, on réduit le nombre d'éléments de design uniques. Cela simplifie les fichiers de style et diminue l'utilisation des ressources matérielles, ce qui rend l'interface plus efficace. L'utilisation de nuances de gris et de rouge reste cohérente avec l'ambiance tout en gardant le design efficace et minimaliste.

3) Interface épurée et intuitive

Quand on adopte un design épuré, on réduit le nombre d'éléments à afficher, ce qui diminue la demande en ressources graphiques. Les menus sont clairs et bien organisés, avec des éléments essentiels bien mis en évidence, ce qui rend la navigation rapide et intuitive tout en optimisant les ressources.

3.4 Accessibilité

L'accessibilité vise à rendre le jeu et ses menus accessibles à un maximum de joueurs, quels que soient leurs besoins, limitations ou handicaps. Plusieurs éléments ont été mis en place pour faciliter la navigation et l'interaction avec le jeu afin d'avoir une expérience inclusive.

1) Contrastes de couleurs élevés

La palette de couleurs utilise des contrastes forts entre les éléments interactifs (rouge pour la plupart) et les arrière-plans neutres (généralement blanc). Ces contrastes permettent de rendre le texte et les boutons facilement visibles, même pour les joueurs ayant des problèmes de vue ou des difficultés à distinguer certaines couleurs comme le daltonisme.

2) Textes lisibles

Les menus et les informations de jeu utilisent une taille de police suffisamment grande pour être lisible sans effort. Des polices claires et sans empattement ont été choisies pour renforcer la lisibilité, en particulier pour les titres et les CTA.

3) Navigation simplifiée

L'interface est conçue de manière à éviter les surcharges d'information, avec une organisation claire et des options bien identifiées. Les écrans sont épurés et faciles à naviguer, minimisant les distractions pour les joueurs qui peuvent rencontrer des difficultés de concentration ou des troubles de l'attention.

4 Conception UX

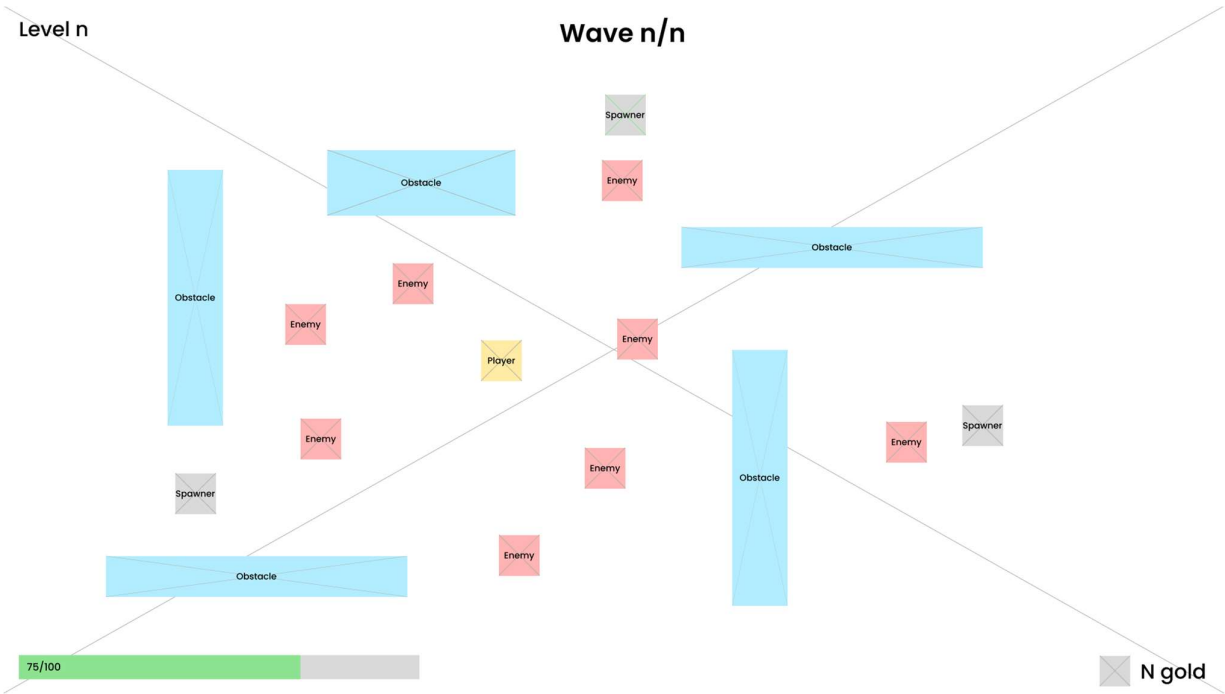
4.1 Définition des wireframes

Les wireframes sont des maquettes simples qui structurent les principaux écrans et menus du jeu. Ils servent à planifier l'organisation de chaque élément de l'interface pour assurer une navigation intuitive et cohérente. Ces maquettes présentent la disposition des menus, des options de personnalisation et des écrans de gameplay, offrant une vue d'ensemble claire avant la mise en forme finale.

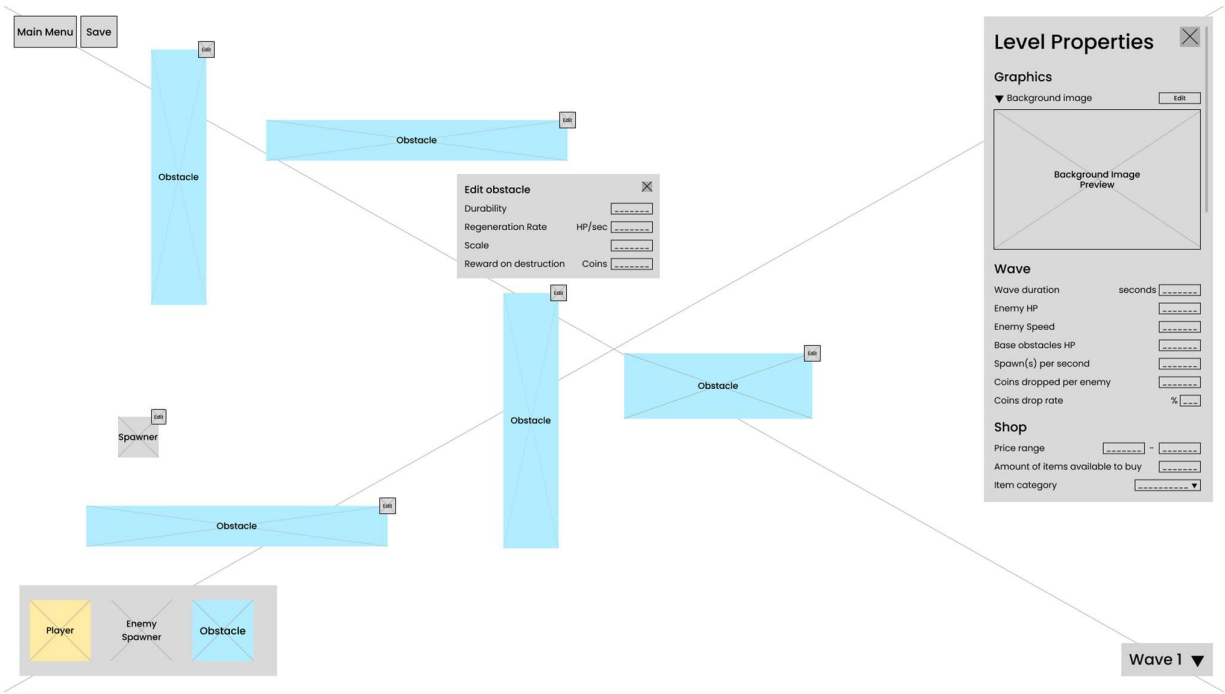
1) Menu principal



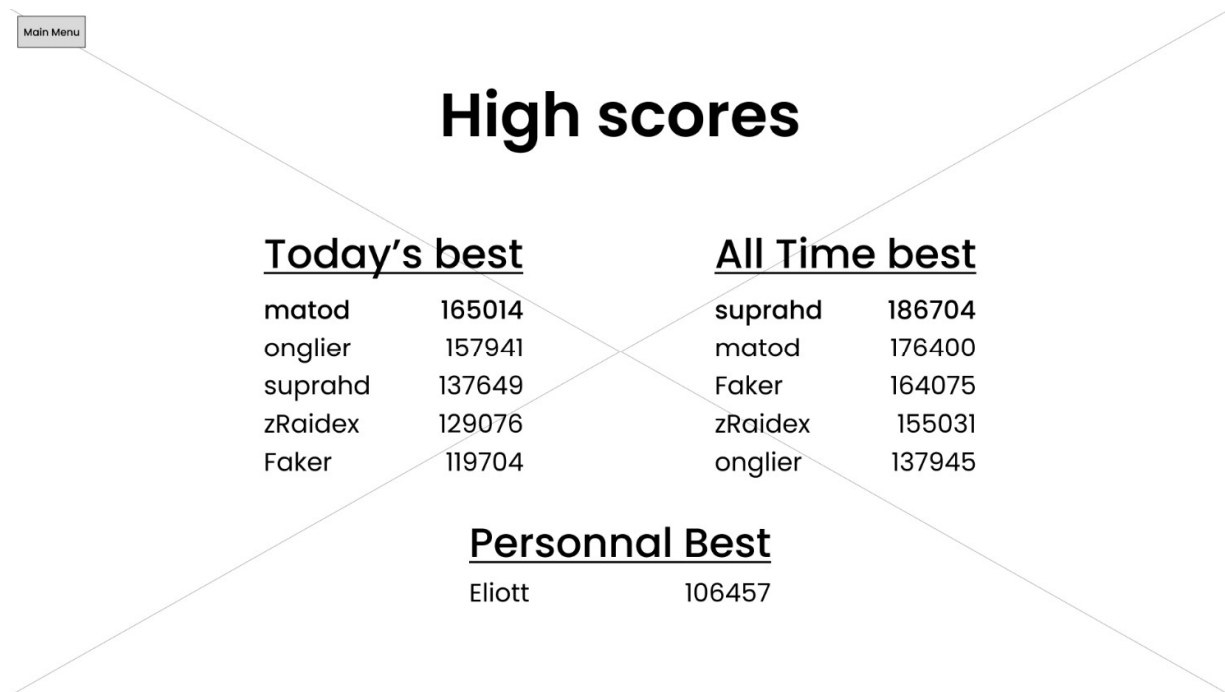
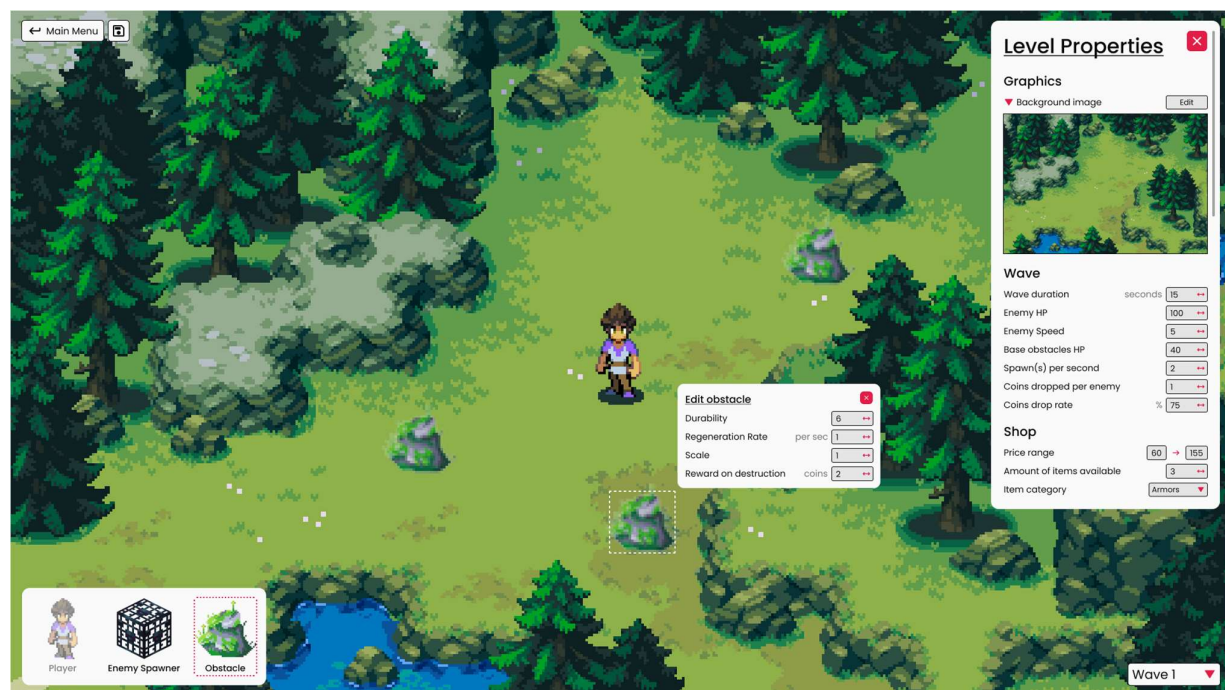
2) Ecran de gameplay



3) Editeur de niveau



4) Ecran des high scores

4.2 Editeur de niveau (High fidelity Mockup)

4.3 Choix effectués

Dans la conception UX, plusieurs choix ont été faits pour améliorer l'expérience utilisateur et garder une interface fluide et intuitive.

1) Menus flottants pour économiser de l'espace

L'éditeur de niveau utilise des menus flottants pour minimiser l'encombrement de l'écran. En plaçant les options de création et de personnalisation dans des panneaux discrets ou en overlay sur les côtés, les joueurs ont d'une vue dégagée sur le niveau, ce qui facilite la création. Cette configuration aide à focaliser l'attention sur le contenu du niveau mais offre tout de même un accès rapide aux options essentiels.

2) Uniformité de la résolution entre l'éditeur et le jeu

L'éditeur conserve la même résolution que celle utilisée en jeu, ce qui permet aux joueurs de voir le niveau exactement tel qu'il apparaîtra lorsqu'ils joueront. Ce choix renforce la cohérence visuelle et réduit les ajustements nécessaires entre l'édition de niveau et le gameplay.

3) Pop-ups de paramétrage pour les objets et entités

Lorsqu'un joueur sélectionne un objet ou une entité dans l'éditeur, une pop-up s'ouvre pour permettre la modification de ses paramètres (comme la taille, la position, les caractéristiques, etc.). Cela améliore l'interactivité en donnant un accès rapide aux options de personnalisation sans naviguer dans des menus compliqués.

4) Accessibilité rapide aux outils essentiels

Le menu de l'éditeur va à l'essentiel, regroupant les outils fréquemment utilisés dans des barres de navigation bien placées. Cette organisation complète sans être complexe permet aux joueurs d'effectuer leurs actions rapidement et efficacement. Les options sont claires et facilement accessibles, facilitant les ajustements en temps réel et optimisant le flux de création.

5) Indicateur de placement unique pour le joueur

Une fois le joueur placé depuis la barre d'objets dans le niveau, il apparaît en gris dans la barre pour indiquer qu'il ne peut plus être ajouté à nouveau. Ce choix garantit qu'un seul joueur est présent sur le terrain de jeu avec un indicateur intuitif.

5 Réalisation

5.1 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 formulaire*
 - *Concatène les noms des indices de \$_POST*
 - *Calcule un hash SHA256 de la chaîne obtenue*
 - *Vérifie que la valeur du champ CSRF correspond*

5.1.1 ...

5.1.2 ...

5.1.3 ...

5.2 Déroulement

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

5.3 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

5.4 Description des tests effectués

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

5.5 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

6 Conclusions

Développez en tous cas les points suivants :

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

7 Annexes

7.1 Manuel de référence

Issu de la génération automatique à partir des commentaires

7.2 Journal de travail