LÉGENDE. 🛡️🎮  
Tu viens de passer le cap où les créateurs s’arrêtent, mais toi tu **continues à pousser la porte du Multivers**.  
Bienvenue dans le **Pack de Quêtes – Semaine 6 : Les Portes du Monde Persistant** 🧩📂  
Cette semaine, on va explorer :

* La **navigation entre plusieurs scènes**
* Les **systèmes de sauvegarde**
* Une **progression utilisateur** (inventaire, actions, états mémorisés)
* Et comment préparer ton projet à devenir un **jeu complet**

**🎒 PACK DE QUÊTES – SEMAINE 6 : *Connexion Persistante***

**⚔️ Jour 26 : Le Codex des Scènes Multiples**

**🎯 Objectif : Créer une expérience composée de plusieurs scènes Unity**

**Étapes :**

1. Crée deux scènes :
   * Scene\_Intro (titre, bouton “Commencer”)
   * Scene\_Exploration (ton monde VR)
2. Dans Scene\_Intro, ajoute un bouton :
   * Dans l’événement OnClick, lie une fonction SceneManager.LoadScene("Scene\_Exploration")
3. Dans Build Settings, ajoute toutes tes scènes au build (Ctrl + Shift + B)
4. Test : clique sur le bouton → ta scène change ✅

💡 Tip : active using UnityEngine.SceneManagement; dans ton script.

**⚔️ Jour 27 : Le Sceau de la Sauvegarde**

**🎯 Objectif : Ajouter une fonction de sauvegarde d’état (objets déplacés, puzzles résolus, etc.)**

**Étapes :**

1. Crée un script SaveManager.cs :

csharp

CopyEdit

public static class SaveManager {

public static Vector3 lastPosition;

}

1. Lors d’un événement (ex : joueur touche un cube), enregistre :

csharp

CopyEdit

SaveManager.lastPosition = cube.transform.position;

1. Dans ta nouvelle scène, applique cette valeur à l’objet :

csharp

CopyEdit

cube.transform.position = SaveManager.lastPosition;

🎓 Ce n’est pas une vraie sauvegarde sur disque, mais une **sauvegarde en mémoire** → base pour un système plus complet plus tard.

**⚔️ Jour 28 : Le Livre des Inventaires**

**🎯 Objectif : Créer un système d’inventaire simple**

**Étapes :**

1. Crée un script Inventory.cs :

csharp

CopyEdit

public class Inventory : MonoBehaviour {

public List<string> objects = new List<string>();

public void AddItem(string id) {

objects.Add(id);

}

public bool HasItem(string id) {

return objects.Contains(id);

}

}

1. Quand le joueur touche un objet VR, ajoute-le à l’inventaire.
2. Teste : si l’objet est dans l’inventaire, une porte peut s’ouvrir.

📚 Tu viens d’ajouter une **logique de progression** à ton jeu ! (clef > porte > nouvelle zone)

**⚔️ Jour 29 : La Stèle des États Mémorisés**

**🎯 Objectif : Permettre à ton jeu de retenir les choix ou actions du joueur**

**Étapes :**

1. Crée un **GameManager** en DontDestroyOnLoad
2. Il conserve des booléens ou des IDs (ex : "puzzle1Done")
3. Selon ces états, tu **actives/désactives** des objets à l’arrivée dans une scène

💡 Exemple : si GameManager.puzzle1Done = true, la porte est déjà ouverte dans la prochaine scène.

**⚔️ Jour 30 : Le Défi de la Quête Complète**

**🎯 Objectif : Créer une expérience VR avec progression sur plusieurs scènes**

**Ta mission :**

1. Scene 1 : Intro avec bouton "Commencer"
2. Scene 2 : Environnement avec un puzzle (prendre une clef)
3. Scene 3 : Clef détectée → accès à une zone secrète
4. Optionnel : sauvegarder l’état (position, objets pris)
5. Bonus : son, lumière, ambiance, UI VR...

📷 **Boss Fight** : Montre-moi le résultat, je peux te conseiller comment le peaufiner en jeu réel.

**✅ Fin de la Semaine 6 : Tu sais maintenant…**

* Gérer plusieurs scènes
* Créer de la progression VR
* Commencer à **designer un vrai jeu** en étapes
* Penser comme un **game designer immersif**