

Plan de travail - Jour 4

On continue avec la trame et on passe aux étapes suivantes pour étoffer ton RPG. Voici ce que tu vas faire demain :

Étape 1 : Gestion des ennemis et des combats

Objectif : Implémenter un système simple de combat et une gestion des ennemis.

1. Création d'une classe `Ennemi` :

- Attributs à inclure :
 - `nom` : le nom de l'ennemi.
 - `pv` : points de vie.
 - `attaque` : valeur d'attaque.
 - `zone` : zone où l'ennemi peut apparaître.
- Méthodes :
 - `attaquer()` : inflige des dégâts au joueur.
 - `recevoir_dégats(dégâts)` : réduit les PV de l'ennemi.

2. Ajout d'une liste d'ennemis dans chaque zone :

- Exemple : loups dans les plaines, gobelins dans la forêt, etc.
- Génération aléatoire d'un ennemi lorsqu'un combat commence.

3. Système de combat :

- Affiche les PV du joueur et de l'ennemi.
- Permet de choisir entre attaquer ou fuir :
 - **Attaquer** : inflige des dégâts à l'ennemi et reçoit des dégâts en retour.
 - **Fuir** : termine le combat mais avec une pénalité (perte d'or, PV, etc.).

Étape 2 : Loots et récompenses

Objectif : Ajouter un système de butin pour rendre les combats gratifiants.

1. Ajout d'un loot pour chaque ennemi :

- Chaque ennemi peut laisser tomber un objet ou de l'or après sa défaite.
- Exemples :
 - Loup → Peau de loup.
 - Gobelins → Petit sac d'or.
- Les loot sont ajoutés automatiquement à l'inventaire du joueur.

2. Création de méthodes dans `Perso` pour gérer les loot :

- `ajouter_objet(objet, quantité)` : ajoute un item à l'inventaire.
 - `ajouter_or(montant)` : augmente l'or du joueur.
-

Étape 3 : Améliorations système de combat

Objectif : Faire le visuel d'un combat

1. Initialisation de Pygame

Avant tout, tu dois initialiser Pygame et créer une fenêtre de jeu où tu affiches tout ça. L'idée est de diviser la fenêtre en deux parties : le haut pour le combat et le bas pour les options.

2. Diviser la fenêtre en deux zones

On va diviser l'écran en deux parties :

- **Haut (Combat)** : Affiche les informations de combat, les personnages et les points de vie.
- **Bas (Choix d'action)** : Affiche les options disponibles (attaquer, fuir).

3. Affichage du Combat

Dans la partie supérieure, on va afficher les éléments suivants :

- Le nom de ton personnage et de l'ennemi.
- Les points de vie actuels de chaque personnage.
- Les actions possibles dans le bas de l'écran (attaquer ou fuir).

4. Création des boutons

Les boutons pour les choix (Attaquer, Fuir) seront en bas. On peut utiliser des rectangles pour simuler des boutons.

5. Logique de Combat

Pour que le jeu réagisse correctement quand tu choisis d'attaquer ou de fuir, tu auras besoin de gérer les événements Pygame (clics sur les boutons) et mettre à jour l'affichage après chaque action.

1.

Ressources nécessaires

- Une liste d'ennemis par zone.
- Idées de lots et récompenses.
- Tes classes actuelles ([Perso](#), [Quete](#), etc.) pour interagir avec les nouvelles fonctionnalités.