

Plan de travail pour demain :

1. Vérification du code actuel :

- **Objectif** : Relire et tester le fichier `main.py` pour s'assurer que tout fonctionne bien, en particulier les déplacements du joueur et la gestion des événements.
- **Action** : Lancer le jeu et vérifier les déplacements du joueur (à l'aide des touches `Z`, `S`, `Q`, `D`). Assurer que l'affichage de la fenêtre et les interactions sont corrects.

2. Amélioration des déplacements du joueur :

- **Objectif** : Ajouter des limites à l'écran pour éviter que le joueur sorte de la fenêtre de jeu.
- **Action** : Modifier le code pour ajouter des conditions qui empêchent le joueur de sortir des bords de l'écran (par exemple, empêcher que `joueur_x` ou `joueur_y` deviennent plus grands ou plus petits que les dimensions de l'écran).

3. Ajout d'une image pour le personnage :

- **Objectif** : Remplacer la gestion de l'avatar du joueur par une image.
- **Action** : Décommenter et améliorer la ligne de code où l'image du personnage est chargée et ajoutée à l'écran. Veiller à bien ajuster l'image pour qu'elle se déplace correctement avec les coordonnées `joueur_x` et `joueur_y`.

4. Implémentation de la gestion des objets et de l'inventaire :

- **Objectif** : Commencer à poser les bases du système d'inventaire, y compris la gestion des objets.
- **Action** :
 - Créer une classe pour gérer les objets (par exemple, `Item`), qui contient les attributs de base comme le nom, l'image, et les effets (potion, arme, etc.).
 - Tester l'ajout d'objets à un inventaire, en créant une simple liste ou dictionnaire pour les stocker.
 - Ajouter une interaction avec l'inventaire (par exemple, une touche pour afficher l'inventaire à l'écran).

5. Structurer les classes de personnages :

- **Objectif** : Commencer à organiser les classes de personnages (joueur, loup, ennemis, etc.) dans des fichiers séparés pour mieux structurer le projet.
- **Action** : Crée un fichier `player.py` et commencer à y transférer la classe `Joueur` (avec les attributs comme les points de vie, l'inventaire, etc.).
 - **Classe Joueur** : Ajouter des méthodes pour gérer la vie, les attaques, et l'inventaire.
 - Ajouter une interaction simple avec l'inventaire (par exemple, ramasser un objet ou utiliser une potion).

6. Gestion des collisions et interactions :

- **Objectif** : Créer un système simple de collisions pour empêcher le joueur de traverser certains objets ou d'entrer dans des zones interdites.
- **Action** : Ajouter des objets comme des murs ou des obstacles dans la carte et tester les collisions avec le joueur. Par exemple, utiliser des rectangles pour détecter si le joueur entre en contact avec un objet ou une zone spécifique.

7. Préparer les bases du système de quête :

- **Objectif** : Initier une structure pour gérer les quêtes dans le jeu.
- **Action** : Créer une classe **Quest** (dans un fichier séparé, par exemple `quest.py`) avec des attributs comme le titre, la description et les objectifs de la quête.
 - Tester l'affichage de la quête sur l'écran.
 - Ajouter une méthode dans la classe **Joueur** pour afficher une quête active et la suivre.

Résumé des tâches à accomplir demain :

1. Vérification du code actuel et tests des déplacements du joueur.
2. Ajouter des limites aux déplacements du joueur (empêcher de sortir de la fenêtre).
3. Ajouter une image pour le personnage et tester les déplacements avec l'image.
4. Commencer à implémenter la gestion d'un inventaire et des objets.
5. Structurer les classes de personnages (joueur, loup, ennemis).
6. Ajouter un système simple de gestion de collisions.
7. Commencer à préparer la gestion des quêtes dans le jeu.

Estimation du temps pour chaque tâche :

- **Vérification du code actuel** : 30 min
- **Amélioration des déplacements** : 30 min
- **Ajout d'une image pour le joueur** : 45 min
- **Gestion d'inventaire et objets** : 1h
- **Structuration des classes de personnages** : 1h
- **Gestion des collisions** : 45 min
- **Préparer les bases du système de quête** : 30 min

Total estimé pour la journée : environ **5 heures**.

Astuce : Commence par les tâches les plus simples (comme les déplacements et l'ajout d'images) et passe ensuite aux aspects plus complexes (comme l'inventaire et la gestion des collisions). Cela te permettra de garder le rythme tout en avançant.