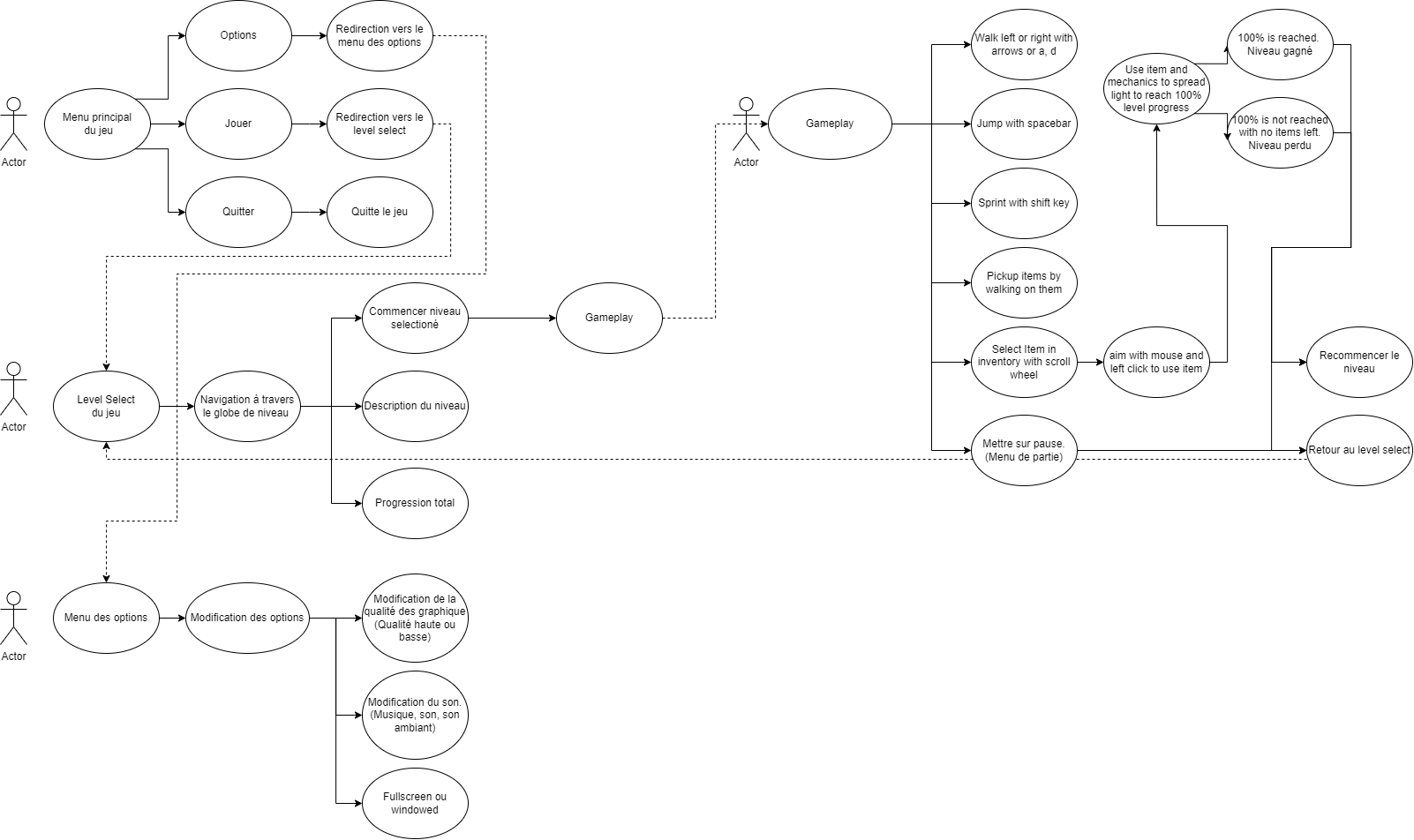
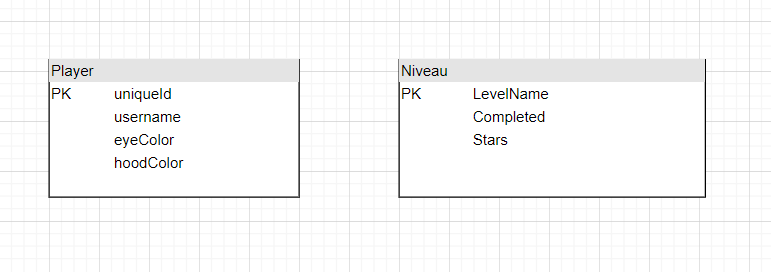
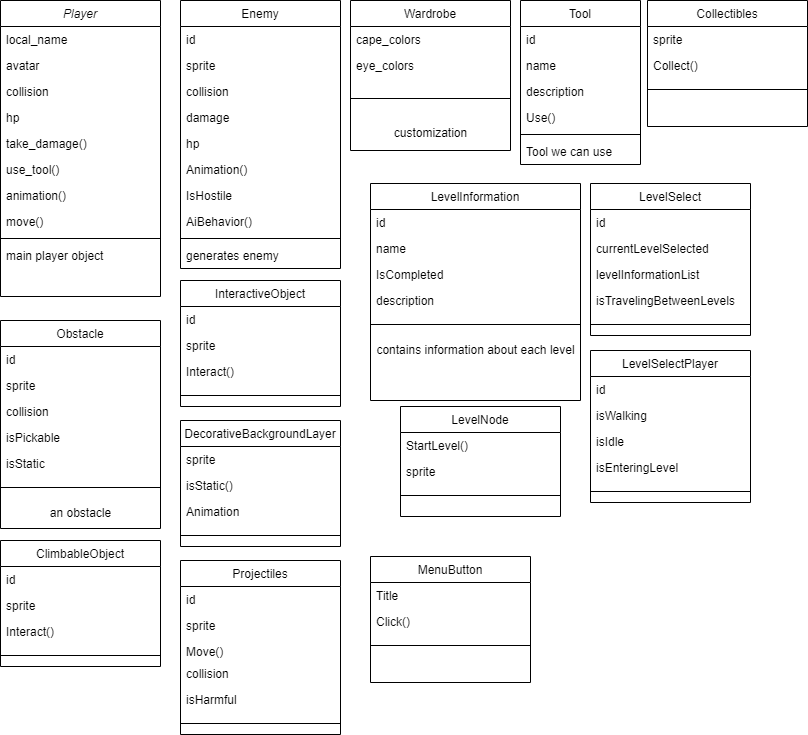
Shémas de cas d’utilisation



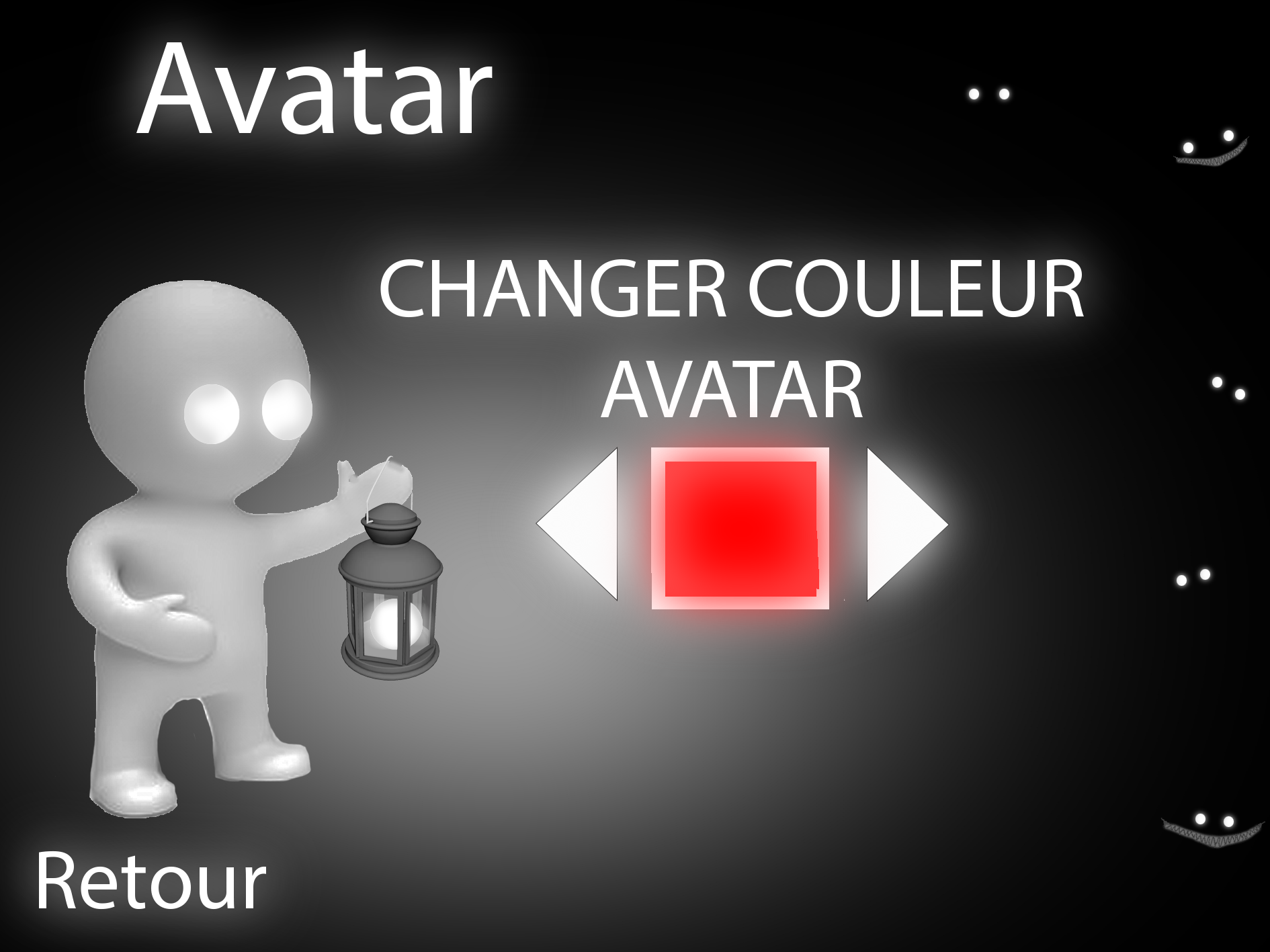
Schémas de Base de donné

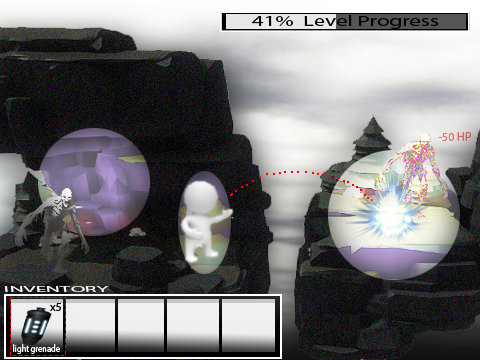


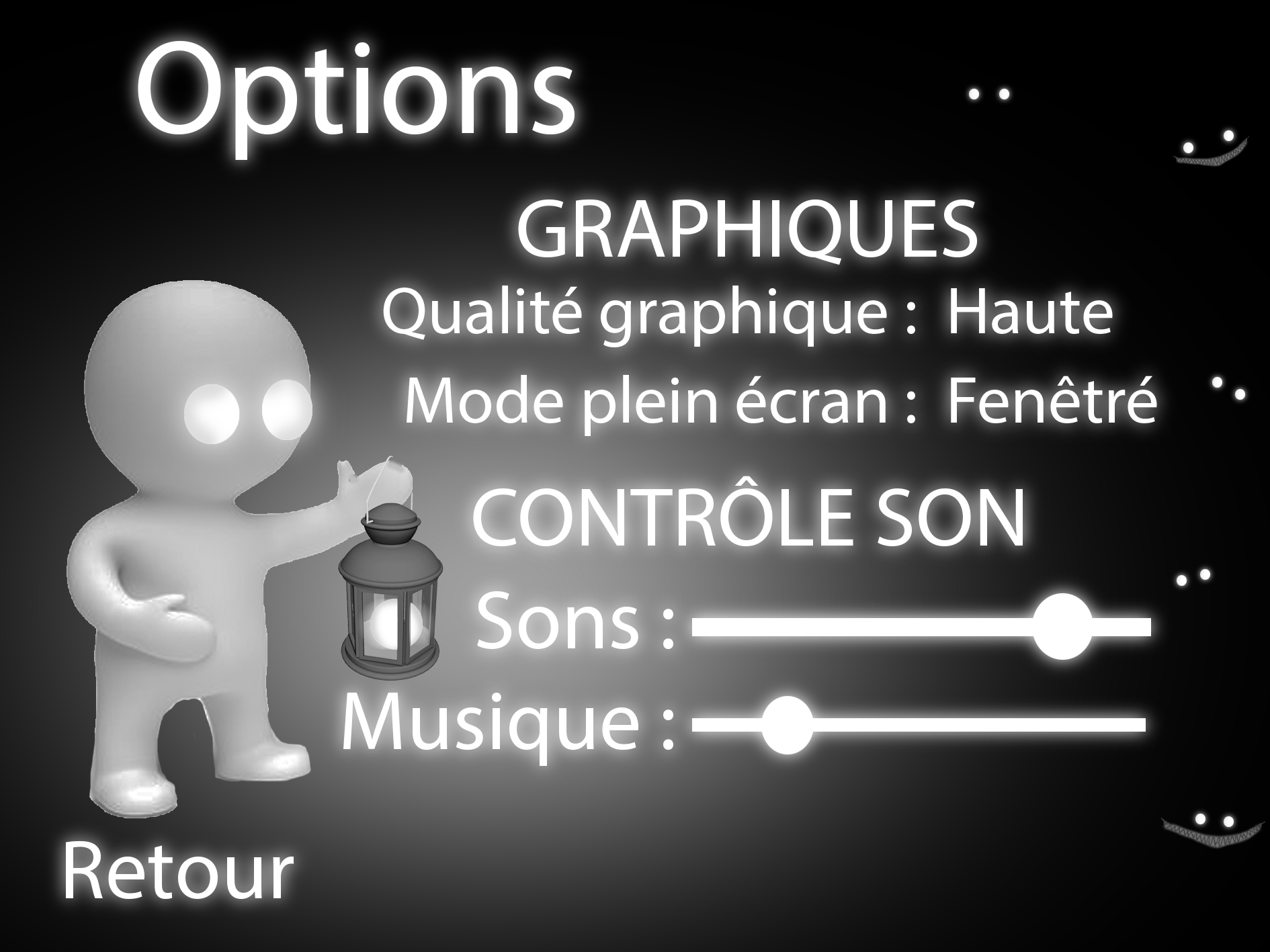
Shémas de Classes

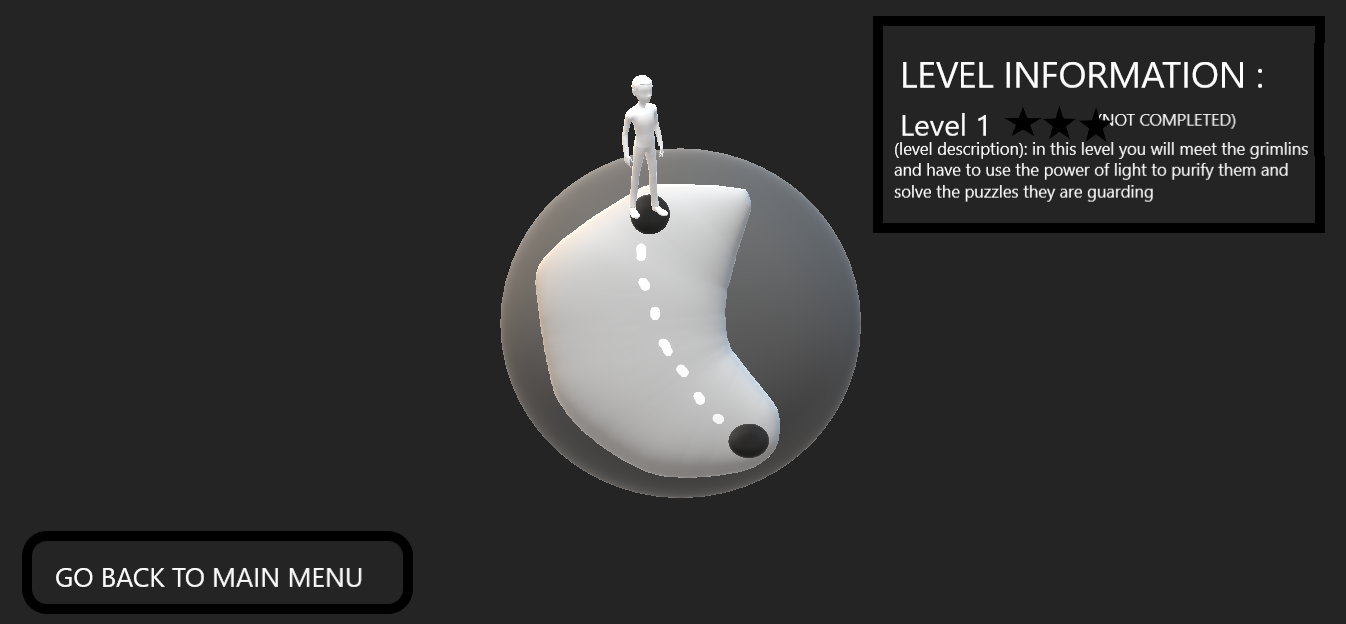


Maquettes









Elements de conception:

**Structures de données:**

**1. Un dictionnaire:**  
 Nous allons utiliser un dictionnaire qui contiendra l'ensemble des méthodes différentes de déplacement en dépendant du type de l’ennemi. La clé sera le type de l'ennemi et la valeur sera la méthode de déplacement. Cela évitera de faire un if else ou switch case a chaque fois que la méthode sera appelée.

**2. Une Liste:**

Nous allons avoir une liste d' id de tous les ennemis présents dans un niveau pour pouvoir effectuer des opérations sur l’ensemble de ces ennemis à l'aide de boucles for.

**3. Un Stack:**

Nous allons intégrer des stacks pour faire la logique d’un certain puzzle de triage de disques (de type Tower of Hanoi)

**4.Doubly linked list:**

Nous allons créer une liste doublement chaînée qui stocke les informations pour chaque niveau du jeu pour imprimer la progression dans le lobby des jeux. L'utilisateur ne peut parcourir la carte que par ordre de niveaux. Pour modifier les informations de niveau affichées à l'écran, une liste doublement chaînée est appropriée.

**Patron de conception:**

**1. game loop:**

Ce patron qui est déjà intégré à Unity va pouvoir update tous les éléments a chaque tic de jeu et pouvoir faire des événement asynchrones sans l’usage de threads.

**2. Observers:** Ce patron est déjà intégré à Unity sur les touches des contrôles du jeu pour faire réagir le jeu au touche du clavier

**3. Singleton:**

Nous allons créer une classe de type singleton qui contient tout le garde robe (l’ensemble des cosmétiques possiblement utilisable par l’utilisateur.)

**Expression régulière :**

Utilisation de regex pour vérifier que le nom d’utilisateur relier à la sauvegarde ne contient seulement que des lettres et des chiffres (pas d'autres caractères spéciaux)

**Algorithme**:

Comportement IA des ennemi donc algorithme de détection des environ (ray-tracing)

**Mathématique:**

L’utilisation de vecteurs pour tous variable de position 2 dimensionnel et 3 dimensionnel.

**Librairie générique:**

Génération de asset 3d de solide platoniciens de la couleur et texture voulu pour la création d’un collectible (gem du chaos) qui va utiliser cette librairie pour ça texture. Des modèle 3d de solide platoniciens peuvent être utilisés pour d'autres projets donc nous considérons cette librairie comme générique.