Structure des fonctions Unity, manuel et scripting

# Organigramme des classes et fonctions essentielles unity

Il faut distinguer les classes et les fonctions. Pour simplifier (vous entrerez dans les détails plus tard), les classes contiennent des fonctions, qui sont ce que vous appelez pour réaliser une action. Translate est une fonction qui appartient à la classe Transform. C’est une manière de créer des catégories.

## Les essentielles

* **MonoBehaviour** : classe de base. Tout script en dérive. C'est ce qui permet d'utiliser Update, Start et autres fonctions.
* **GameObject** : base pour tout objet existant dans Unity. Permet d'accéder à toutes les autres principales. Gère l'activation, le tag, l'instance (cours prochain) et la destruction.
* **Transform** : position, rotation et échelle d'un objet. Toute transformation sur ces paramètres.

## Debug et affichage visuel

* **GUI** : Affichage sur l'écran (barre de vie, flèches, texte, etc.)
* **GUIText** : sorte de GUI, spécifiquement un texte.
* **GUITexture** : sorte de GUI, spécifiquement une texture.
* **Gizmos** : comme un GUI, mais uniquement visible en mode Editor. Très utile pour la visualisation de données sans interférer sur l'esthétique du jeu (les icônes de Light, Camera ou les cubes en wireframe sont des Gizmos).
* **Debug** : ensemble de commandes pour aider à débuguer son code (texte en console, géométries diverses, proche du Gizmos).

## Mise en scène

* **Camera** : tous paramètres d'une caméra (field of view, frustrum, couleur d'arrière plan, etc.)
* **Light** : tous paramètres d'une lumière (couleur, étendue, priorité, etc.)
* **AudioSource** : le son à jouer, quand le jouer, l'arrêter, modifier son pitch, etc.
* **Particle** : appliquer des effets particuliers sur une particule.
* **Input** : gérer les appuis de l'utilisateur.

## Outils et utilitaires

* **Mathf** : toutes les commandes mathématiques prévues par Unity et optimisée pour le temps-réel.
* **Quaternion** : tous paramètres propres aux Quaternions (les rotations).
* **Random** : l'aléatoire.
* Ray : lancé de rayon. Comme un laser, ça permet de détecter la si quelque chose le traverse et de récupérer des information sur ce qui est entré en collision.
* **Color** : Tout paramètre de couleur, modification de celle-ci.
* **Time** : Tout paramètre de temps (dont le célèbre et indétrônable "deltaTime").
* **Vector3** : tout paramètres et modification d'un vecteur. Utile pour appliquer des modifications subtiles, des variations.
* **Screen** : paramètres de l'écran (taille , orientation, etc.)
* **Texture** : paramètres d'une texture (pixel à tel endroit, modification, etc.)

## Physique

* **Collider** : tout ce qui concerne les colliders (ce qui capte la collision). Permet de faire appel à un Trigger (zone de contact) ou une Collision (surface de contact).
* **Collision** : tout paramètre de collision (indépendament de ce qui capte). A exploiter avec **ContactPoint** qui permet de récupérer la normal, l'autre objet ou la localisation du contact.
* **Rigidbody** : Contrôle de la position d'un objet à travers une simulation physique.
* **Physics** : ensemble de commandes utiles pour la détection de rigidbodies comme l'overlapsphere ou le raycast.

## Affichage

* **Material** : tout ce qui concerne les modifications d'un matériau.
* **Animation** : tout ce qui concerne les animations (Legacy), c'est-à-dire l'ancien système.
* **Animator** : tout ce qui concerne l'animation par automate à état (Mecanim). Plus évolué que l'animation.

# Fonctions du MonoBehaviour essentielles

## Avant Update

* **Awake** : quand le script apparaît dans la scène.
* **Start** : quand le script est exécuté pour la première fois (donc, une seule fois).
* **OnEnable** : quand le script est activé.
* **OnDisable** : quand il est désactivé.

## Boucles temps-réel

* **Update** : appelé à chaque frame.
* **FixedUpdate** : appelé à fréquence fixe. Dedans, on inclut nos calculs de physique.
* **LateUpdate** : appelé après que tous les calculs Update soient réalisés.

## Contextuelles

* **OnLevelWasLoaded** : dès que le niveau est chargé. Pratique pour les scripts qui passent d’une scène à une autre.
* **OnDestroy** : dès que le script (ou le gameObject) est détruit.
* **OnBecameVisible** : dès que le gameObject contenant un renderer apparaît à la camera. Noter que la camera de l’éditeur (Scene View) peut compter.
* **OnBecameInvisible**: dès que le gameObject contenant un renderer disparaît de la camera. Noter que la camera de l’éditeur (Scene View) peut compter.

## Inputs

* **OnMouseDown** : dès que le clic gauche (LMB) est appuyé, au-dessus de l’objet s’il a un collider.
* **OnMouseEnter** : dès que le curseur passe au-dessus du collider.
* **OnMouseExit** : dès qu’il en sort.
* **OnMouseOver** : tant qu’il reste au-dessus.
* **OnMouseUp** : dès que le clic gauche est relâche, au-dessus du collider.

## Affichage

* **OnGUI** : Gère l’affichage à l’écran de textes, textures et autres informations à l’écran. Voir la classe GUI : <http://docs.unity3d.com/ScriptReference/GUI.html>
* **OnDrawGizmos**: C’est dedans qu’on ajoute les fonctions tirées de Gizmos pour l’affichage visuel d’aide dans l’éditeur.