

J05 – Outils – Outils Graphique

Property Drawer

Property Drawer

Permet de contrôler la façon dont l'interface graphique d'une classe sérialisable va s'affiche dans la fenêtre Inspecteur.

Plus besoin d'écrire un inspecteur personnalisé complet.

S'occupe de redéfinir l'interface pour les propriétés

Existe des Property Drawer intégrés

Existe des Property Drawer Personnalisé

Existe des Decorate Drawer intégré

Existe des Decorate Drawer Personnalisé (Pa traité)

Exite des PropertyAttribute

Build-in Drawers

- **Range**

[Range (0, 1)]

```
public float floatRange = 0.5f;
```

[Range (0, 100)]

```
public int intRange = 50;
```

- **Multiline**

[Multiline (2)]

```
public string stringMultiline = "This text is using  
a multiline property drawer";
```

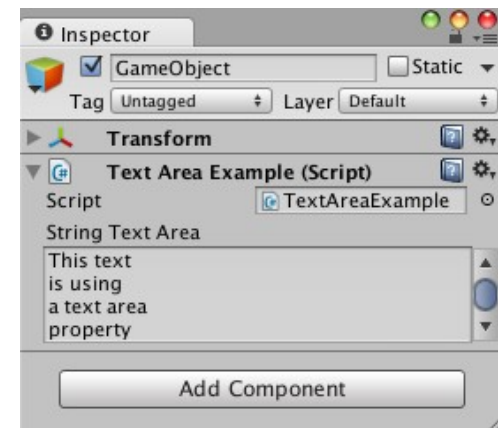
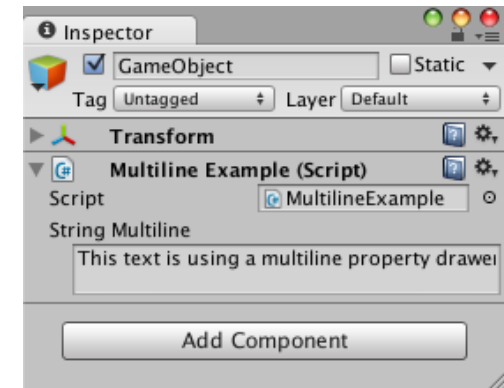
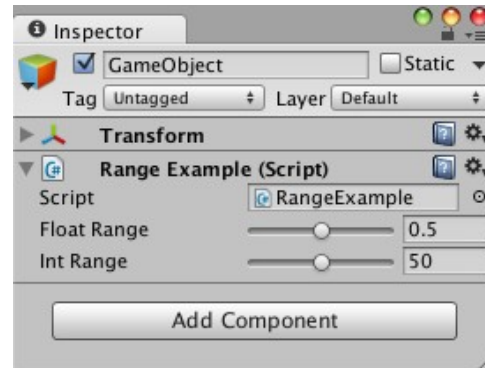
- **TextArea**

[TextArea (2, 4)]

```
public string stringTextArea = "This text \n is using \n  
a text area \n property \ndrawer";
```

- **ContextMenu , ContextMenuItem**

Cf cours précédents



Built-in Decorator Drawers

- **Header**

[Header("This is a group of variables")]

public int varA = 10;

public int varB = 20;

- **Space**

public int varC = 10;

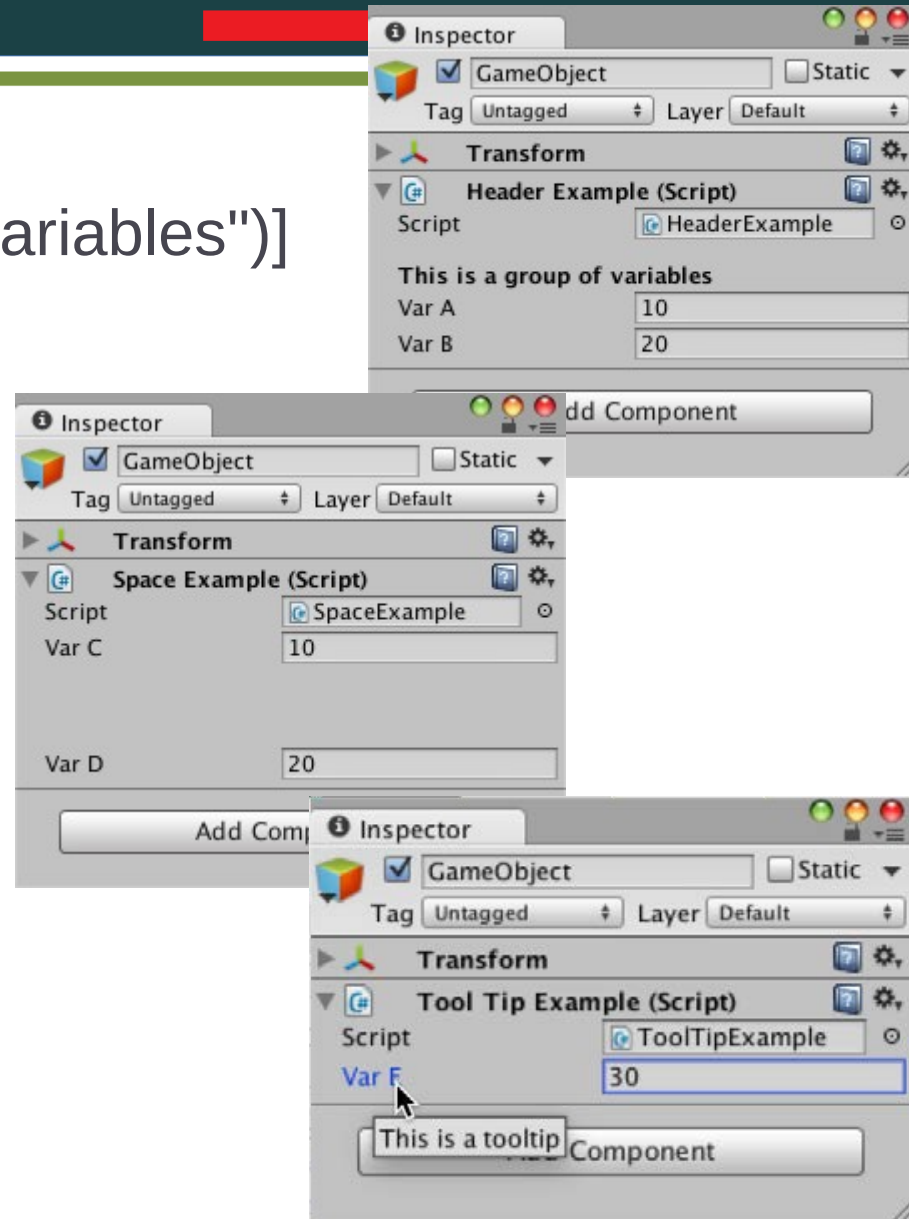
[Space(40)]

public int varD = 20;

- **TextArea**

[Tooltip("This is a tooltip")]

public int varE = 30;



SerializedObject / SerializedProperty

- Dans Custom inspector autre méthode pour mettre la main sur élément ciblé
- Existe la variable serializedObject
- Un SerializedObject représentant le ou les objets inspectés.
- Préférable d'utiliser SerializedObject quand travail avec des prefabs ou quand on travail sur l'édition multiple. procédé intégré a Unity
- Ne pas utiliser dans onSceneGui

<https://docs.unity3d.com/Manual/script-Serialization.html>

SerializedObject / SerializedProperty

Méthodes :

`serializedObject.FindProperty ("attribut")`

`property.FindPropertyRelative("champ classe")`

Pour mise a jour :

Dans `OnInspectorGui` :

- `serializedObject.Update ()`; au debut
- `serializedObject.ApplyModifiedProperties()`; à la fin

Mettre la main sur les valeurs

`property.intValue/floatvalue...`

Méthodes/propriétés pour tableau serialisé :

- `SerializedProperty.arraySize`
- `SerializedProperty.arrayElementType`
- `SerializedProperty.GetArrayElementAtIndex`
- `SerializedProperty.InsertArrayElementAtIndex`
- `SerializedProperty.DeleteArrayElementAtIndex`

Custom Property Drawer

Conditions :

- Doit avoir une classe out struct serializable pas dans Editor
- Doit avoir une classe qui hérite de PropertyDrawer dans Editor
- Doit avoir l'attribut [CustomPropertyDrawer (typeof(ClasseCible))]

Méthodes :

- GetPropertyHeight : donner la hauteur du champs dans l'inspecteur
- OnGui comme OnInspectorGUI :

Attention : que EditorGUI et GUI autorisé, pas de layout automatique ici.
EditorGUILayout pas supporté.

Exemples :

- <https://blogs.unity3d.com/2012/09/07/property-drawers-in-unity-4/>
- <https://unitylist.com/p/na1/Unity-Sprite-Property-Drawer>

PropertyAttribute

Permet de faire des attributs personnalisés

Classe qui hérite de PropertyAttribute.

Si on veut un attribut nommé Arme on devra appeler cette class ArmeAttribute

On aura alors un Attribut [Arme]

Puis CustomPropertyDrawer du type ArmeAttribute