

# Dialogue system – Documentation technique

Le 04-06-2021 par Alexandre Venet <https://twitter.com/alhomepage>

## 1. Introduction

---

Ce fichier présente **objets** utilisés pour le système de dialogue de *Reliquia: The Immortality Stone*, ainsi que les éléments **exposés** (publics, privés sérialisés), ou utiles à la compréhension, des **scripts** utiles pour une intervention par programmation.

## 2. Scripts

---

Les scripts sont distribués dans des **dossiers** du projet à des fins de clarté :

- **Game** : ce qui relève des objets de la scène, les composants,
- **Interfaces** : les interfaces C# à implémenter,
- **ScriptableObjects** : les classes définissant les fichiers instances SO,
- **Types** : autres types utiles de structure de données.

Si certains types ne peuvent pas apparaître en *Inspector*, cela est mentionné au cas par cas.

Tous les scripts, hors *packages*, appartiennent à l'**espace de nom** « **AlexandreDialogues** ».

## 3. Editor Scripting

---

Les fichiers de mise en forme pour l'éditeur (*Editor Scripting*) se trouvent et doivent rester dans le dossier **Assets/Editor**. Ils concernent :

1. **DialogueManager**,
2. **DialogueRef**,
3. **Dialogue**,
4. **HelpFile**.

Le **menu** d'*Unity* a une entrée : **Tools/Reliquia**. Les sous-entrées sélectionnent les fichiers d'aide :

- **HelpFile** (fichier intégré à l'interface),
- **notice d'utilisation** (pdf),
- **documentation technique** (pdf).

## 4. Prefabs

---

Deux *prefabs* sont fournis :

1. **CharacterUIObject** : ensemble d'éléments d'UI pour un personnage,

2. **DialogueManager** : l'objet comprenant le gestionnaire de dialogue et tous les éléments d'UI relatifs au dialogue.

## 5. Game

### 5.1. Dialogue

#### 5.1.1. DialogueManager

Classe
Accès : public
Héritage : MonoBehaviour
Singleton
Mis en forme par Editor Scripting : - <b>EDialogueManager</b> .

Propriété : <b>Instance</b>	
Accès : public	Type : DialogueManager
Modificateur : static	Accesseur (get), pas de mutateur (set)
Propriété permettant d'accéder aux autres membres de la classe, en tant que la classe est définie comme Singleton.	

Champ : <b>_inventory</b>	
Accès : private	Type : Inventory
Référence au fichier ScriptableObject d'inventaire.	

Champ : <b>_defaultCamera</b>	
Accès : private	Type : GameObject
Référence à une caméra de dialogue par défaut, caméra virtuelle <i>Cinemachine</i> .	

Champ : <b>_canvas</b>	
Accès : private	Type : GameObject
Référence au Canvas d'UI enfant.	

Champ : <b>_textLastCharIdentifier</b>	
Accès : private	Type : char
Caractère unique identifiant la fin du texte de réplique, styles inclus.	

Classe : <b>UIObjects</b>	
Accès : private	
Classe interne à <b>DialogueManager</b> pour définir l'ensemble des références suivantes.	
Champ : <b>m_character</b>	
Accès : public	Type : GameObject
Un objet d'UI de personnage.	
Champ : <b>m_image</b>	
Accès : public	Type : Image (UnityEngine.UI)
Le composant image de l'objet précédent et qui reçoit un <i>sprite</i> (voir <b>Character</b> ).	
Champ : <b>m_name</b>	
Accès : public	Type : TextMeshProUGUI
Composant accueillant le texte du nom du personnage.	

Champ : <b>_UICharacterLeft</b>	
Accès : private	Type : UIObjects
Voir <b>UIObjects</b> .	

Champ : <b>_UICharacterRight</b>	
Accès : private	Type : UIObjects
Voir <b>UIObjects</b> .	

Champ : <b>_replyText</b>	
Accès : private	Type : TextMeshProUGUI
Le composant accueillant le texte de réplique.	

Champ : <b>m_controlledScripts</b>	
Accès : public	Type : List<MonoBehaviour>
Collection de scripts destinés à implémenter l'interface <b>IEnableForDialogue</b> . Son contenu est testé dans Awake() pour initialiser <b>m_IEnableForDialogue</b> .	

Champ : <b>m_EnabledForDialogue</b>	
Accès : public	Type : List<IEnabledForDialogue>
Collection de scripts implémentant l'interface <b>IEnabledForDialogue</b> . Elle est initialisée dans Awake() avec les contenu de <b>m_controlledScripts</b> qui implémentent effectivement l'interface. Cette collection est utilisée par <b>DialogueEntry</b> pour déclencher ou non <b>EnableMe(bool value)</b> .	

Champ : <b>m_conditionsBools</b>	
Accès : public	Type : List<bool>
Collection de booléens issus des méthodes personnelles utilisées comme condition de démarrage de dialogue dans une entrée de dialogue (type <b>StartConditionsTestType.Method</b> ).	

Méthode : <b>DialogueCheckStart(DialogueRef dialogueRef)</b>	
Type retourné : void	
Méthode de vérification d'entrée de dialogue. Exemple d'appel : <i>DialogueManager.Instance.DialogueCheckStart(_dialogueRef);</i>	

### 5.1.2.DialogueRef

Classe
Accès : public
Héritage : MonoBehaviour
Mise en forme par Editor Scripting : <ul style="list-style-type: none"> <li>- <b>EDialogueRef</b> : paramètres généraux et ouverture de fenêtre,</li> <li>- <b>EDialogueRefWindow</b> : édition des entrées de dialogue,</li> <li>- <b>EBaseWindow</b> : classe de base de <b>EDialogueRefWindow</b>.</li> </ul>

Champ : <b>m_dialogueRefType</b>	
Accès : public	Type : DialogueRefType
Le type d'interaction choisi (voir <b>DialogueRefType</b> ). Exemple d'utilisation : <i>if (_dialogueRef.m_dialogueRefType == DialogueRefType.InteractiveZone){//...}</i>	

Champ : <b>m_levelProgress</b>	
Accès : public	Type : LevelProgress
Référence au fichier ScriptableObject de progression. Voir <b>LevelProgress</b> .	

Champ : <b>m_virtualCamera</b>	
Accès : public	Type : GameObject
Référence à la caméra virtuelle pour cette entrée de dialogue. Peut rester null si aucune caméra n'est prévue.	

Champ : <b>m_myDialogs</b>	
Accès : public	Type : DialogueEntry[]
Collection d'entrées de dialogue. Voir <b>DialogueEntry</b> .	

## 6. Interfaces

### 6.1. IEnableForDialogue

Interface
Accès : public
Implémenter cette interface pour préparer un script de façon à l'inclure dans la collection <b>DialogueManager.m_controlesScripts</b> (et par suite <b>DialogueManager.m_IEnableForDialogue</b> ) afin que sa méthode <b>EnableMe(bool value)</b> soit déclenchable au début et à la fin d'un dialogue.

Méthode : <b>EnableMe(bool value)</b>
Type retourné : void
Méthode appelée par <b>DialogueManager</b> avec : <ul style="list-style-type: none"> <li>- <i>false</i> si <b>DialogueEntry.m_useControlledScriptsAtStart</b> <i>true</i>,</li> <li>- <i>true</i> si <b>DialogueEntry.m_useControlledScriptsAtEnd</b> <i>true</i>.</li> </ul> Exemple d'utilisation : <pre>public void EnableMe(bool value){ _isActive = value; }</pre>

## 7. ScriptableObjects

### 7.1. Character

Classe
Accès : public
Héritage : ScriptableObject
Entrée dans le menu de création d'asset : ScriptableObjects/Dialogue/Character.

Champ : <b>m_spriteLeft</b>	
Accès : public	Type : Sprite
Référence au <i>sprite</i> du personnage qui s’affiche à gauche de l’écran.	

Champ : <b>m_spriteRight</b>	
Accès : public	Type : Sprite
Référence au <i>sprite</i> du personnage qui s’affiche à droite de l’écran.	

## 7.2. Dialogue

Classe
Accès : public
Héritage : ScriptableObject
Entrée dans le menu de création d’ <i>asset</i> : ScriptableObjects/Dialogue/Dialogue.
Mise en forme par Editor Scripting : <ul style="list-style-type: none"> <li>- <b>EDialogue</b> : ouverture de fenêtre,</li> <li>- <b>EdialogueWindow</b> : édition du dialogue,</li> <li>- <b>EBaseWindow</b> : classe de base de <b>EdialogueWindow</b> .</li> </ul>

Champ : <b>m_test</b>	
Accès : public	Type : string
Si ce champ est renseigné, alors le texte est affiché en console lors du déclenchement du dialogue.	

Champ : <b>m_casting</b>	
Accès : public	Type : Role[]
Collection décrivant la distribution des rôles pour ce dialogue.	

Champ : <b>m_replies</b>	
Accès : public	Type : Reply[]
Collection de répliques pour ce dialogue.	

## 7.3. Inventory

Classe
Accès : public
Héritage : ScriptableObject
Entrée dans le menu de création d' <i>asset</i> : ScriptableObjects/Dialogue/Inventory

Champ : <b>m_inventoryStrings</b>	
Accès : public	Type : string
Collection de textes décrivant l'inventaire du personnage jouable.	

## 7.4. LevelProgress

Classe
Accès : public
Héritage : ScriptableObject
Entrée dans le menu de création d' <i>asset</i> : ScriptableObjects/Dialogue/LevelProgress

Champ : <b>m_levelStep</b>	
Accès : public	Type : int
Le niveau actuel de progression. Lors du chargement du fichier, cette variable prend la valeur de <b>m_initValue</b> .	

Champ : <b>m_initValue</b>	
Accès : public	Type : int
La valeur initiale, par défaut 0. Lors du chargement du fichier, cette valeur est affectée à <b>m_levelStep</b> .	

## 8. Types

### 8.1. Dialogue

#### 8.1.1. DialogueColorsRef

Classe
Accès : public

Champ : <b>ColorsHash</b>	
Accès : public	Type : Dictionary<string, int>
Non affichable en <i>Inspector</i> . Collection des champs <b>Name</b> et <b>HashCode</b> des styles de <b>TMPRO Reliquia StyleSheet</b>	

#### 8.1.2. Reply

Classe
Accès : public

Champ : <b>m_role</b>	
Accès : public	Type : Role
Rôle utilisé pour la réplique courante.	

Champ : <b>m_stageDirections</b>	
Accès : public	Type : string
Chaîne de caractères des didascalies de réplique.	

Champ : <b>m_styleColorHash</b>	
Accès : public	Type : int
Valeur du <b>HashCode</b> d'un style de <b>TMPRO Reliquia StyleSheet</b> .	

Champ : <b>m_charSpeed</b>	
Accès : public	Type : float
Temps d'affichage d'une lettre de dialogue. En éditeur, valeur limitée entre 0 et 1.	



Champ : <b>m_spaceSpeed</b>	
Accès : public	Type : float
Temps d’affichage d’un espace entre lettres de dialogue. En éditeur, valeur limitée entre 0 et 1.	

Champ : <b>m_text</b>	
Accès : public	Type : string
Le texte de la réplique.	

Champ : <b>m_replyEvent</b>	
Accès : public	Type : UnityEvent
L’événement de la réplique. N’est déclenché que s’il a des méthodes abonnées.	

### 8.1.3. Role

Classe
Accès : public

Champ : <b>m_character</b>	
Accès : public	Type : Character
Un personnage pour ce rôle.	

Champ : <b>m_characterView</b>	
Accès : public	Type : RoleCharacterView
Une valeur de point de vue. Voir <b>RoleCharacterView</b> .	

### 8.1.4. RoleCharacterView

Type : enum
Accès : public
Valeurs : <ul style="list-style-type: none"><li>- Left</li><li>- Right</li></ul>
Décrit la position d'un personnage à l'écran de dialogue : à gauche ou à droite. Décrit également l'angle de vue pour sélectionner le <i>sprite</i> à afficher (par exemple : image du personnage vu de ¾ droite à gauche, image du personnage vu de ¾ gauche à droite).

## 8.2. DialogueEntry

### 8.2.1. DialogueEntry

Classe	
Accès : public	
Champ : <b>m_name</b>	
Accès : public	Type : string
Nom du dialogue.	
Champ : <b>m_useControlledScriptsAtStart</b>	
Accès : public	Type : bool
Si <i>true</i> , alors au début du dialogue <b>DialogueManager</b> lance la méthode <b>EnableMe(false)</b> de l'interface <b>IEnableForDialogue</b> .	
Champ : <b>m_eventAtStart</b>	
Accès : public	Type : UnityEvent
Un événement au démarrage du dialogue, qui ne se déclenche que s'il a des méthodes abonnées.	
Champ : <b>m_dialogueSO</b>	
Accès : public	Type : Dialogue
Référence à un fichier de dialogue pour cette entrée de dialogue.	
Champ : <b>m_startconditions</b>	
Accès : public	Type : StartConditionsTest[]
Collection des conditions de démarrage.	
Champ : <b>m_dialogueAltSO</b>	
Accès : public	Type : Dialogue
Dialogue alternatif dans le cas des conditions de démarrage <b>Object</b> ou <b>Method</b> .	
Champ : <b>m_levelChangesAtEnd</b>	
Accès : public	Type : bool
Si <i>true</i> , alors à la fin du dialogue, le niveau de <b>LevelProgress</b> change.	

Champ : <b>m_nextLevel</b>	
Accès : public	Type : int
Si <b>m_levelChangesAtEnd</b> est à <i>true</i> , alors <b>LevelProgress</b> prend cette valeur.	

Champ : <b>m_useControlledScriptsAtEnd</b>	
Accès : public	Type : bool
Si <i>true</i> , alors à la fin du dialogue <b>DialogueManager</b> lance la méthode <b>EnableMe(true)</b> de l'interface <b>IEnableForDialogue</b> .	

Champ : <b>m_eventAtend</b>	
Accès : public	Type : UnityEvent
Un événement à la fin du dialogue, qui ne se déclenche que s'il a des méthodes abonnées.	

## 8.3. DialogueRef

### 8.3.1. DialogueRefType

Type : enum
Accès : public
Valeurs : <ul style="list-style-type: none"> <li>- InteractiveZone</li> <li>- TriggerZone</li> </ul>
Le type d'interaction souhaitée avec un objet présentant le composant <b>DialogueRef</b> : <ul style="list-style-type: none"> <li>- <b>InteractiveZone</b> : par entrée utilisateur,</li> <li>- <b>TriggerZone</b> : lancement automatique.</li> </ul>

## 8.4. StartConditions

### 8.4.1. StartConditionsOperator

Type : enum
Accès : public
Valeurs : <ul style="list-style-type: none"> <li>- EqualTo</li> <li>- NotEqualTo</li> <li>- LessThan</li> <li>- LessThanOrEqualTo</li> <li>- GreaterThan</li> <li>- GreaterThanOrEqualTo</li> </ul>

Opérateurs de comparaison utilisés pour établir une condition de démarrage de dialogue de type **Level**. Correspondance syntaxique C# :

- ==
- !=
- <
- <=
- >
- >=

#### 8.4.2. *StartConditionsTest*

Classe

Accès : public

Champ : **m\_type**

Accès : public

Type : StartConditionsTestType

Le type de condition de démarrage utilisée pour ce dialogue.

Champ : **m\_operator**

Accès : public

Type : StartConditionsOperator

L'opérateur de comparaison choisi pour un type de comparaison **Level**.

Champ : **m\_level**

Accès : public

Type : int

La valeur sur laquelle effectuer la comparaison.

Champ : **m\_object**

Accès : public

Type : string

Si la condition de démarrage est un **objet**, alors c'est le **nom** de cet objet.

Champ : **m\_eventCondition**

Accès : public

Type : UnityEvent

Si la condition de démarrage est de type **Method**, ceci est l'événement qui se déclenche s'il a des méthodes abonnées.

### 8.4.3. *StartConditionsTestType*

Type : enum
Accès : public
Valeurs : <ul style="list-style-type: none"><li>- Level</li><li>- Object</li><li>- Method</li></ul>
Le type de condition de démarrage d'un dialogue. <ul style="list-style-type: none"><li>- <b>Level</b> : par niveau d'un <b>LevelProgress</b>,</li><li>- <b>Object</b> : par présence d'un objet dans l'inventaire,</li><li>- <b>Method</b> : par script dans une méthode personnalisée.</li></ul>