

Projet : Ajouter une UI basique pour le score et la santé du personnage

Exercice : Ajouter une UI Basique pour le Score et la Santé du Personnage

Dans cet exercice, vous allez apprendre à créer une interface utilisateur (UI) simple dans Unity pour afficher le **score** et la **santé** du personnage. L'objectif est de concevoir des éléments UI (textes, barres de santé) et les lier à la logique du jeu.

Étapes de l'Exercice :

1. Créer les éléments UI dans Unity :

1. Ouvrir la scène Unity :

- Ouvrez votre projet Unity et assurez-vous que la scène de jeu est prête.

2. Ajouter un Canvas pour l'UI :

- Allez dans le menu **GameObject > UI > Canvas** pour ajouter un **Canvas**.
- Un Canvas est l'élément parent de tous les composants UI dans Unity. Il assure que tous les éléments sont rendus à l'écran.

3. Ajouter un Text pour afficher le score :

- Cliquez sur le Canvas dans la hiérarchie.
- Allez dans le menu **GameObject > UI > Text**.
- Un objet **Text** sera ajouté au Canvas. Renommez-le en **ScoreText**.
- Dans l'**Inspector**, modifiez la propriété **Text** pour afficher quelque chose comme "Score: 0". Ce texte affichera le score du joueur pendant le jeu.

4. Ajouter une barre de santé (Health Bar) :

- Sélectionnez le Canvas.

- Allez dans le menu **GameObject > UI > Image**.
- Renommez l'objet en **HealthBar** et ajustez sa taille et position.
- Ajoutez une **Image** (l'objet qui représentera la barre de santé). La barre doit ressembler à une image remplie d'une couleur (par exemple, rouge).
- Vous pouvez ajouter un **Slider** si vous souhaitez une barre de santé dynamique :
 - Allez dans **GameObject > UI > Slider**.
 - Renommez-le en **HealthSlider** et ajustez sa taille pour qu'il ressemble à une barre de santé.
 - Dans l'**Inspector**, assurez-vous que le **Value** du Slider est mis à 100 (représentant 100% de la santé).

2. Créer un script C# pour gérer le score et la santé :

1. Créer un script pour gérer l'UI :

- Dans l'**Asset Explorer**, faites un clic droit dans votre dossier **Scripts**, puis **Create > C# Script**.
- Nommez-le **GameManager**.
- Ouvrez le script **GameManager.cs** et ajoutez les variables nécessaires pour gérer le score et la santé. Voici un exemple de script :

```
using UnityEngine;
using UnityEngine.UI;

public class GameManager : MonoBehaviour
{
    public int score = 0; // Score du joueur
    public int maxHealth = 100; // Santé maximale
    private int currentHealth; // Santé actuelle
    public Text scoreText; // Référence au Text du score
    public Slider healthSlider; // Référence à la barre de santé

    void Start()
```

```

{
    currentHealth = maxHealth; // Initialiser la santé
    UpdateUI(); // Mettre à jour l'UI au démarrage
}

void UpdateUI()
{
    // Mettre à jour l'affichage du score
    scoreText.text = "Score: " + score;

    // Mettre à jour la barre de santé
    healthSlider.value = (float)currentHealth / maxHealth;
}

// Méthode pour ajouter des points au score
public void AddScore(int points)
{
    score += points;
    UpdateUI();
}

// Méthode pour réduire la santé du joueur
public void TakeDamage(int damage)
{
    currentHealth -= damage;
    if (currentHealth < 0) currentHealth = 0; // La santé ne peut pas être inférieure à 0
    UpdateUI();
}

// Méthode pour restaurer la santé
public void Heal(int healAmount)
{
    currentHealth += healAmount;
    if (currentHealth > maxHealth) currentHealth = maxHealth; // La santé ne peut pas dépasser la valeur maximale
    UpdateUI();
}

```

```
}  
}
```

3. Lier les éléments UI dans le script :

1. Attacher le script au Canvas :

- Sélectionnez le **Canvas** dans la hiérarchie.
- Glissez et déposez le script **GameManager** sur le **Canvas** ou dans l'**Inspector** sous le Canvas.

2. Lier les objets UI au script :

- Dans l'**Inspector**, vous verrez les champs publics du script **GameManager** : **scoreText** et **healthSlider**.
- Faites glisser l'objet **ScoreText** (le Text) dans le champ **Score Text**.
- Faites glisser l'objet **HealthSlider** (le Slider) dans le champ **Health Slider**.

4. Tester et ajuster :

1. Testez l'UI :

- Appuyez sur **Play** pour tester le jeu.
- Le score doit s'afficher et changer lorsque vous appelez la méthode **AddScore()** dans votre jeu.
- La barre de santé doit également s'ajuster en fonction des changements de santé lorsque vous appelez **TakeDamage()** ou **Heal()**.

2. Manipulation du score et de la santé :

- Pour tester, vous pouvez ajouter des appels dans **Update** ou un autre script pour simuler des événements de changement de score et de santé.

```
void Update()  
{  
    // Exemple de test pour ajouter des points au score ou  
    infliger des dégâts  
    if (Input.GetKeyDown(KeyCode.Space)) // Ajouter des po  
        ints lorsque la touche espace est pressée
```

```

    {
        AddScore(10);
    }

    if (Input.GetKeyDown(KeyCode.H)) // Infliger des dégâts
    lorsque la touche H est pressée
    {
        TakeDamage(20);
    }

    if (Input.GetKeyDown(KeyCode.G)) // Soigner lorsque la
    touche G est pressée
    {
        Heal(10);
    }
}

```

5. Personnalisation de l'UI :

- **Style et positionnement :** Vous pouvez utiliser les propriétés de l'**Inspector** pour ajuster la couleur, la taille de texte, la police, la position des éléments UI, etc.
- **Animations :** Pour rendre l'UI plus dynamique, vous pouvez animer la barre de santé avec un Tweening (comme expliqué dans les précédents cours) ou ajouter des effets visuels.

Conclusion :

Vous avez maintenant une interface utilisateur basique qui affiche le **score** et la **santé** du personnage. Vous pouvez l'enrichir davantage en ajoutant d'autres éléments UI ou en personnalisant l'apparence des éléments existants. Cette interface simple peut être utilisée dans des jeux de plateforme, d'aventure ou de puzzle pour afficher des informations essentielles au joueur.