

Moonlighter



Analyse théorique :

1. De quel type de jeu s'agit-il ?

Moonlighter est un jeu d'action-RPG avec des éléments de rogue like et de gestion. Il mélange l'exploration de donjons et la gestion d'une boutique, ça permet de jongler l'aventure et la stratégie.

2. Après une heure de jeu, nommez les différentes mécaniques de gameplay rencontrées.

- Exploration des donjons avec des combats en temps réel.**

- **Système de loot où on ramasse des objets en battant des monstres ou en ouvrant des coffres.**
- **Gestion de l'inventaire, avec des objets qui ont des effets de positionnement.**
- **Vente d'objets dans la boutique, avec un système de tarification basé sur la demande des clients.**
- **Amélioration du personnage en achetant des armes, armures et potions.**
- **Progression du village, en attirant de nouveaux marchands qui aident à améliorer l'équipement.**

3. Le jeu peut être séparé en deux phases de gameplay bien distinctes. Détaillez chacune d'elles.

- **Exploration des donjons : Le joueur combat des monstres, ramasse du butin et essaie d'atteindre le boss de chaque donjon.**
La mort entraîne la perte d'une partie du butin, ce qui pousse à bien gérer son inventaire et ses ressources.
- **Gestion de la boutique : Le joueur fixe les prix des objets récupérés, observe les réactions des clients**

et ajuste les tarifs pour maximiser ses gains. Il doit aussi gérer son stock et améliorer sa boutique pour attirer plus de clients.

4. La génération des donjons se fait selon des règles spécifiques auxquelles s'ajoute la randomisation. Comment appelle-t-on ce type de génération ? Expliquez les règles de génération de Moonlighter.

La génération des donjons suit un modèle procédural avec une randomisation partielle, ce qui signifie que les salles sont générées aléatoirement selon certaines règles. Ce type de génération est appelé "roguelike procedural generation".

- Les donjons sont constitués de salles interconnectées.**
- Chaque salle est choisie parmi un ensemble prédéfini de salles possibles.**
- Les ennemis et les objets sont placés de manière semi-aléatoire pour garder un équilibre dans la difficulté.**

5. L'inventaire du héros possède certaines mécaniques intéressantes de positionnement et de bonus sur cases. Expliquer en termes de programmation comment est-il possible de mettre en place ce système.

L'inventaire du héros a des mécaniques spéciales : certains objets ont des effets qui obligent le joueur à les placer à des endroits spécifiques (exemple : un objet détruit celui à sa gauche si mal positionné).

- **En programmation, cela peut être géré avec un tableau à deux dimensions représentant la grille de l'inventaire.**
- **Chaque case du tableau stocke un objet et ses propriétés (ex : restrictions de position).**
- **Une vérification est effectuée lorsqu'un objet est déplacé pour voir s'il respecte ses contraintes.**

6. La vente des objets dans le magasin est définie selon plusieurs paramètres. Expliquez les et détaillez, étapes par étapes (conditions, pourcentage, inventaire, rabais, ...), le déroulement d'une vente à un client.

Chaque objet a un prix de base mais la demande des clients influence la valeur réelle.

Le joueur fixe un prix et observe la réaction des clients (satisfaits, indifférents, trop cher, ou super content donc trop bas).

La demande varie avec le temps : si un objet est trop vendu, sa valeur diminue.

Le client choisit l'objet, se dirige à la caisse et valide son achat si le prix lui convient.

Possibilité d'appliquer des rabais pour écouler du stock.

7. Personnel : le jeu vous a-t-il plu ? Pourquoi ?

Oui, j'ai beaucoup aimé Moonlighter ! J'ai trouvé le mélange entre donjons et gestion super intéressant.

Le fait de devoir choisir entre continuer un donjon ou rentrer vendre ses objets ajoute une vraie stratégie.

De plus, les graphismes en pixel art et la musique sont vraiment immersifs.

Pratique :

1. Définissez, selon vous, une des mécanique de gameplay principale du jeu, et reproduisez là sous le moteur de jeu Unreal Engine 5 (vous pouvez faire cette mécanique sur un projet de n'importe quel type, en 2D ou 3D, du moment que l'on "ressent" une similitude avec la mécanique originale).

J'ai choisi la mécanique qui permet de nous téléporter d'une zone de donjon à l'autre.

Bonus : J'ai ajouté les attaques et les roulades comme dans le jeu pour renforcer l'immersion.

2. Décrivez, étape par étape, comment vous avez mis en place cette mécanique.

- J'ai créé un static mesh, j'ai choisi une porte comme static mesh.**

- J'ai ensuite créé une box collider qui permet de détecter quand le joueur entre en contact avec.
- J'ai ensuite utilisé un cast to BP qui me permet de connecter le joueur a la porte pour que le code se déclenche quand il passe dans le box collider.
- J'ai ensuite récupéré les informations de la position du joueur grâce à son transform location.
- J'ai terminé avec un set actor location and rotation que j'ai coché sur "TP" ce qui me permet de choisir la position du joueur une fois qu'il passera dans la box collider.
- J'ai déplacé dans la scène ce qui permet de set la nouvelle position du player.

3. Expliquez en quoi cette mécanique est primordiale tout au long du jeu.

Cette mécanique est une des mécaniques les plus importante dans le jeu, cela permet au joueur de se téléporter dans les différentes salle du donjon en étant fluide et sans casser le rythme du jeu.