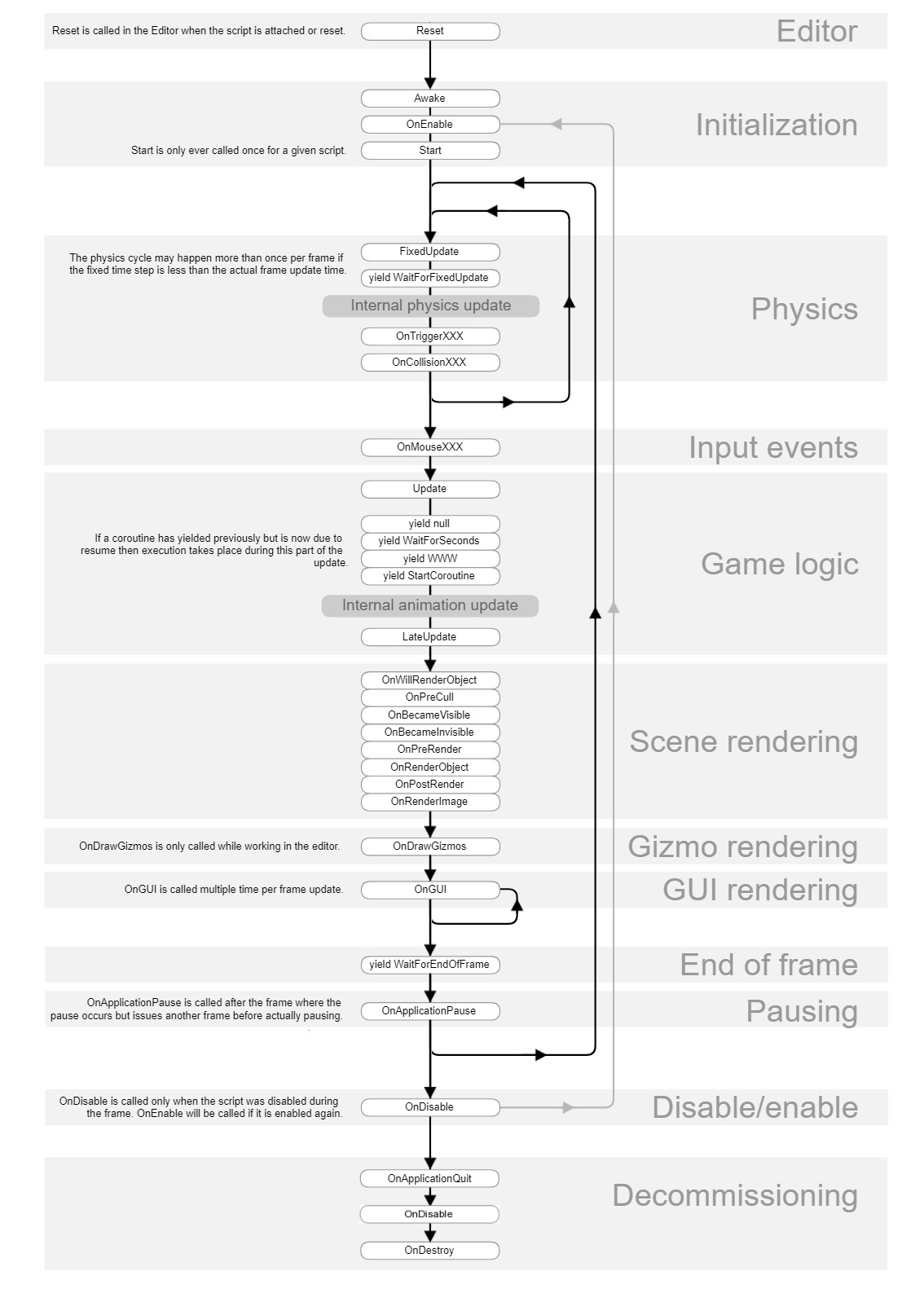
**Ciclo de vida del script en Unity**



A diferencia de otros programas, los scripts no son ejecutados en el orden predeterminado en comparación a un script ordinario. El orden a continuación es el que usa el programa (tomando en cuenta que es desde el inicio hasta el final el orden jerárquico):

**RESET:** Esta se usa para inicializar las propiedades de script cuando es la primera vez que se adjunta el script al programa y cuando el comando “Reset” es empleado.

**INITIALIZATION:** Es un conjunto de funciones que se usan inmediatamente comienza una escena.

***Awake:*** función que siempre se llama antes de cualquier función Start y justo después de que un prefab es instanciado. Se hace para activar el propio script por primera vez, después se recurre al start.

***OnEnable:*** función es llamada justo después de que el objeto esté activado. Se usa cuando es creada una instancia MonoBehaviour en el momento en que el objeto es cargado, es decir, cada vez que el objeto sea llamado para dar a entender que está en uso.

Estas funciones son llamadas antes que todos los scripts (Start, Update, etc.) siempre y cuando el objeto al que le pertenece el script no esté instanciado durante el gameplay.

**FIRST FRAME:** Esta, que es por lo regular referida como “Start”, es llamada antes de la primera actualización de frame solo si la instancia del script está activada. Útil para llamar variables, inicializar componentes y establecer referentes a otros objetos.

Estas funciones son llamadas antes que todos los scripts a continuación (Update, etc.) siempre y cuando el objeto al que le pertenece el script no esté instanciado durante el gameplay.

**BETWEEN FRAMES:** Estas funciones se ejecutan a lo largo que se cargan los frames del juego, actualizando los cambios que va teniendo el script. Puede ser desde iteraciones, animaciones, posicionamiento de cámara y jugador, etc.

***FixedUpdate:*** La primera en ejecución. Esta puede ser llamada varias veces en el mismo frame. Esta se encarga de todos los cálculos físicos y actualizaciones que ocurren inmediatamente después del propio FixedUpdate. Se usa por lo regular como un temporizador viable para el control de los frames.

***Update:*** Se usa una vez por frames y es la actualización de los estados del frame. Eso quiere decir que es la ideal para usar funciones que requieran información actualizada constante como la entrada de usuarios, presionar botones, animaciones, etc. Además, establece un orden de ejecución entre las funciones integradas.

***LateUpdate:*** Usada una vez por frame después de que Update haya sido ejecutado. Esta es usada para los cálculos que faltaron por finalizar de Update o por cálculos que dependan de las acciones de Update. Por ejemplo, el movimiento de la cámara dependiendo de cómo se mueva un personaje, lo que hace que el personaje se mueva antes de que la cámara reaccione al cambio.

***Rendering:*** Usada para controlar la apariencia visual de los objetos en escena. Cómo se ven, cómo se verán después de una determinada acción, aplicar efectos visuales en un determinado ambiente, etc. Esta parte de un proceso:

* ***OnPreCull:*** llamado antes de que la cámara corte la escena. Culling, que es la acción de cortar, determina qué objeto son visibles a la cámara. Es llamado justo antes de que el Culling tome lugar.
* ***OnBecameVisible/Invisible:*** llamado cuando un objeto se vuelve visible/invisible a cualquier cámara.
* ***OnPreRender:*** llamado antes de que se comience a renderizar la escena.
* ***OnRenderObject:*** usado después de que toda la renderización de la escena sea hecha.
* ***OnPostRender:*** llamado después de que se finalice a renderizar la escena.
* ***OnRenderImage:*** para permitir el post-procesamiento de la imagen después de que la escena esté completa.
* ***OnGUI:*** se utiliza para dibujar elementos de la interfaz de usuario (UI) en la pantalla durante la ejecución del juego.
* ***OnDrawGizmos:*** para dibujar Gizmos (representar información sobre el objeto en el Editor, como posiciones, direcciones, colisiones, rutas, etc)

***Coroutines:*** funciones que pueden suspender su ejecución durante un número especificado de fotogramas. Se utilizan principalmente para realizar tareas que necesitan o pueden ser realizadas de manera asincrónica. es una función que puede suspender su ejecución (yield) hasta que la YieldInstruction finalice.

* ***Yield:*** La coroutine continua después de que todas las funciones en Update hayan sido llamadas en el siguiente frame.
* ***Yield WaitForSeconds:*** continua después de un retrase de tiempo específico y que todas las funciones en Update hayan sido llamadas.
* ***Yield WaitForFixedUpdate:*** continua después de que todas las funciones de FixedUpdate hayan sido llamadas.
* ***Yield StartCoroutine:*** encadena la coroutine y va a esperar hasta que coroutine MyFunc haya sido completado primero.

**QUIT:** Son funciones que llaman a todos los objetos activos en la escena para finalizar su función o ir descomprimiendo la información de datos.

***OnDisable:*** función para dictar que un comportamiento se vuelve deshabilitado o inactivo.

***OnApplicationQuit:*** útil para proporcionar a los usuarios una manera de cerrar la aplicación de manera controlada desde dentro del juego.

**DESTROY:** Usado en caso de que el objeto desee ser destruido o sacarlo de escena. Con OnDestroy, cuando se efectúe la última actualización de frame para el objeto, este se eliminará.