

Moonlighter

OCAMPO Esteban G2

Analyse théorique :

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

C'est un jeu en pixel art type action RPG avec une gestion des boutiques.

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

Après une heure de jeu j'ai trouvé les mécaniques suivantes :

Exploration des donjons, collecte d'objets, gestion de l'inventaire, retour au village, gestion de la boutique, Interaction avec les PNJ.

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

Moonlighter se distingue par deux phases de gameplay bien distinctes, chacune avec son propre rythme et ses défis.

Phase d'exploration et de combat (Donjons) Cette phase nous plonge dans la peau de Will, un marchand qui rêve d'aventure. On peut explorer des donjons mystérieux et dangereux.

Phase de gestion de la boutique (Village) Après l'adrénaline des donjons, nous revenons au calme du village. Ici, on devient marchand, gérant la boutique avec soin.

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 dans Moonlighter est de la génération procédurale. Les règles sont les suivantes.

Structure en salles interconnectées

Les donjons sont composés de salles carrées ou rectangulaires reliées entre elles par des portes.

Chaque salle peut contenir des ennemis, des pièges, des coffres, des ressources ou des éléments décoratifs.

Thèmes et styles visuels

Chaque donjon a un thème visuel et mécanique distinct (par exemple, un donjon forestier, un donjon désertique, etc.).

Les ennemis, les pièges et les objets sont adaptés au thème du donjon, ce qui ajoute de la variété et de la cohérence.

Niveaux de difficulté progressifs

Les donjons sont divisés en étages, chaque étage étant plus difficile que le précédent.

Plus tu descends profondément, plus les ennemis deviennent coriaces et les pièges sont plus dangereux.

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.

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.

Dans Moonlighter, la vente des objets dans le magasin est un mécanisme central qui combine gestion, stratégie et observation.

Prix de base de l'objet

Chaque objet a un prix de base, qui est déterminé par sa rareté, son utilité et son type (arme, armure, ressource, etc.).

Réaction du client

Les clients réagissent au prix que tu fixes.

Leur réaction peut être :

Heureux : Le prix est bien en dessous de leur budget.

Neutre : Le prix est raisonnable.

Mécontent : Le prix est trop élevé.

Ces réactions te donnent des indices pour ajuster tes prix.

Demande du marché

Certains objets sont plus recherchés que d'autres. Par exemple, une potion de soin pourrait se vendre plus facilement qu'un objet décoratif rare.

Bonus de placement

Si tu places un objet sur une case de l'inventaire qui offre un bonus (comme une augmentation de valeur), cela peut influencer son prix de vente.

Rabais et promotions

Tu peux appliquer des rabais temporaires pour attirer plus de clients ou vider ton stock rapidement.

Déroulement d'une vente à un client

Voici les étapes de ce qui se passe lorsqu'un client entre dans ton magasin et souhaite acheter un objet

Étape 1 : Le client entre dans le magasin

Un client arrive dans ta boutique et se dirige vers un objet exposé.

Le client a un budget caché et une tolérance au prix qui déterminent s'il achètera l'objet ou non.

Étape 2 : Observation de la réaction du client

Lorsque le client s'approche de l'objet, il affiche une réaction en fonction du prix que tu as fixé :

Émoticône heureuse : Le prix est bien en dessous de son budget.

Émoticône neutre : Le prix est acceptable.

Émoticône mécontente : Le prix est trop élevé.

Étape 3 : Ajustement du prix (optionnel)

Si le client semble mécontent, tu peux ajuster le prix avant qu'il ne décide d'acheter ou de partir.

Par exemple, si tu baisses le prix de 10 %, le client pourrait devenir plus intéressé.

Étape 4 : Décision d'achat

Le client décide d'acheter ou non en fonction de :

Le prix fixé : S'il est dans sa fourchette de tolérance, il achète.

La demande du marché : Si l'objet est très recherché, le client est plus susceptible de l'acheter même à un prix élevé.

Les bonus de placement : Si l'objet est sur une case bonus, sa valeur perçue peut augmenter.

Étape 5 : Transaction

Si le client achète l'objet :

Tu reçois l'argent correspondant au prix fixé. L'objet est retiré de ton inventaire.

Si le client ne l'achète pas : Il part sans rien acheter, et tu perds une opportunité de vente.

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

Ce que j'apprécie particulièrement dans les jeux vidéo, c'est la liberté qu'ils offrent, et Moonlighter nous donne cette qualité. Le jeu nous propose une exploration variée et captivante des donjons, la possibilité de vendre les objets à notre guise en fixant nous-mêmes les prix, ainsi que des choix stratégiques pour améliorer nos armes et développer notre base. Cette combinaison d'éléments donne une véritable liberté au joueur, permettant à chacun de jouer à son rythme et selon ses préférences. C'est sans doute pour ces raisons que j'ai tant apprécié ce titre.

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 de recréer la mécanique d'inventaire qui sera activée avec la touche "E" dans lequel on peut ramasser des items et les voir dans notre inventaire.

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

Étape 1 : Créer un Blueprint structure dans laquelle on va renseigner les variables qui vont nous servir à composer notre item.

Étape 2 : Ensuite on va créer un datatable dans lequel on va renseigner le structure qu'on vient de créer, ici on aura toute la data des items.

Étape 3 : Après on va commencer à créer notre actor component qui va nous permettre de rendre le système fluide et ici on codera toute la logique de l'inventaire en créant différentes fonctions et variables.

Étape 4 : Une fois la logique du code codé on va créer un widget et ici on va créer l'UI de notre inventaire.

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

Dans le jeu on doit recevoir constamment des items, on doit faire constamment des améliorations et le système d'inventaire nous permet de gérer tout ça d'une façon assez confortable.

4. Pourquoi cette mécanique vous a-t-elle plu ?

Cette mécanique m'a plu car elle est indispensable pour ce type des jeux et Moonlighter l'a créé de telle façon que c'est agréable à voir et intuitif à utiliser.