Survive Cavazzino

Game design document

Table des matières :

1. Histoire

2. Personnages
   - 2.1. Joueur
   - 2.2. Ennemis
     - 2.2.1. Concepts
     - 2.2.2. Mécanique
     - 2.2.3. Typologie

3. Gameplay
   - 3.1. Concept Général
   - 3.2. Structure d'une partie
   - 3.3. Système de combat
   - 3.4. La caméra
   - 3.5. Contrôles

4. Design
   - 4.1. Interface utilisateur
   - 4.2. Personnages
   - 4.3. Map
   - 4.4. Objets
   - 4.5. Sons

5. Documentation technique
   - 5.1. Technologies utilisées
   - 5.2. Diagramme UML

L'Histoire de la Cave

L'Éveil

Vous êtes Leromz. Du moins, c'est le nom qui résonne dans votre crâne vide lorsque vous vous extirpez d'un sommeil sans rêve, enchaîné dans une cave humide. L'air est vicié, chargé de l'odeur de clope et de zaza.

Vous ne savez pas qui vous a enfermé ici, ni pourquoi. La seule chose qui compte, c'est cette pulsation dans vos tempes : **Survivre.**

D'un effort surhumain, vous brisez vos fers. Votre seule arme : un morceau de métal froid arraché au mur. La cave grouille. Des choses, nées des ténèbres, se jettent sur vous.

L'Éveil de la Bête

Le combat est brutal, viscéral. Chaque créature que vous abattez libère une essence, une lueur que vous absorbez instinctivement. Elle vous emplit d'une puissance brute, addictive. Le temps lui-même semble se figer lorsque cette puissance vous submerge, vous offrant le choix : durcir votre peau, aiguiser vos sens, ou maîtriser de nouveaux instruments de mort.

Vous êtes efficace. Vous êtes rapide. Vous devenez une machine à tuer.

Le Compte à Rebours

Mais quelque chose d'autre s'est enclenché. Un compte à rebours invisible.

Ce n'est pas une horloge. C'est un pouls. À chaque monstre que vous éventrez, à chaque essence que vous dévorez, ce pouls s'accélère. Ce n'est pas le temps qui vous est compté ; c'est votre humanité qui s'effrite.

Plus vous tuez de monstres, plus vous vous rapprochez de votre véritable éveil. Vous ne le sentez pas encore, mais vous changez. Pour tuer le monstre, vous en devenez un.

Le Miroir

Le compte à rebours atteint son terme.

Le chaos de la cave s'arrête net. Silence absolu. Les créatures restantes s'effondrent en poussière. Devant vous, un miroir antique, voilé de brume, se matérialise.

Vous vous approchez. Votre reflet vous fixe. Ce n'est plus Leromz. C'est une silhouette tordue, les yeux brillants d'une faim que vous reconnaissez. C'est le monstre que vous avez nourri.

La créature dans le miroir sourit. Elle tend la main et traverse la surface comme si c'était de l'eau.

Il se tient là. **Cavazzino Cavazzini.** C'est vous.

Il ne veut pas vous faire de mal. Il veut que vous *acceptiez*. Le combat qui s'engage n'est pas pour votre liberté. C'est pour votre âme.

L'Acceptation

En terrassant votre reflet, vous ne trouvez pas la sortie. Vous ne faites que briser la dernière illusion de votre humanité. Le miroir éclate.

La cave semble prendre une profonde inspiration. Les murs suintent, les ténèbres s'épaississent. Les monstres reviennent, plus rapides, plus forts, infinis. Le but n'est plus de s'échapper. Le but est de *régner*.

Le Cycle

Mais même un monstre peut mourir. Inévitablement, vous succombez. Le poids des ténèbres est trop lourd. L'obscurité vous reprend.

...Puis, la lumière.

Vous vous extirpez d'un sommeil sans rêve. Enchaîné. Le goût de la rouille dans la bouche. En brisant vos fers, votre regard est attiré par un coin sombre de la cellule. Sur le sol, à peine visible, une pile de barres de fer rouillées, tordues, brisées par un usage répété. Des centaines.

Vous comprenez. Vous n'êtes pas le premier Leromz. Vous n'êtes qu'une nouvelle tentative. Vous ramassez votre arme. La cave attend.

---

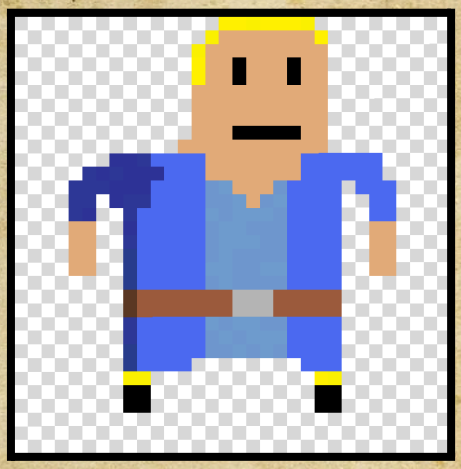
2 - Personnage

2.1 - Joueur

Pour l'instant un seul personnage jouable Leromz.

2.1.1 Concept et Identité

- **Nom du protagoniste :** Leromz
- **Concept central :** Le parcours du joueur est une quête d'acceptation de soi. La progression n'est pas uniquement statistique, elle est également visuelle. À mesure que le joueur gagne en puissance, son apparence physique évolue, le rapprochant de sa forme finale "Cavazzino Cavazzini".



(1 er Artwork de leromz en 32x32)

(Artwork final de leromz en 36x21)

2.1.2 Spécifications initiales (Leromz)

- **Rôle :** Personnage de base, point de départ du joueur.
- **Équipement initial :** Arme de corps à corps basique (ex: "Bâton en fer" ou "Barre de métal").
- **Attributs de base :** Statistiques équilibrées (Santé, Dégâts, Vitesse).
    - *Note : Les valeurs exactes sont à définir lors de la phase d'équilibrage.*

2.2 – Ennemis

👹 Orc Brutal

Description :
Massif, agressif et prêt à écraser tout ce qui respire. Les Orcs Brise-Crâne forment la première ligne des forces hostiles. Leur peau épaissie et leurs muscles hypertrophiés les rendent particulièrement résistants aux dégâts légers.

Comportement :

- Avance lentement mais inexorablement vers le joueur.
- Frappe au corps-à-corps avec une attaque lourde.

Rôle dans le jeu :
Ennemi de "pression" qui oblige le joueur à se déplacer et gérer l'espace.

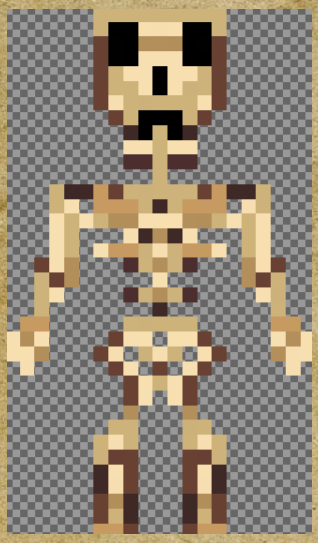
😈 Demon Lactalin

Description :
Petit démon vicieux, rapide, imprévisible. Son apparence innocente cache une agressivité hors pair. Les Démons Lactalin surgissent en meute et saturent l'écran, obligeant le joueur à rester mobile.

Comportement :

- Se déplace rapidement
- Peu résistant
- Harceleur

Rôle dans le jeu :
Ennemi de harcèlement. Il crée du chaos et force le joueur à revoir sa priorité de ciblage.



💀 Squelette

Description :
Un assemblage d'os animés par la magie ancienne de la Cave Azza. Les squelettes ne ressentent ni douleur, ni fatigue. Leurs articulations craquent au moindre mouvement, créant une ambiance oppressante.

Comportement :

- Marche lentement mais en continu, implacable.
- Inflige des dégâts réguliers mais prévisibles c'est l'ennemi lambda

Rôle dans le jeu :
Ennemi "endurant" qui parfait les vagues en comblant l'espace et mettant la pression sur la durée.

2.2.1 Concept Général (Lore)

Les ennemis sont les manifestations physiques de l'inconscient de Leromz (peurs, pulsions, culpabilité). Ils sont décrits comme des "reflets fragmentés" de lui-même, nés du lieu (la "cave"). Le combat est une métaphore de la lutte interne du protagoniste pour "renaître".

2.2.2 Mécaniques Globales

- **Type de combat :** Tous les ennemis de base utilisent des attaques au corps à corps. Lorsqu'ils touchent le joueur, celui-ci prend des dégâts.
- **Scaling :** Les statistiques (points de vie, dégâts) des ennemis augmentent progressivement, plus le temps passe plus la difficulté augmente. le joueur à également une stat difficulté pour augmenter le nombre et la puissance des ennemis

2.2.3 Typologie des Ennemis

1.Orc Brutal

- **Concept :** Ennemi brute et tanky.

- **Comportement IA :** Se déplace lentement mais de manière constante. Priorise le joueur en ligne droite, en exerçant une pression continue.
- **Spécificité :** Très résistant. Inflige des dégâts élevés. Force le joueur à gérer son positionnement.

2. Demon Lactalin

- **Concept :** Ennemi agile et harceleur.
- **Comportement IA :** Extrêmement rapide. Arrive souvent en groupe. Tente d'encercler ou saturer l'écran.
- **Spécificité :** Très faible en points de vie. Dégâts faibles mais difficiles à ignorer. Parfait pour mettre le joueur sous pression constante.

3. Skull Squelettique

- **Concept :** Ennemi lambda / équilibré.
- **Comportement IA :** Avance à vitesse moyenne. Frappe de manière régulière et prévisible.
- **Spécificité :** Statistiques équilibrées. Sert de "remplissage" dans les vagues. Implacable sur la durée lorsqu'ils sont nombreux.

3 - Gameplay

3.1.1 Genre et Piliers

- **Genre :** Jeu de survie en arène (type "Survivor-like"), Rogue Lite.
- **Objectif principal :** Survivre le plus longtemps possible à des vagues d'ennemis continues.
- **Objectif secondaire :** Vaincre l'unique Boss (Cavazzino Cavazzini) pour débloquer le mode "Survie infinie".
- **Condition de défaite :** Mort du joueur.

3.1.2 Mécaniques Fondamentales

- **Progression (XP) :**
  - Les ennemis vaincus génèrent des ressources (XP).
  - La collecte d'XP permet au joueur de monter en niveau et de choisir des améliorations.
- **Évolution (Thématique) :**
  - La progression du joueur (via les améliorations) n'est pas seulement statistique, elle est visuelle.
  - Le personnage de "Leromz" se transforme physiquement pour se rapprocher de sa forme finale (Cavazzino Cavazzini), illustrant une perte d'humanité.
- **Scaling :**
  - La difficulté (statistiques et densité des ennemis) augmente de manière continue, parallèlement à la puissance acquise par le joueur.

3.1.3 Condition de Fin de Partie (End Game)

- **Contre-la-montre :** Survivre le temps que le chrono tombe à 0
- **Mode Infini :** Le jeu entre dans une phase de survie infinie avec un scaling de difficulté exponentiel, jusqu'à la mort inévitable du joueur. L'objectif devient le score (temps de survie).

3.2 Boucle de Gameplay (Core Loop)

La structure d'une tentative unique ("run") suit un cycle continu basé sur 5 phases :

1. Phase de Combat (Survie)

- Le joueur évolue dans la cave.
- Le système génère des vagues d'ennemis continues ou séquencées selon le timing de la partie.
- Le rôle du joueur est l'esquive et le positionnement stratégique et le bon choix de build.

2. Phase de Collecte (Ressource)

- Les ennemis vaincus génèrent une ressource (XP) à leur mort.
- Cette ressource doit être collectée activement par le joueur en marchant dessus.

3. Phase de Progression (Évolution)

- La collecte d'un seuil d'XP défini déclenche une montée de niveau, Level Up.
- Au Level Up, le jeu se met en pause et présente au joueur un choix d'améliorations (ex: 1 choix parmi 3 options aléatoires).
- Les améliorations concernent les armes, les statistiques passives.

4. Phase de Cycle (Mort)

- La mort du joueur met fin à la tentative.
- Toute la progression acquise *durant* cette tentative (niveaux, améliorations) est perdue.
- Le joueur retourne à l'écran initial pour commencer une nouvelle tentative.

3.3 Système de Combat

3.3.1 Mécanique d'Attaque (Automatique)

- **Principe :** Le joueur ne possède aucune commande d'attaque manuelle (pas de bouton "tirer" ou "frapper").
- **Exécution :** Les armes et compétences équipées par le joueur s'activent automatiquement.
- **Cadence :** Chaque arme/compétence possède sa propre cadence de tir ou temps de rechargement (Cooldown).

3.3.2 Système de Ciblage

- **L'arme tire des projectiles, qui partent tout droit et continuent jusqu'à la fin de leur range.**
- **Les projectiles ont une vitesse, une range (portée), et une chance de dégâts critiques ainsi qu'un multiplicateur de dégât.**
    - *Note : Intégration d'un système de ciblage automatique. Certaines améliorations pourraient affecter cette priorité (ex: "viser l'ennemi le plus fort", "viser un ennemi aléatoire").*

3.3.3 Les armes

- Le jeu propose 4 armes :
    - Arme de départ: le bâton de bois
    - Au niveau 2, sélection parmi 3 armes :
        - L'épée
        - La dague
        - Le bâton magique de feu
- Chaque arme a son propre style de jeu: l'épée est l'arme classique, la plus équilibrée; la dague fait moins de dégâts et a une petite portée, mais frappe très vite; et le bâton magique de feu tape lentement mais one-shot tous les ennemis et possède une longue range.

3.3.4 Rôle Actif du Joueur

Le gameplay du joueur ne repose pas sur l'exécution des attaques, mais sur la stratégie de survie :

- **Positionnement :** Déplacement constant pour esquiver les ennemis et les projectiles.
- **Optimisation :** Choisir les bonnes améliorations pour créer des synergies (le "build").
- **Gestion des ressources :** Se déplacer pour collecter l'XP sans prendre de risques excessifs.

3.4 La caméra

La caméra de *Survive Cavazzini* est entièrement centrée sur le joueur. Elle suit **Leromz** en permanence, afin de maintenir son personnage comme point de référence principal.
Ce choix nous a permis de renforcer la lisibilité de l'action dans un jeu où le danger vient de toutes les directions.

Le joueur reste toujours au centre de la zone visible, tandis que l'environnement (la cave, les salles, les couloirs) défile autour de lui.

3.5 Les contrôles

Le joueur se déplace à l'aide des touches ZQSD, permettant d'explorer librement la carte et d'éviter les ennemis. La touche Échap (ESC) ouvre un menu de pause qui fige le gameplay et affiche les statistiques du personnage en temps réel. Lorsqu'un niveau est gagné, un clic gauche permet de sélectionner une amélioration parmi les choix proposés, rendant la progression intuitive.

4.0 Design

Le design du jeu repose sur une direction artistique rétro en pixel-art, pensée pour être lisible et extensible facilement (dans le cadre de potentiels ajouts de features).

L'objectif était de proposer une ambiance sombre mais dynamique, tout en gardant un style simple compatible avec nos contraintes de production.

Nous avons également intégré des codes visuels clairs pour que le joueur identifie rapidement les ennemis, les projectiles, les bonus et les éléments interactifs.

4.1 Interface Utilisateur (UI)

L'interface a été conçue pour être minimaliste et efficace : HP, XP, kills, timer.

Les écrans de **Pause**, **Game Over** et **Victoire** ont été intégrés avec des overlays et des stages.

L'objectif était d'avoir une UI fonctionnelle, réactive et facilement modulable.

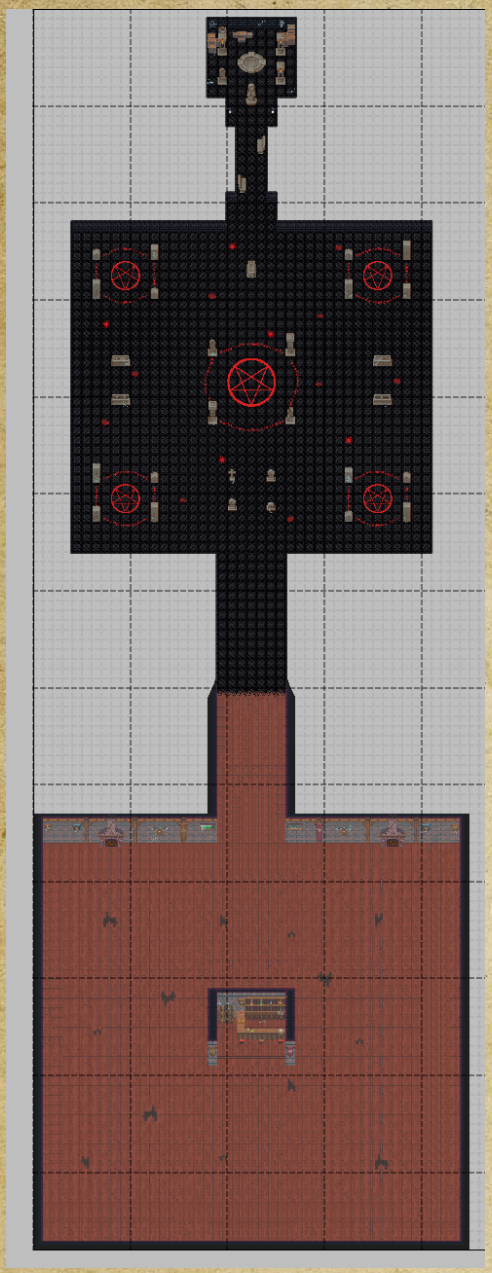
4.2 Personnages

Le joueur a été créé en pixel-art puis intégré dans le moteur via des sprites animées (marche, idle...). Concernant les ennemis un simple sprite leur a été associé

Chaque personnage possède des statistiques propres grâce à notre structure orientée objet (héritage de LivingEntity), ce qui facilite la création de nouveaux ennemis ou de futures classes jouables.

Les animations directionnelles apportent plus de dynamisme au gameplay.

4.3 Map



La carte du jeu a été réalisée avec **Tiled**, puis importée dans LibGDX via les outils de parsing intégrés.

Le level design se base sur une grande zone composée de 2 grandes salles pensées pour offrir un gameplay de survie dynamique.

Le personnage apparaît dans une petite salle de 'spawn'.

Une fois sorti de cette salle, la partie commence, et l'accès à cette zone devient alors impossible.

Les collisions sont gérées par l'object layer "collisions" de la map, permettant au joueur et aux ennemis d'interagir de manière cohérente avec le décor.

La map a été réalisée à partir de sprites faites à la main, et également de sprites gratuites trouvées sur Internet.

4.4 Écrans

Le jeu s'articule autour de cinq écrans principaux gérant le flux de la partie. L'interface est lisible et réactive.

4.4.1. Menu Principal

Cet écran est le point d'entrée du joueur au lancement de l'application.

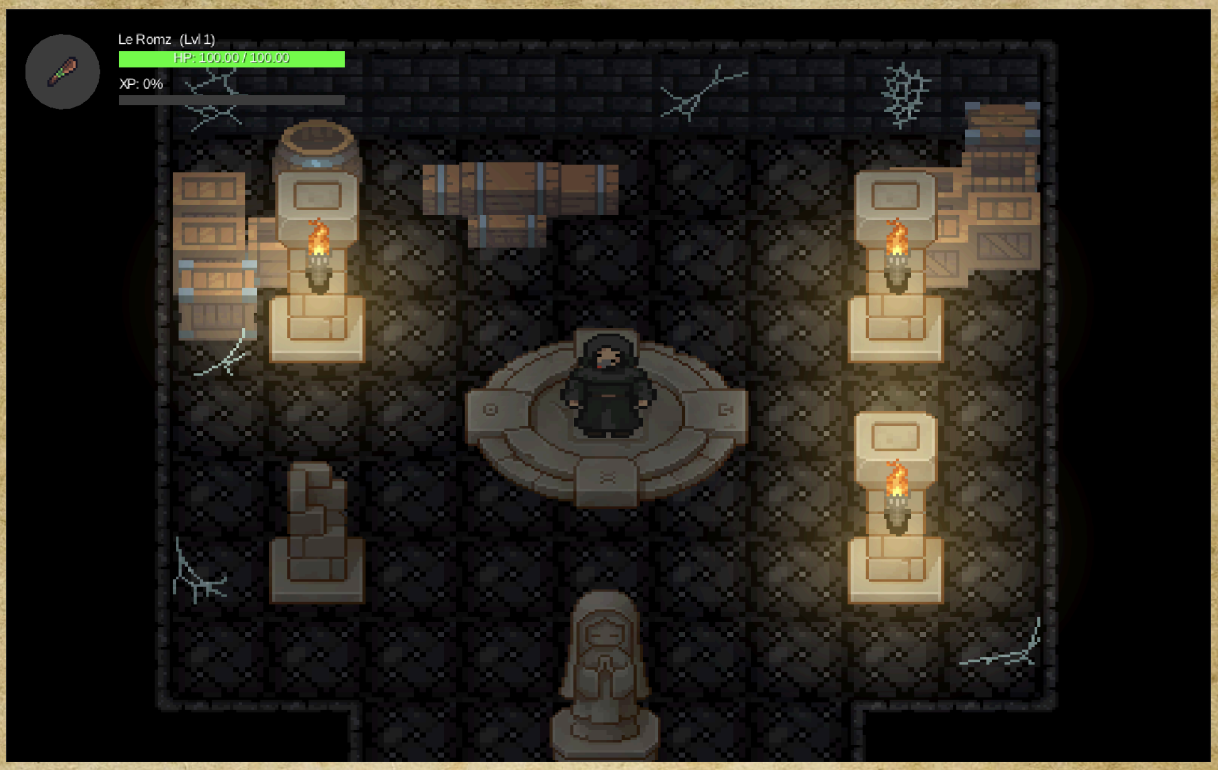
- **Fonctionnalités :**
  - **Jouer :** Lance la boucle de jeu principale (voir *Gameplay*).
  - **Options :** Ouvre un sous-menu pour régler le son, l'affichage du timer et les contrôles.
  - **Quitter :** Ferme l'application.
- **Visuel :** L'arrière-plan doit plonger le joueur dans l'ambiance dès le début.
- **Référence visuelle :**

4.4.2. Écran de Jeu (HUD & Gameplay)

C'est l'écran actif lorsque la partie est en cours. Il affiche l'action en temps réel ainsi que l'interface tête haute (HUD).

- **Déclencheur :** S'active lorsque le joueur appuie sur "Jouer" depuis le menu.
- **Fonctionnalités :**
  - Gestion des déplacements et des attaques du joueur.
  - Affichage des ennemis et des projectiles.
  - **HUD :** Barre de vie, barre d'expérience, compteur d'ennemis tués et chronomètre.

- **Référence visuelle :**

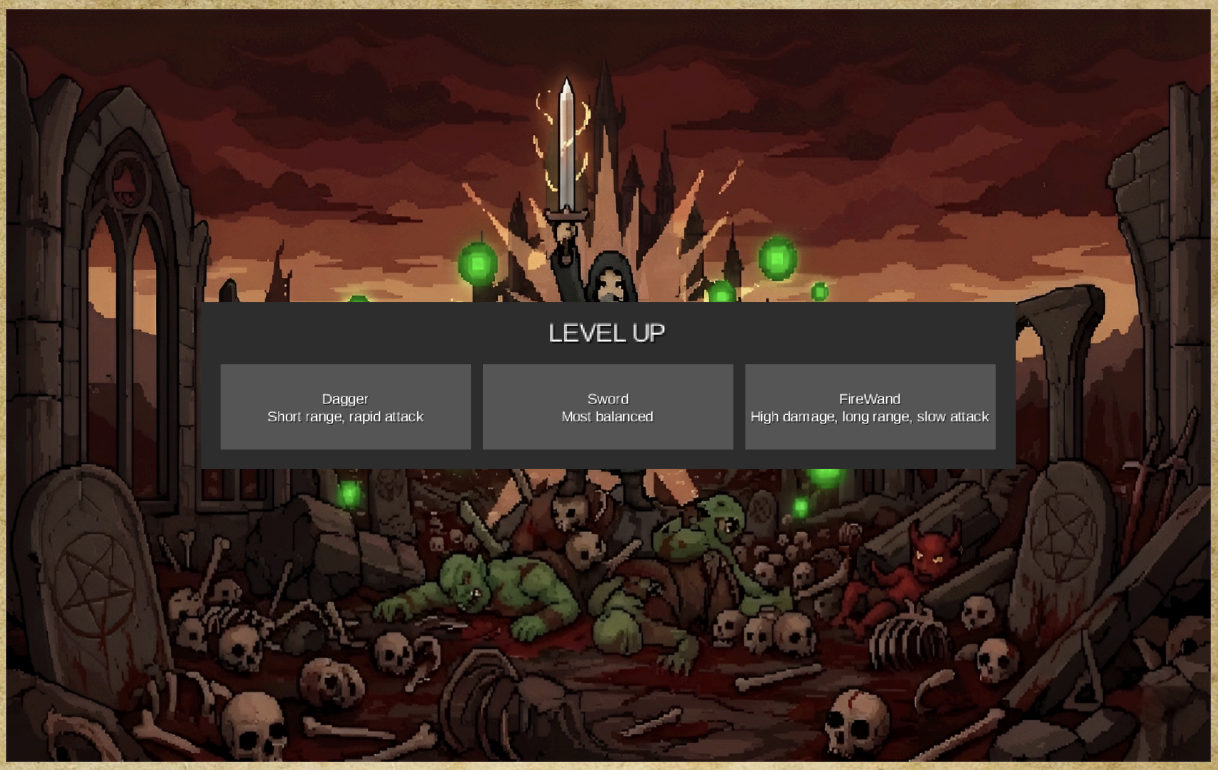


4.4.3. Écran de Montée de Niveau (Level Up)

Cet écran interrompt momentanément le jeu pour permettre au joueur de progresser.

- **Déclencheur :** S'active automatiquement lorsque la barre d'expérience est pleine.
- **Fonctionnalités :**
  - Mise en pause du temps de jeu.
  - **Sélection RNG :** Propose 3 choix aléatoires d'améliorations (augmentation de statistiques ou nouvelle arme/amélioration d'arme).
  - La sélection d'un choix reprend le jeu immédiatement.
- **Visuel :** Utilise un fond spécifique.

- **Référence visuelle :**

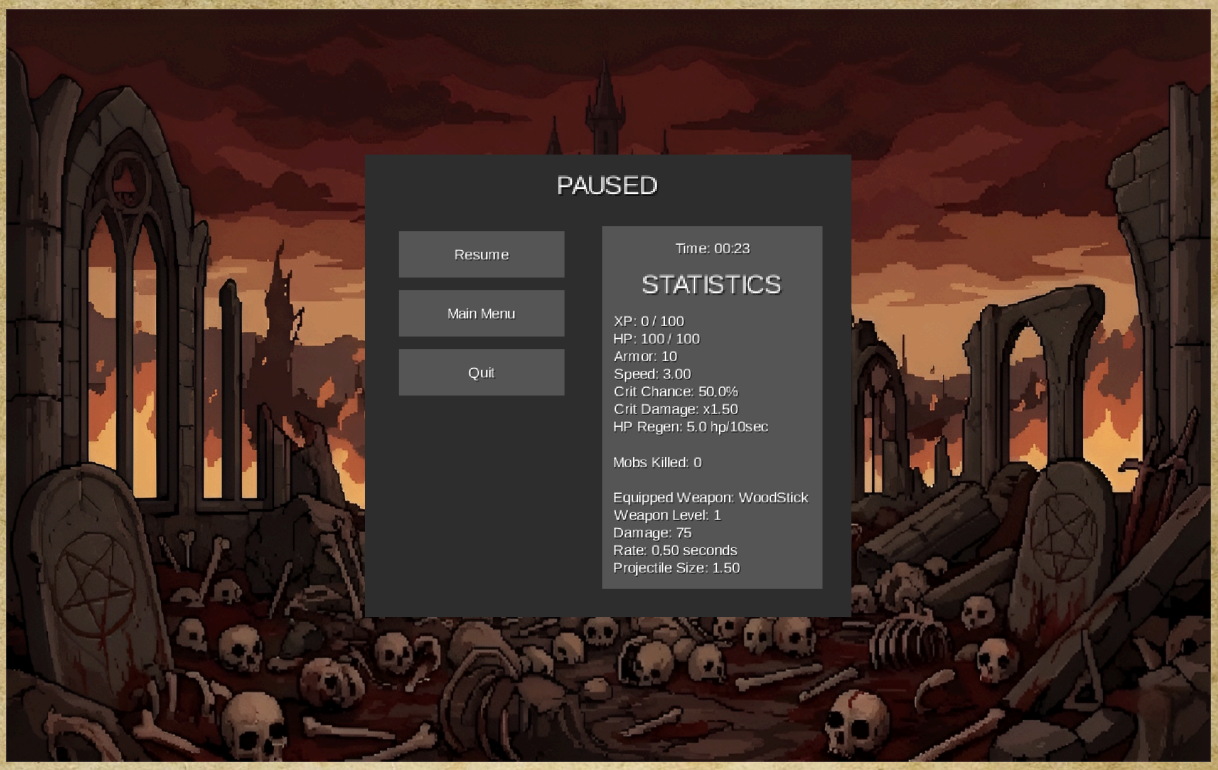


4.4.4. Menu de Pause

Permet au joueur de s'interrompre sans perdre sa progression actuelle.

- **Déclencheur :** S'active par l'appui sur la touche Échap (Esc) durant le jeu.
- **Fonctionnalités :**
  - Gèle la boucle de jeu.
  - **Statistiques :** Affiche le récapitulatif détaillé des stats actuelles du personnage.
  - **Boutons :** "Reprendre la partie", "Retour au Menu Principal" ou "Quitter le jeu".

- **Référence visuelle :**

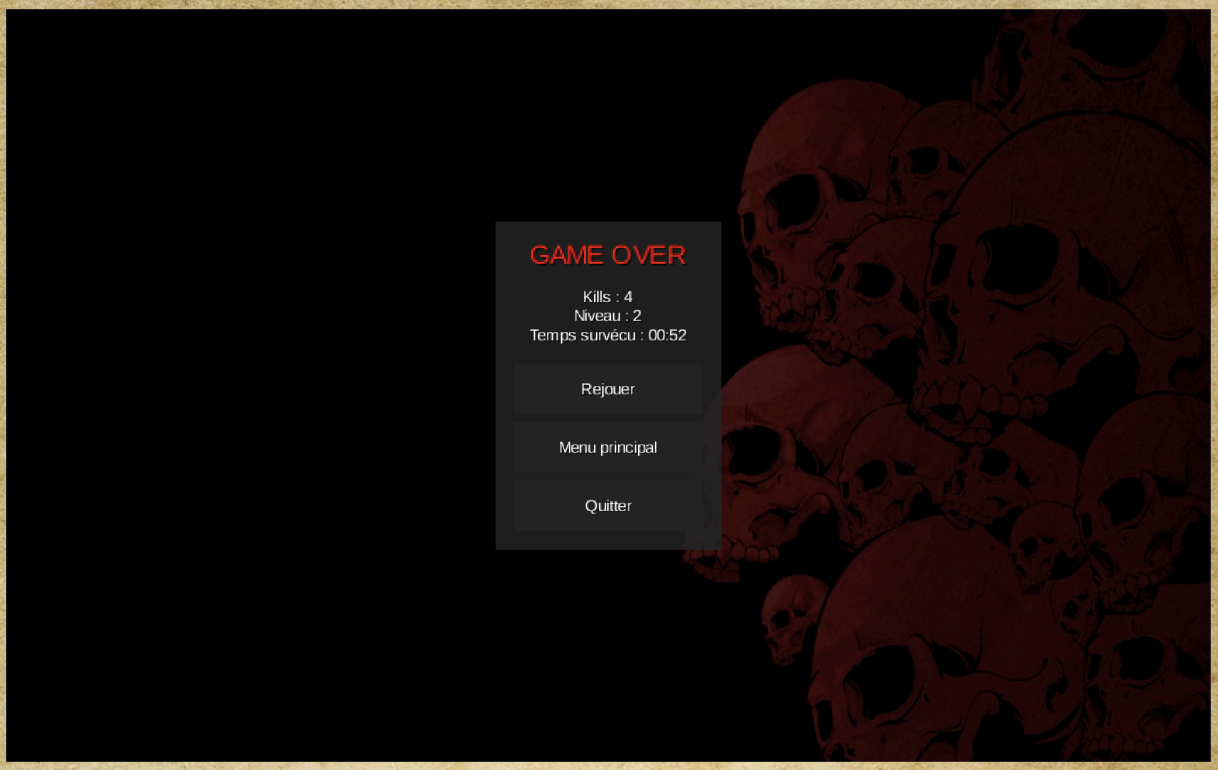


4.4.5. Écran de Fin de Partie (Victoire / Défaite)

Cet écran conclut la session de jeu. La structure est identique pour la victoire ou la défaite, seule l'ambiance visuelle change.

- **Déclencheurs :**
  - **Mort :** Les points de vie du joueur tombent à 0.
  - **Victoire :** Le joueur survit jusqu'à la fin du temps imparti (ex: 30 minutes) ou tue le boss final.
- **Informations affichées :**
  - Résultat (Victoire ou Game Over).
  - Temps total survécu.
  - Nombre total d'ennemis vaincus.
- **Visuel & Feedback :**
  - Utilise la même image de fond de base.
  - **Défaite :** Application d'un filtre rouge sombre.
  - **Victoire :** Application d'un filtre vert.

- Référence visuelle :



4.5 Son

Les sons ont été ajoutés à travers LibGDX Audio.music, avec un système de gestion centralisée pour faciliter l'équilibrage sonore. Les musiques ont été composées par un contact de Alexandre.

5.0 Documentation Technique

5.1 Architecture Logicielle et Environnement

5.1.1 Moteur et Rendu

- **Langage de programmation : Java.**
- **Framework principal : libGDX.** Il sera utilisé comme bibliothèque multiplateforme (Desktop, etc.) pour gérer la boucle de jeu, le rendu graphique, la gestion des entrées (inputs) et l'audio.
- **API Graphique : OpenGL.** Le rendu 2D sera géré via les abstractions de libGDX (ex: SpriteBatch) basées sur OpenGL (ES).

5.1.2 Outils de Production

- **Éditeur de niveaux : Tiled** (https://www.mapeditor.org/).
- **Format des niveaux :** Le niveau (map) sera créé avec Tiled et exporté au format .tmx. Le moteur devra implémenter un "parser" pour charger et interpréter ces fichiers (ex: via les helpers libGDX TmxMapLoader).

5.1.3 Gestion des Assets

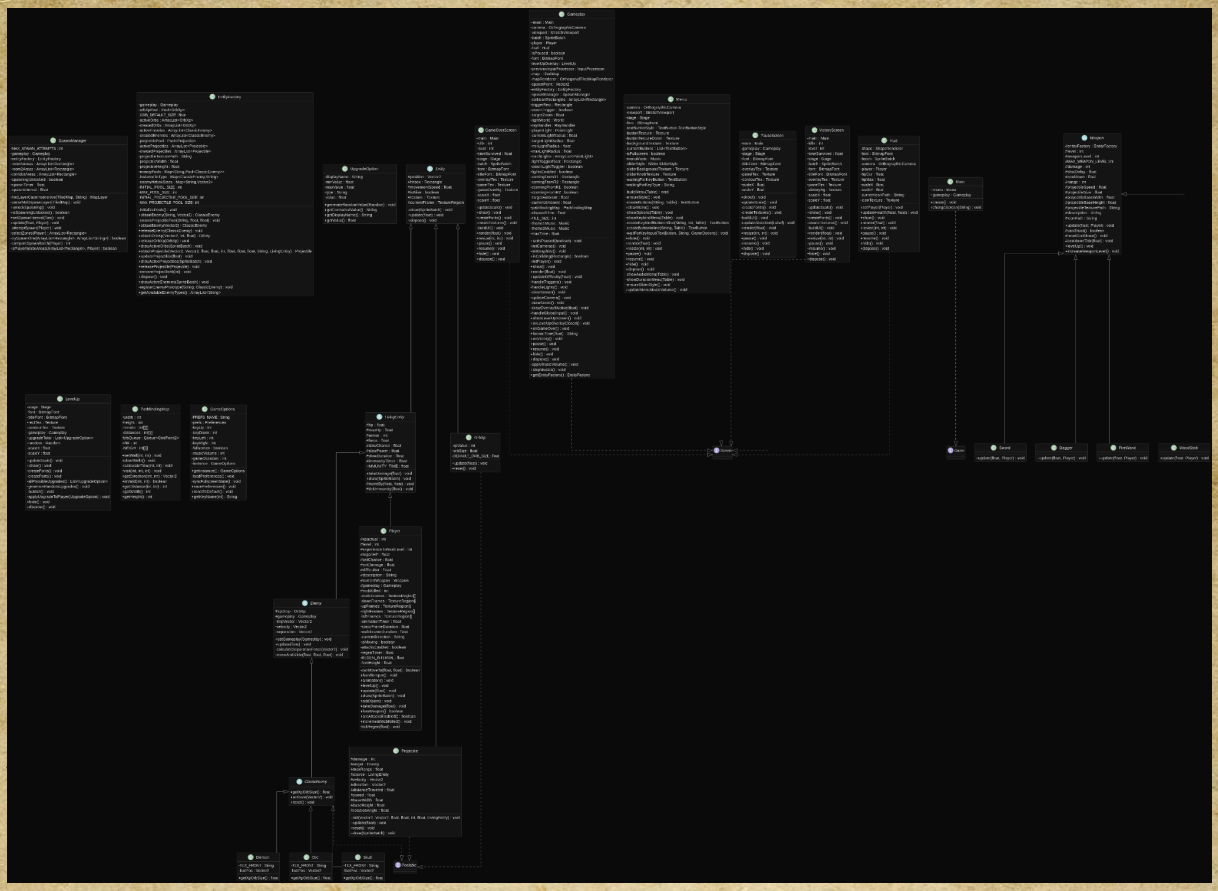
Statut des assets : Placeholders (Ressources temporaires).

- **Source :** Les ressources graphiques (sprites, UI) et sonores sont issues de banques libres de droit ou créées à la main.

Outils de création des skins :

- https://www.pixilart.com/
- https://www.gimp.org/downloads/

5.1.4 Système de Build et Dépendances

- **Outil de build : Gradle.**
- **Rôle :** Gradle gérera l'intégralité du cycle de vie du projet, incluant :
  - La gestion des dépendances (libGDX et ses extensions).
  - La compilation du code source.
  - La création des livrables ("builds") pour les différentes plateformes ciblées (ex: JAR exécutable pour Desktop).

5.2 - Diagramme UML