TP1 : Encapsulation en C# pour Unity

Objectif du TP

Comprendre et appliquer le principe d'encapsulation dans le contexte du développement de jeux vidéo avec Unity.

Contexte

Vous travaillez sur un jeu d'action-aventure et vous avez découvert un script de gestion de personnage mal conçu. Ce script expose toutes ses données en public, ce qui rend le code fragile et difficile à maintenir.

Problématiques identifiées

Dans le script `PlayerCharacterBroken.cs` fourni, plusieurs problèmes ont été introduits par l'absence d'encapsulation :

1. La santé du personnage peut devenir négative ou dépasser sa valeur maximale
2. La vitesse de déplacement peut être modifiée à des valeurs aberrantes (négatives ou extrêmement élevées)
3. L'or peut être modifié directement depuis n'importe quelle classe sans passer par des méthodes dédiées
4. Le mode invincibilité peut être activé sans aucune condition ou limite de temps

Consignes

1. Analysez le script `PlayerCharacterBroken.cs` et identifiez tous les problèmes liés à l'absence d'encapsulation.

2. Créez un nouveau script `PlayerCharacter.cs` en appliquant le principe d'encapsulation :

    - Rendez privées toutes les variables d'instance qui ne doivent pas être modifiées directement
    - Utilisez l'attribut `[SerializeField]` pour les variables qui doivent rester visibles dans l'inspecteur Unity
    - Implémentez des propriétés avec des getters et setters pour contrôler l'accès aux données
    - Ajoutez des validations pour empêcher les états invalides (santé négative, etc.)
    - Créez des méthodes publiques pour les opérations spécifiques (prendre des dégâts, gagner/dépenser de l'or, etc.)

3. Assurez-vous que votre implémentation corrige tous les problèmes identifiés et rend le code plus robuste.

Code à améliorer

Le fichier `PlayerCharacterBroken.cs` est fourni dans le répertoire du TP. Analysez-le et créez votre nouvelle version correctement encapsulée.

Critères d'évaluation

- Application correcte du principe d'encapsulation
- Qualité des validations de données
- Cohérence des méthodes publiques proposées
- Facilité de maintenance et de compréhension du code produit

Rendu attendu

- Un fichier `PlayerCharacter.cs` contenant votre implémentation avec encapsulation
- Un court document expliquant vos choix d'implémentation et comment ils résolvent les problèmes identifiés

Bonus

- Implémentez un système de régénération de santé qui utilise correctement l'encapsulation
- Ajoutez un système de points d'expérience et de niveau avec propriétés encapsulées
- Créez un système de statistiques qui dépendent du niveau (force, agilité, etc.)