

## Enregistrement des données sur disque

### Enregistrement de données simples : PlayerPrefs

PlayerPrefs permet d'enregistrer les préférences des joueurs sur disque entre les sessions de jeux.

#### Éditeur / Standalone

Sur **Mac OS X** PlayerPrefs sont stockés dans `~/Library/Preferences`, dans un fichier `unity.[company name].[product name].plist`, où les noms de société et de produit sont les noms mis en place dans les paramètres du projet (Project Settings). Le même fichier `.plist` est utilisé pour les projets exécutés dans l'éditeur et de façon autonome.

Sur **Windows**, PlayerPrefs sont stockés dans le Registre sous `HKCU\Software\[company name]\[product name] key`, où les noms de société et de produit sont les noms mis en place dans les paramètres du projet.

Sur **WebGL**, PlayerPrefs sont stockées dans l'API IndexedDB du navigateur. Il y a un fichier de préférences par URL de WebGL et la taille du fichier est limitée à 1 mégaoctet. Si cette limite est dépassée, `SetInt`, `SetFloat` et `SetString` ne vont pas stocker la valeur et vont provoquer un `PlayerPrefsException`.

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

#### Fonctions de classe

- **SetInt** Permet d'enregistrer une préférence de type int (nombre entier) en lui associant une clé.
- **GetInt** Permet de récupérer une valeur de type int (nombre entier) à l'aide de la clé fournie.
- **SetFloat** Permet d'enregistrer une préférence de type float (nombre décimal) en lui associant une clé.
- **GetFloat** Permet de récupérer une valeur de type float (nombre décimal) à l'aide de la clé fournie.
- **SetString** Permet d'enregistrer une préférence de type string (chaîne de texte) en lui associant une clé.
- **GetString** Permet de récupérer une valeur de type string (chaîne de texte) à l'aide de la clé fournie.
- **HasKey** Renvoie true si la clé existe dans les préférences.
- **DeleteKey** Supprime la clé et sa valeur correspondante dans les préférences.
- **DeleteAll** Supprime toutes les clés et les valeurs des préférences. À utiliser avec prudence.
- **Save** Enregistre toutes les préférences modifiées sur le disque.

#### Exemple

```
//retourne vrai si le fichier (ou la clé) existe
bool infoExiste = PlayerPrefs.HasKey("FichierNoms");

PlayerPrefs.SetInt("Pointage", 10); //enregistre une valeur et crée la clé
int score = (PlayerPrefs.GetInt("Pointage")); //récupère la valeur

PlayerPrefs.DeleteKey("FichierNoms"); //efface le fichier (la clé) du disque
```

## Enregistrement des données complexes : PlayerPrefsX

Cette librairie de fonctions permet d'enregistrer des **tableaux** de données (Vectore3D, booléen, etc.) en utilisant la classe PlayerPrefs de Unity. Pour utiliser **PlayerPrefsX**, il faut importer le fichier de script de **PlayerPrefsX** dans le projet. Toutes les fonctions sont de type **static**, ce qui permet de les appeler dans d'autres scripts en utilisant la commande : **PlayerPrefsX.NomFonction()** .

### Exemple

```
string[] tableauNoms; // tableau des noms
int[] tableauScores; // tableau des scores

void Sauvegarde ()
{
    //Écriture des données
    PlayerPrefsX.SetStringArray("FichierNoms", tableauNoms); // écriture du tableau des noms
    PlayerPrefsX.SetIntArray("FichierScores", tableauScores); // écriture du tableau des scores

    //Lecture des données
    tableauNoms = PlayerPrefsX.GetStringArray("FichierNoms"); // lecture du fichier des noms
    tableauScores = PlayerPrefsX.GetIntArray("FichierScores"); // lecture du fichier des scores
}
```

Pour voir d'autres exemples d'utilisation de PlayerPrefsX consultez : <http://wiki.unity3d.com/index.php/ArrayPrefs2>

## Enregistrement des données sur un serveur : classe WWW

La Classe **WWW** de Unity permet l'accès simple aux données de pages Web. Il s'agit d'un petit module permettant de récupérer le contenu des URL.

### Fonctionnement :

- Vous lancez un téléchargement en arrière-plan en appelant WWW (url) qui renvoie un nouvel objet WWW.
- Vous pouvez inspecter la propriété **isDone** pour voir si le téléchargement est complété ou utiliser un **Yield** qui permettra d'attendre automatiquement que le téléchargement soit complété.

Il existe également des fonctionnalités pour créer des textures à partir d'images téléchargées sur le Web et pour diffuser et charger de nouveaux fichiers.

À utiliser si vous souhaitez obtenir des données d'un serveur Web pour l'intégration dans un jeu tel que des listes de pointage, de nouveaux médias à intégrer à votre jeu, etc.

La classe **WWW** peut être utilisée pour envoyer des requêtes **GET** et **POST** au serveur. GET sera utilisé par défaut et POST si vous fournissez un paramètre *postData*.

Voir aussi: **WWWForm** → permet de créer des données de formulaire valides pour le paramètre *postData*.

**Remarque:** les URL passées à la classe WWW doivent être '%' échappées -->remplacer les espaces par des %

## Exemple

```
// Obtenir un texte et une image à partir d'une URL et modifier la texture d'un objet
public class wwvExemple : MonoBehaviour
{
    public string url = " http://i.imgur.com/w10mfz.jpg";
    void Start()
    {
        StartCoroutine (Telecharge ());
    }

    IEnumerator Telecharge ()
    {
        //obtient le contenu d'un fichier texte sur un serveur
        WWW www1 = new WWW("http://jeu.tim.cmaisonneuve.qc.ca/test/infoArecuperer.html");
        yield return www1;
        print (www1.text);

        //télécharge une image
        WWW www2 = new WWW(url);
        yield return www2;
        Renderer leRenderer = GetComponent<Renderer>();
        leRenderer.material.mainTexture = www2.texture;
    }
}
```

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

## Consignes pour l'exercice

- Importez le script PlayerPrefX qui se trouve sur le site duc ours. Ce script doit rester dans l'onglet projet - il n'est pas utilisé dans la *hierarchy* - puisqu'il contient des fonctions statiques.
- Modifiez le script de gestion des meilleurs pointages réalisé au dernier cours. Il faut maintenant enregistrer les informations et les récupérer.
- S'il y a des résultats déjà enregistrés (**PlayerPrefs.HasKey**) il faut les récupérer au démarrage du jeu et les afficher.
  - Pour permettre l'affichage du contenu des tableaux à partir de différents endroits dans votre code, placez la boucle qui permet cet affichage dans une fonction (ex: AfficherTableau()).
- Lorsqu'on enregistre un nom (et un score) dans les tableaux, il faut les enregistrer sur le disque également.
- Lorsqu'on appui sur la touche « échappe » (*KeyCode.Escape*) les fichiers sur le disque sont effacés (*PlayerPrefs...*), les tableaux sont vidés et l'affichage à l'écran est mis à jour.