

Éléments d'interfaces – UI

Champ pour saisir le texte de l'utilisateur : UI.InputField

- À partir de *Hierarchy* faire *Create /UI/InputField*
- Modifiez les propriétés du champ (position, taille, ancrage, etc.) comme vous l'avez déjà fait pour les boutons.
- Dans l'inspecteur, l'évènement *OnEndEdit()* détermine l'action à prendre lorsqu'on termine la saisie avec la touche « entrée » ou lorsque le focus est retiré du champ (par exemple si on clic ailleurs dans l'interface).



Plus d'information ici : <https://docs.unity3d.com/ScriptReference/UI.InputField.html>

Utilisation de champ de saisie dans un script

- Il faut déclarer une variable de type `InputField`. (**using UnityEngine.UI** doit être présent au début du script)
- La propriété `.text` de l'objet UI permet d'accéder au texte saisi dans le champ.

Exemple

```
//...
using UnityEngine.UI;

public class Exemple : MonoBehaviour {

    public InputField objetInputField; //variable qui doit être assignée par l'objet de InputField dans l'inspecteur
    Text ageUI; //affiche la valeur dans l'interface UI
    int age; //variable de type int pour mémorise une valeur
    void Update ()
    {
        string texteSaisi = objetInputField.text; // Le contenu du texte saisi ira dans la variable texteSaisi.
        bool reussit = int.TryParse( objetInputField.text, out age); // Le texte saisi ira dans une variable de type int.
        if( reussit == true) //valide si la valeur est numérique
        {
            ageUI.text = age.ToString(); // convertit en string et affiche le texte saisi dans le texte UI
        }
        else
        {
            print(" valeur entrée invalide");
        }
    }
}
```

Conversion d'une chaîne en valeur numérique :

bool reussit = int.TryParse(string varChaine , out variableEntiere)

Convertit une chaîne à une valeur entière et retourne la valeur vraie. Si la chaîne n'est pas convertible à une valeur entière alors la valeur *false* est retournée. Et la variableEntiere est égale à 0.

Il existe aussi **float.TryParse(string varChaine , out variablFloat)**

Création et gestion des tableaux C#

Déclaration d'un tableau : Les crochets ajoutés à la fin d'un type indiquent qu'il s'agit d'un tableau

Exemple

```
int[] TableauValeurs;
string[] TabNoms;
public GameObject[] TabInventaires;
```

Allocation d'un tableau

Les tableaux en C# doivent avoir une taille fixe (pas de push ou pop, etc.). Par contre, il est possible de modifier la taille lors de l'exécution (runTime).

Initialisation :

Un tableau **privé** peut être initialisé lors de la déclaration ou dans une fonction (la fonction Start() par exemple)

Un tableau **public** peut être initialisé lors de la déclaration, dans une fonction ou dans l'**inspecteur**.

Exemple

```
//...
public class Exemple : MonoBehaviour {

int[] tableauScores = new int[10]; //un tableau de 10 éléments , les valeurs par défaut sont 0;
int[] tableauValeurs = new int[8] { 10,15,20,25,30,35,40,45 }; //un tableau de 8 éléments ayant des valeurs fixes
public string[] tableNoms; //un tableau donc la taille et les valeurs qui sera initialisé dans l'inspecteur;
public GameObject[] tabInventaires ;
void Start () {
    tableauScores = new int [10]; //redéfinition de la taille du tableau, MAIS efface le contenu
    tabInventaires = new GameObject[10]; //initialisé avec des valeurs nulles
    tableauValeurs = tableauScores ; //affecter un tableau à un autre

    tableauValeurs.CopyTo( tableauScores , 0 );
    //copie les 8 valeurs du tableau dans l'autre tableau en commençant par l'élément 0, le tableau scores garde sa dimension
}
}
```

Accès aux éléments

Exemples

```
public int[] tableauValeurs = new int[10];
void Start()
{
    tableauValeurs[0] = 10;
    tableauValeurs[1] = tableauValeurs[0] + 20;
    //imprime chaque valeur du tableau
    for(int i=0 ; i< tableauValeurs.Length ; i++)    // i : indice dans le tableau
    {
        print( tableauValeurs[i].ToString() );
    }
}
```

```
public string[] tableauNoms = new string[10]; // valeurs définies dans l'inspecteur
public Text nomsUI;

//fonction qui cumule le contenu du tableau dans une variable de type string et affiche la
//chaîne finale dans un champ de text

void AfficheLesNoms()
{
    string chaineNoms = "";
    for( int i=0 ; i<10 ; i++)
    {
        chaineNoms += tableauNoms[i] + " \n" ;
    }
    nomsUI.text = chaineNoms; //affiche la chaîne contenant les noms dans l'interface UI
}
```

Trier des tableaux

```
//trier le tableau des noms dans l'ordre croissant
System.Array.Sort(tableauNoms);

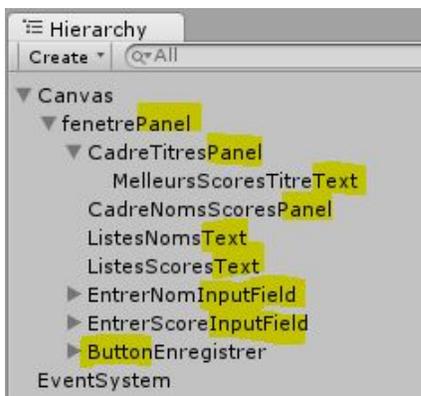
//trier deux tableaux selon l'ordre croissant des scores et faire suivre les noms
correspondants dans le tableau des noms
System.Array.Sort(tableauScores,tableauNoms);
System.Array.Reverse(tableauScores) ;    //inverse l'ordre des éléments
```

Consignes pour l'exercice

Au lancement, une fenêtre s'affiche. Cette fenêtre doit ressembler le plus possible au démo.

Modifiez les images, les couleurs, la police de caractère, les points de pivots des textes, etc. Les images et la police des éléments UI se trouvent dans l'onglet *Project*.

Le *Canvas* doit ressembler à l'image suivante. Notez les surlignés pour connaître le type de UI que vous devez utiliser. UI/Panel est un type d'image pour créer des cadres.



Remarque : N'oubliez pas de permettre le redimensionnement du canvas dans l'inspecteur (Canvas Scaler : Scalewith Screen Size).

Créez un script pour l'élément *Canvas*. Ce script contient :

- deux variables tableaux de 10 éléments, un pour les noms (string) et l'autre pour les scores (int)
 - les 4 variables nécessaires pour les textes UI : la liste des noms, la liste des scores et les champs de saisi UI pour le nom et pour le score.
- la fonction publique «EnregistrerNomScore()» qui :
 - vérifie si le score à enregistrer est plus grand que le score le plus petit du tableau (si le tableau est trié et inversé alors la valeur la plus petite se trouve à la dernière case). Et si la chaîne du nom saisi n'est pas vide ("") alors :
 - ajoute le nom du joueur à la dernière case du tableau des noms
 - ajoute le score du joueur à la dernière case du tableau des scores
 - trie les tableaux selon le score
 - cumule la liste des noms et des scores, (il faut créer une boucle et cumuler le contenu de chaque tableau dans des variables de type *string*. Il faut ajouter un "**\n**" pour changer de ligne entre les éléments). Les 10 meilleurs scores sont affichés dans l'ordre décroissant. **Les cases non remplies ne doivent pas s'afficher.**
 - affiche la chaîne des noms dans le UI texte ListeNomsText.
 - affiche la chaîne des scores dans le UI texte ListeScoresText.

- Ajoutez un événement `OnClick()` sur le bouton UI « BoutonEnregistrer »
- Lorsque le joueur écrit son nom et appuie sur le bouton « Enregistrer », la fonction « `EnregistrerNomScore()` » est appelée.