**MOTEUR DE JEU**

Les étapes à réaliser :

* Définir les acteurs ainsi que leurs besoins
* Réaliser un digramme de cas d’utilisation
* Réaliser un diagramme d’activité
* Réfléchir au stockage de données

L’architecture se décompose généralement en trois niveaux : la présentation, la logique métier et la persistance des données.

Au niveau de la présentation, le moteur de jeu gère l'interface utilisateur, les graphiques et l'interaction avec le joueur. Les acteurs impliqués à ce niveau sont les concepteurs d'interface utilisateur et les joueurs eux-mêmes, ayant des besoins axés sur l'expérience visuelle, la convivialité et la réactivité du jeu.

La logique métier, quant à elle, englobe les règles du jeu, l'intelligence artificielle, la gestion des événements et autres fonctionnalités cruciales. Les développeurs de jeux et les concepteurs de niveaux sont les acteurs clés à ce niveau, cherchant à satisfaire des besoins tels que des mécaniques de jeu fluides, des ennemis réactifs et une progression engageante.

Enfin, la persistance des données gère le stockage et l'accès aux informations du jeu, comme les scores, les paramètres et les sauvegardes. Les administrateurs de bases de données et les analystes de données s'occupent des exigences liées à ce niveau, telles que la fiabilité des données, la récupération après panne et la gestion efficace de grandes quantités d'informations.

Les différents acteurs et leurs besoins :

* Joueurs
* Se connecter aux serveurs
* Personnaliser ses préférences
* Accéder à une partie
* Acheter des extensions
* Développeurs
  + Ajouter des extensions
  + Ajouter des personnalisations
  + Améliorer les performances des serveurs
  + Corriger des bugs

Toutes les actions mènent ici :

* + Tester le jeu

**Diagramme de cas d’utilisation :**

Une image contenant texte, diagramme, capture d’écran, cercle

Description générée automatiquement

**Diagramme d’activité :**

On a décidé de séparer les joueurs des développeurs car ils sont sur deux terrains différents.

Les joueurs vont jouer basiquement tandis que les développeurs auront leur terrain de jeu afin de tester les nouvelles fonctionnalités implémentées et de corriger certains bugs.

Une image contenant diagramme, texte, capture d’écran, ligne

Description générée automatiquement

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

**Stockage des données :**

Stockage des fichiers binaires (maps, textures, modèle 3D) :

Système de stockage distribué pour gérer les fichiers binaires volumineux, tels que Google Cloud Storage. Ce service offre une mise à l'échelle facile pour le stockage de gros fichiers.

Pourquoi : prix moi élevé, et capacité de stockage bcp plus intéressante

Base de données pour les données structurées (armes, menus, etc.) :

Base de données qui prend en charge la mise à l'échelle horizontale pour gérer un grand nombre de requêtes. Une base de données NoSQL comme MongoDB peut être un bon choix.

Sharding : MongoDB offre une fonctionnalité de sharding qui permet de distribuer les données sur plusieurs serveurs. Cela permet de répartir la charge de travail et d'assurer une réponse rapide aux requêtes, même lorsque la quantité de données augmente.

Réplication : Pour garantir une haute disponibilité et une tolérance aux pannes, MongoDB prend en charge la réplication des données. Les données sont copiées sur plusieurs nœuds, ce qui permet au système de continuer à fonctionner même en cas de défaillance d'un nœud.