

Création d'une interface simple pour le jeu (affichage du score, menus)

L'interface utilisateur (UI) est un élément crucial de l'expérience de jeu. Une interface simple et intuitive peut améliorer considérablement l'engagement du joueur. Ce cours se concentre sur la création d'une **interface simple** dans Unity, incluant l'affichage du **score** et la mise en place de **menus** de base.

1. Structure de l'UI dans Unity

Avant de commencer à créer les éléments d'interface, il est important de comprendre la structure de l'UI dans Unity. Unity utilise le composant **Canvas** pour rendre les éléments de l'UI visibles à l'écran.

- **Canvas** : Un **Canvas** est l'élément de base dans Unity qui contient tous les éléments UI comme les textes, les boutons et les images.
 - **UI Elements** : Les éléments comme les **Text**, **Button**, **Slider**, **Image**, etc., sont ajoutés au Canvas pour former l'interface.
-

2. Affichage du Score

a) Créer un Text pour le score

1. Création du Canvas :

- Cliquez sur **GameObject** > **UI** > **Canvas** pour créer un nouveau Canvas.
- Le Canvas contient tous les éléments d'UI que nous allons créer (score, boutons, etc.).

2. Ajout du Text :

- Dans le Canvas, faites un clic droit > **UI** > **Text** pour ajouter un nouvel élément de texte.

- Vous verrez apparaître un objet texte sur l'écran. Vous pouvez le déplacer et ajuster sa position.
- Dans l'inspecteur, ajustez le texte pour qu'il soit clair et lisible (par exemple : **"Score : 0"**).

3. Modification de l'apparence :

- Changez la police, la taille et la couleur du texte dans l'inspecteur pour le rendre plus visible (par exemple, une taille de police de 30 et une couleur blanche).
- Placez-le dans un endroit visible, comme le coin supérieur gauche de l'écran.

4. Mise à jour du score avec un script :

- Créez un script en C# pour mettre à jour le texte du score en fonction des actions du joueur.
- Exemple de script pour gérer le score :

```
using UnityEngine;
using UnityEngine.UI;

public class GameManager : MonoBehaviour
{
    public Text scoreText; // Référence au texte UI
    private int score = 0; // Score du joueur

    void Start()
    {
        UpdateScore();
    }

    public void IncreaseScore(int amount)
    {
        score += amount;
        UpdateScore();
    }

    void UpdateScore()
```

```
{
    scoreText.text = "Score: " + score.ToString();
}
}
```

- Dans ce script, le **Text scoreText** est lié à l'élément texte du score dans l'UI.
- La méthode **IncreaseScore()** permet d'ajouter au score et de le mettre à jour en temps réel.

5. Associer le Text au Script :

- Dans l'inspecteur, faites glisser l'élément texte de score dans le champ **scoreText** du script GameManager.

3. Création de Menus Simples

a) Menu Principal

1. Créer un menu principal :

- Faites un clic droit dans le Canvas et sélectionnez **UI > Panel** pour créer un panneau qui servira de fond pour votre menu.
- Changez sa couleur et ajustez la taille pour qu'il couvre toute la scène si nécessaire.

2. Ajouter des boutons de menu :

- Faites un clic droit sur le **Panel** que vous venez de créer > **UI > Button**.
- Vous pouvez créer plusieurs boutons pour les actions comme **"Jouer"**, **"Options"**, **"Quitter"**.
- Ajustez la taille et la position des boutons pour que le menu soit clair et accessible.
- Changez le texte des boutons en cliquant sur le bouton dans l'hierarchie et en modifiant le composant **Text** qui est enfant du bouton.

3. Création d'un script pour gérer le menu :

- Ajoutez un script C# pour gérer l'interaction avec le menu.
- Exemple de script pour un menu avec des boutons :

```

using UnityEngine;
using UnityEngine.SceneManagement;

public class MainMenu : MonoBehaviour
{
    public void StartGame()
    {
        // Charger la scène de jeu
        SceneManager.LoadScene("GameScene");
    }

    public void QuitGame()
    {
        // Quitter l'application
        Application.Quit();
    }
}

```

- Dans ce script, **StartGame()** charge la scène de jeu et **QuitGame()** ferme l'application.

4. Associer le script aux boutons :

- Dans l'inspecteur, pour chaque bouton, vous pouvez associer les méthodes `StartGame()` ou `QuitGame()` aux actions du bouton.
- Pour ce faire, sélectionnez le bouton, puis dans l'inspecteur sous **Button (Script)**, cliquez sur **+** dans la section **On Click()**. Glissez ensuite l'objet contenant le script **MainMenu** et sélectionnez la méthode correspondante.

4. Transition entre le Jeu et le Menu

Pour rendre le jeu interactif, vous aurez besoin de gérer la transition entre le menu principal et le jeu lui-même.

1. Créer une nouvelle scène pour le jeu :

- Allez dans **File** > **New Scene** pour créer une nouvelle scène.
- Sauvegardez cette scène sous le nom **"GameScene"**.

2. Ajouter une gestion de transition :

- Lorsque le joueur clique sur "**Jouer**", le jeu doit charger la scène du jeu.
 - Assurez-vous que le nom de la scène dans le script correspond à celui de votre scène réelle.
-

5. Conclusion

L'UI dans Unity peut sembler complexe au début, mais une fois que vous comprenez les bases de la création de **textes**, de **boutons**, et de **menus**, vous pouvez facilement ajouter des éléments d'interface dans vos jeux. L'affichage du **score** et la gestion des **menus** sont essentiels pour l'expérience du joueur, et leur mise en œuvre dans Unity est relativement simple avec l'utilisation des composants **UI** comme **Canvas**, **Text**, et **Button**.