

# **SCRIPTING EN UNITY**

## **MÉTODOS DE MONOBEHAVIOUR**

Ramón Mollá

rmolla at dsic.upv.es - ext. 73549

Grupo de Informática Gráfica

Departamento de Sistemas Informáticos y Computación

# Índice

## Scripts predefinidos en Unity

Awake(), Start()

Update(), FixedUpdate(), LateUpdate()

OnGUI(), OnDisable(), OnDestroy()

# Unity Scripts (I)

Los scripts en Unity3D heredan todos de la clase *MonoBehaviour*

Un script es un componente de GameObject

Un GameObject puede tener muchos scripts

Existen métodos implícitos y específicos de Unity3D que son ejecutados periódicamente

Estos métodos actualizan el estado del GameObject al cual está asignado, actualizan su estado físico (fuerzas), lo transforman (posición, escala y rotación), modifican su comportamiento, mandan mensajes, etc

# Unity Scripts (II)

Estos métodos se pueden sobrescribir en cada clase, si procede

**Awake(), Start()**

Arranque

**OnCollisionEnter(), OnCollisionStay(), OnCollisionExit()** y sus equivalentes en objetos no cinemáticos **OnTrigger**

Simulación física

**OnMouseDown(), OnMouseOver(), OnEnable(), OnDisable() ...**

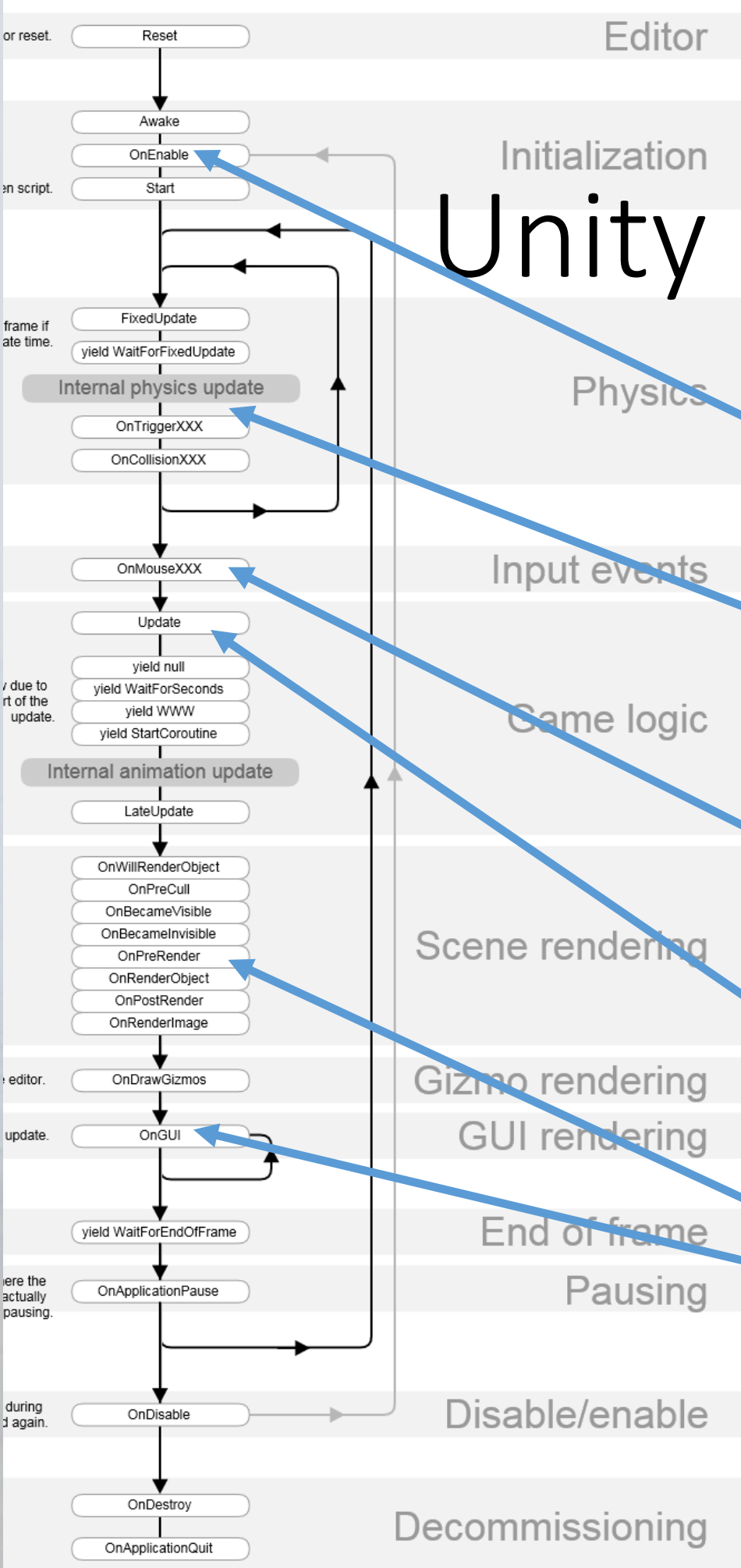
Input

**Update(), FixedUpdate(), LateUpdate()**

Actualización

**OnBecameVisible(), OnBecameInvisible() OnGUI()**

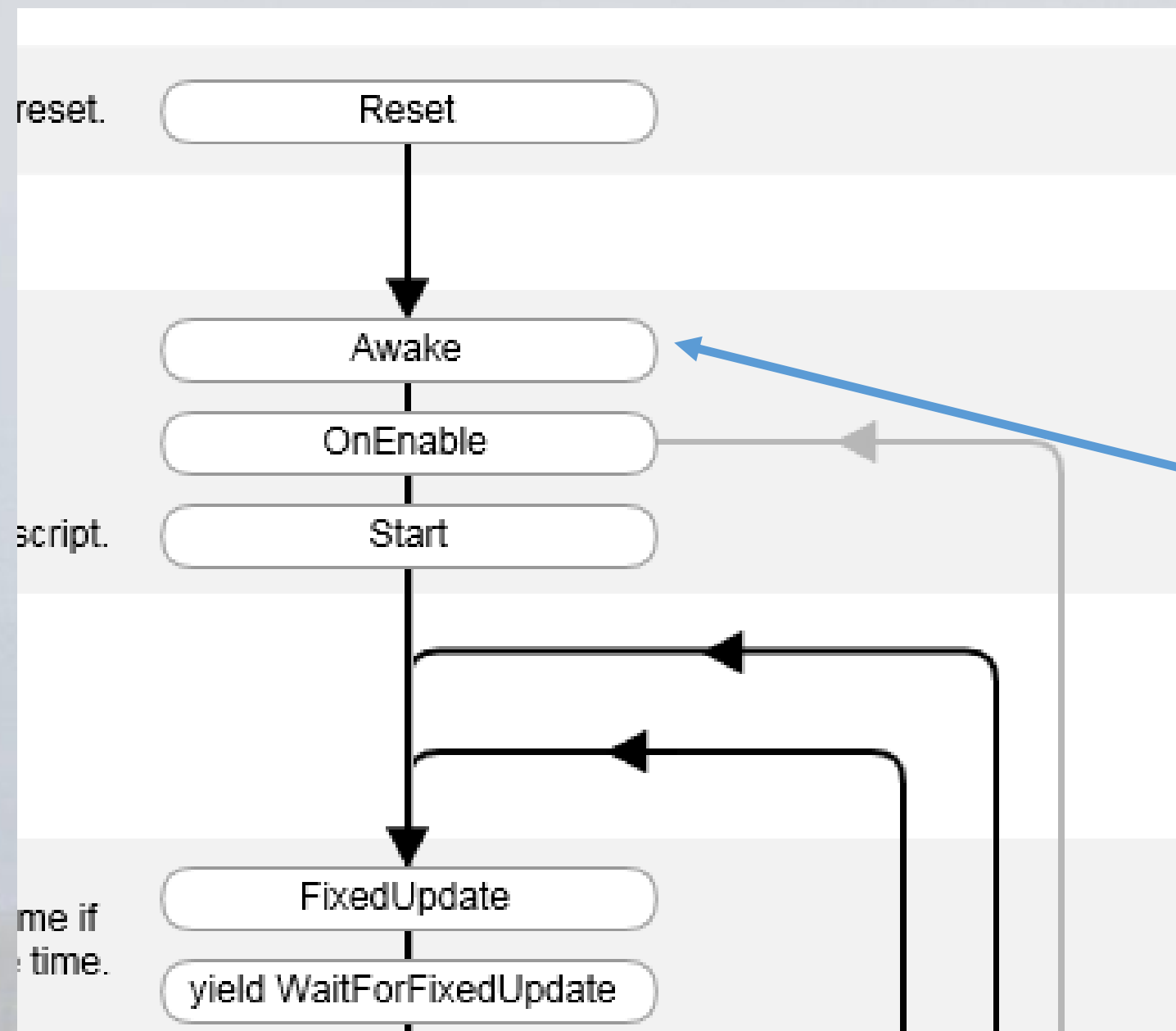
Output





# Unity Scripts (III)

## Awake()



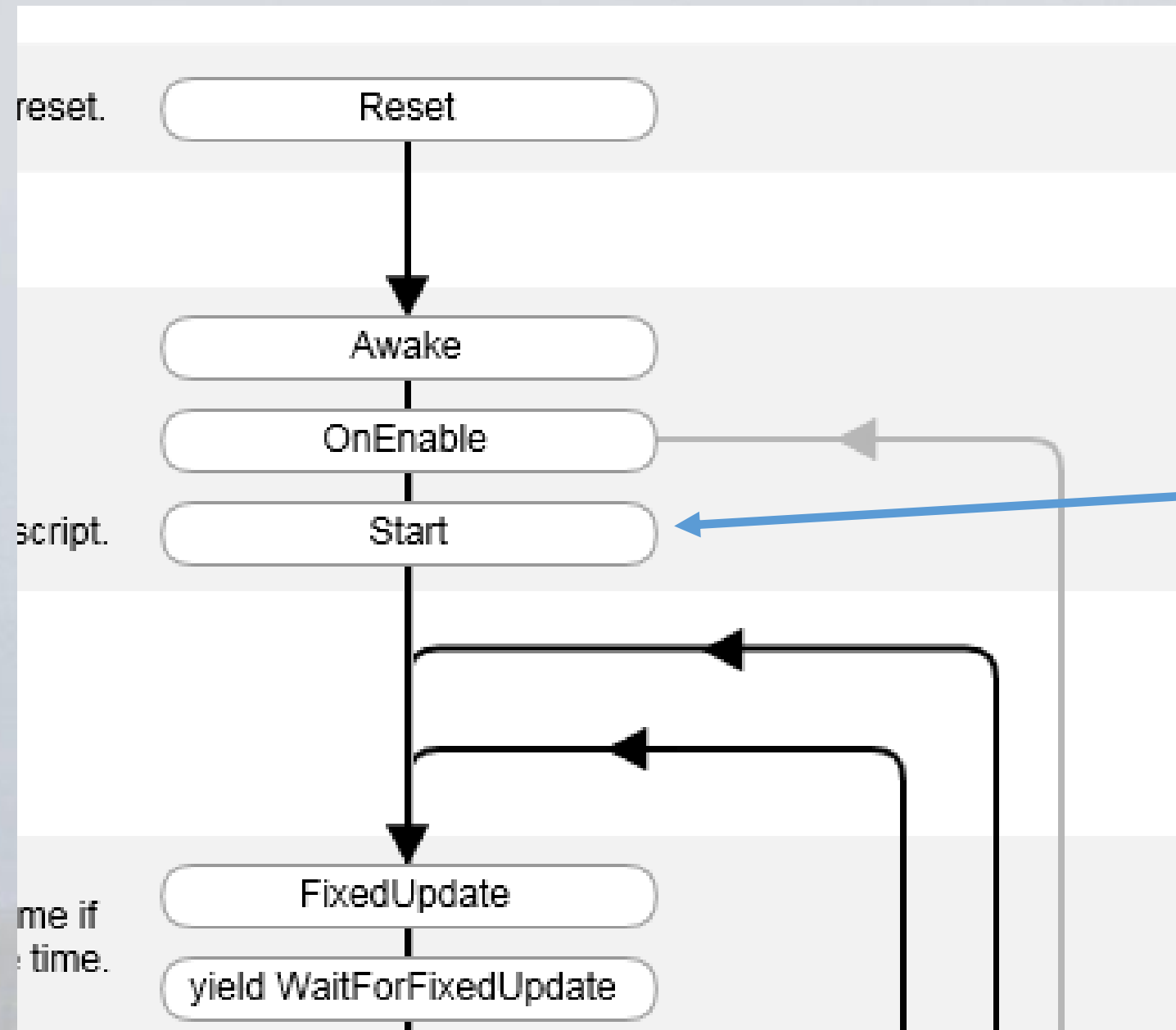
### AWAKE

- Es el primer método ejecutado cuando se inicializa un objeto
- Es el sustituto de los constructores en Unity3D
- Se ejecuta siempre antes del método *Start()* en todos los objetos
- Usar para inicializar referencias cruzadas entre objetos

# Unity Scripts (IV)

## Start()

Awake y Start sólo se ejecutan una única vez al arrancar una escena a la que pertenecen y por ese orden

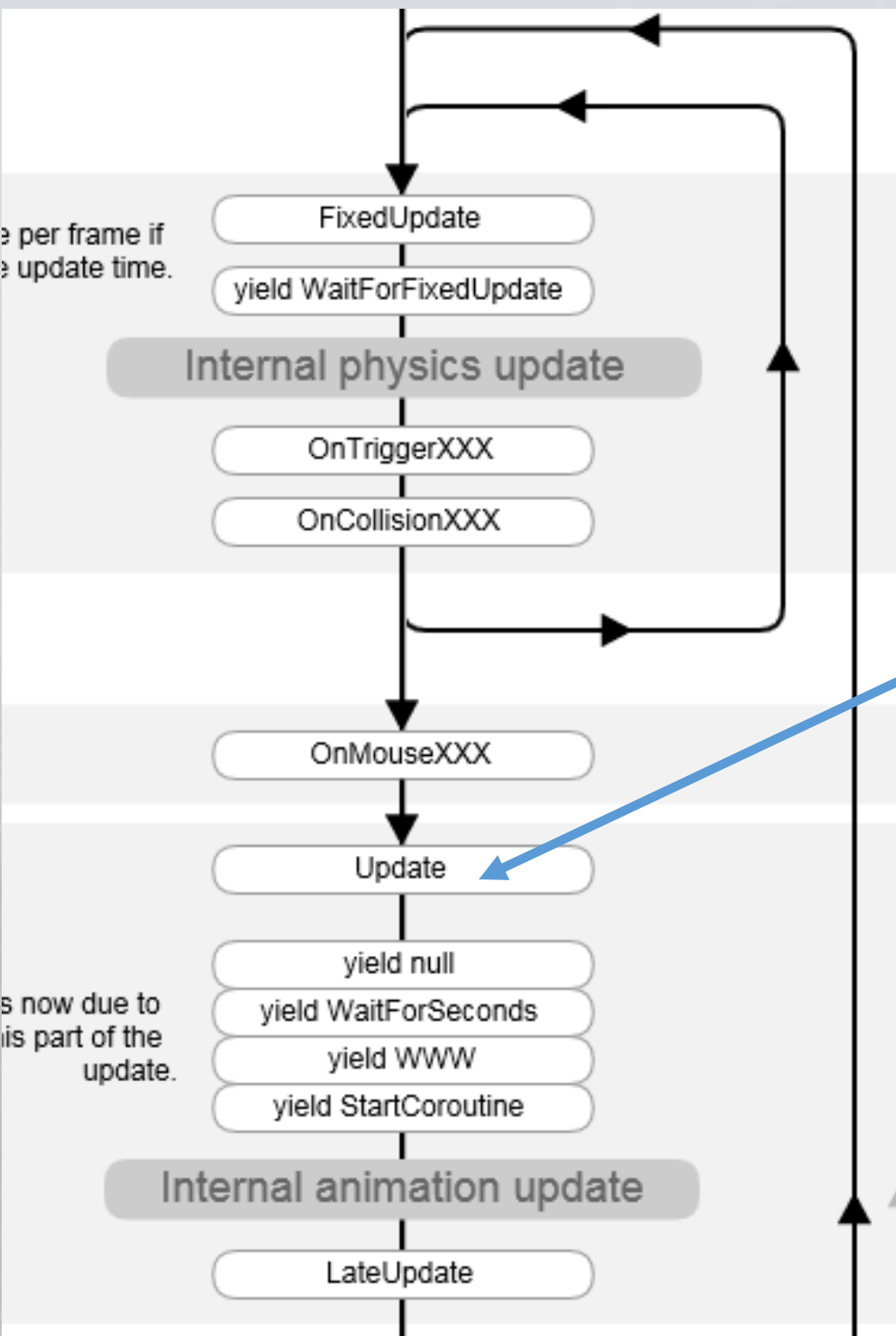


## START

- Es el segundo paso de inicialización después de *Awake()*
- Se ejecuta siempre antes de cualquier invocación al método *Update()*
- *Awake()* y *Start()* inician cualquier objeto en dos pasos
- *Start()* se invoca si objeto está habilitado (marcado en inspector)

# Unity Scripts (V)

Update(), FixedUpdate(), LateUpdate()

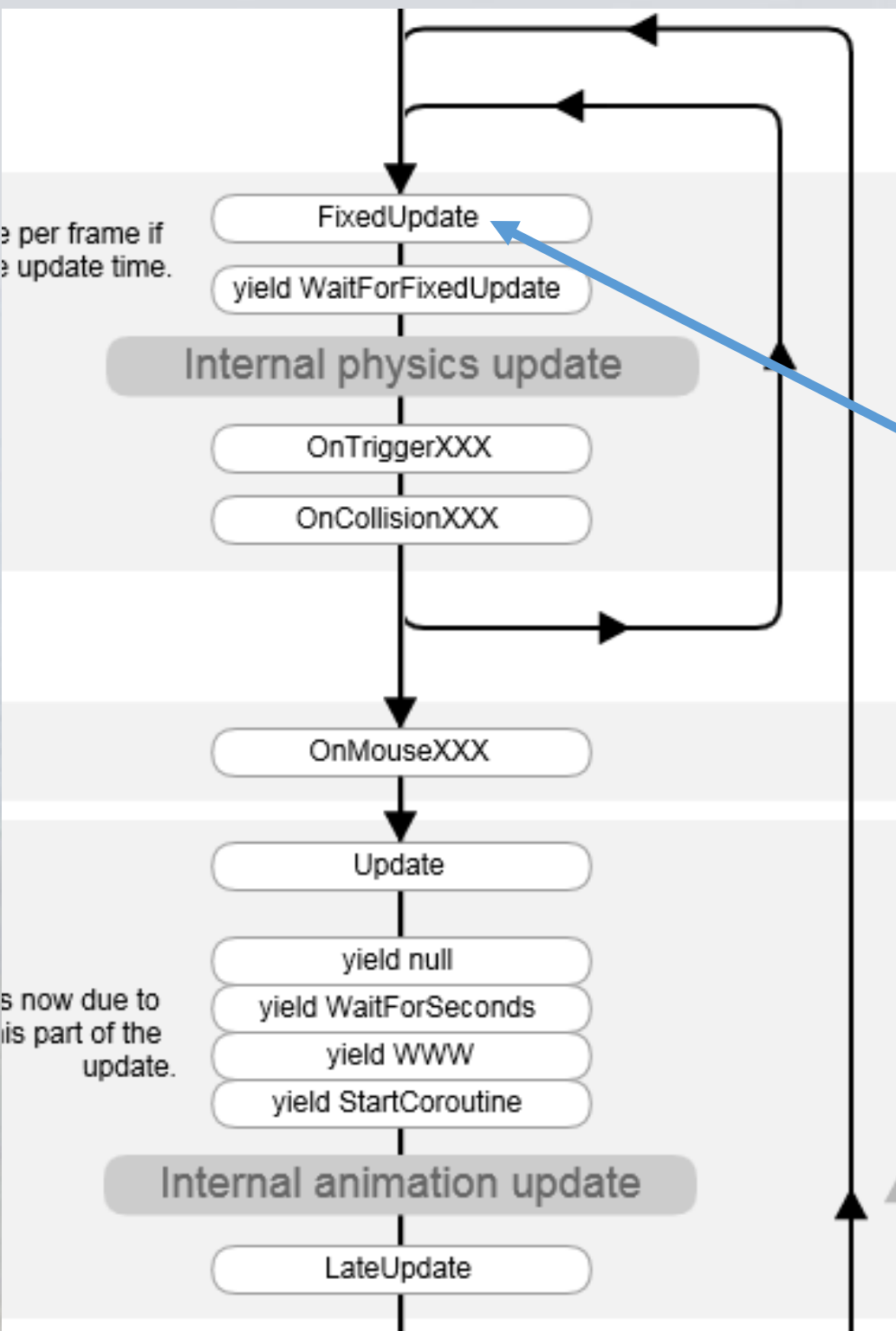


```
void Update() { ... }
```

- Se invoca previamente a cada actualización de la escena
- Se ejecuta sólo si el objeto está habilitado
- Realiza todos los cálculos que no dependen de otros objetos

# Unity Scripts (VI)

Update(), FixedUpdate(), LateUpdate()



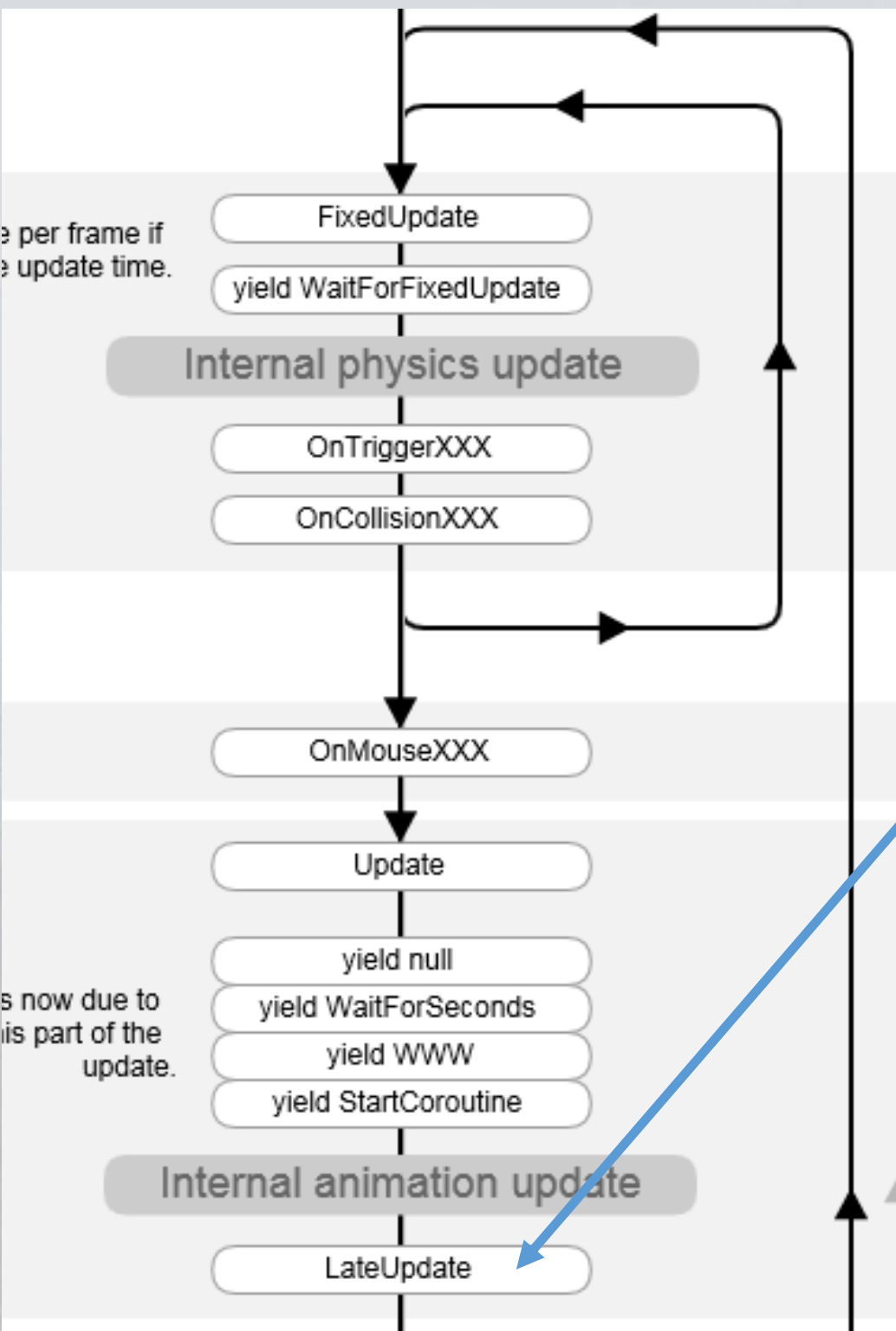
```
void FixedUpdate() { ... }
```

- Se invoca con una frecuencia de muestreo fija antes de que el objeto se actualice físicamente
- Sitio perfecto para realizar cambios en el *Rigidbody* del *GameObject*. Ej: fuerzas constantes
- La frecuencia de muestreo puede ser modificada en los ajustes del proyecto



# Unity Scripts (VII)

Update(), FixedUpdate(), LateUpdate()

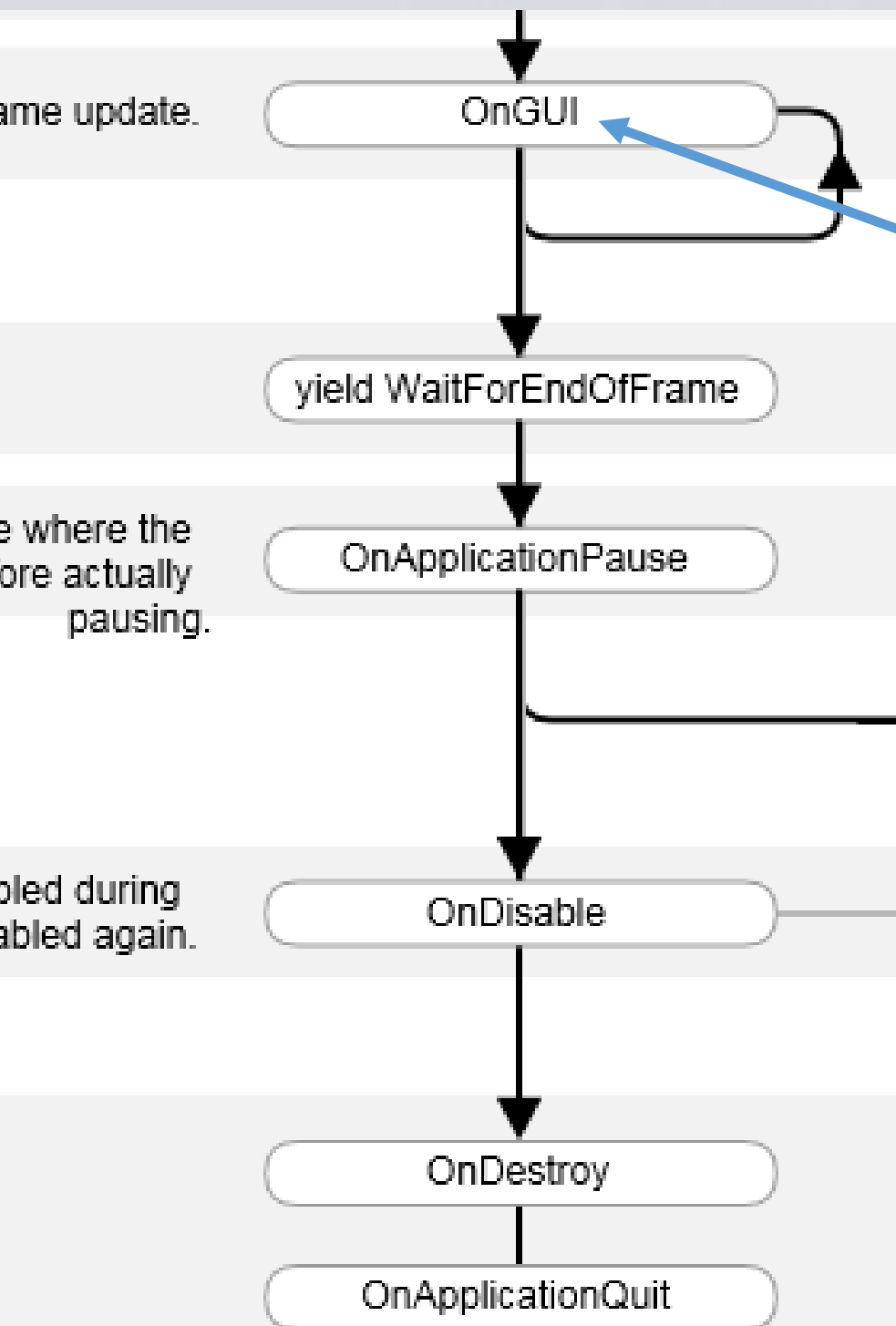


```
void LateUpdate() { ... }
```

- Opera como un *Update* de segundo nivel
- Se invoca después de invocar todos los métodos *Update()* de todos los *GameObjects*
- Emplear cuando el estado de un objeto A depende del estado de otro B. Tras la actualización de los dos, A y B, A verifica estado de B

# Unity Scripts (VIII)

OnGUI(), OnDisable(), OnDestroy()



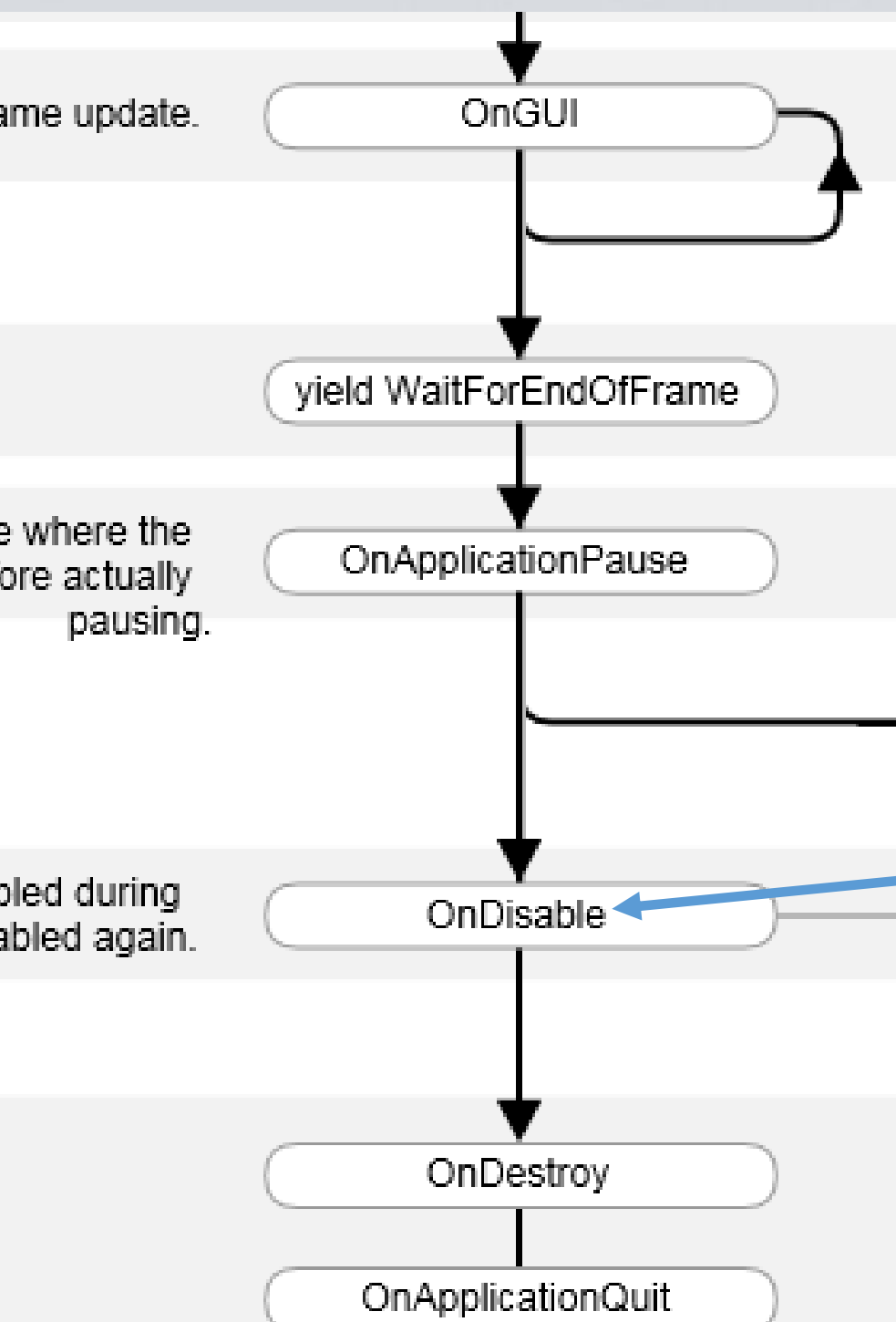
```
void OnGUI () { ... }
```

Siempre que haya que hacer uso  
botones, textos, texturas,...

Trabaja en una capa superior a los  
renders del juego

# Unity Scripts (VIII)

OnGUI(), OnDisable(), OnDestroy()



```
void OnDisable() { ... }
```

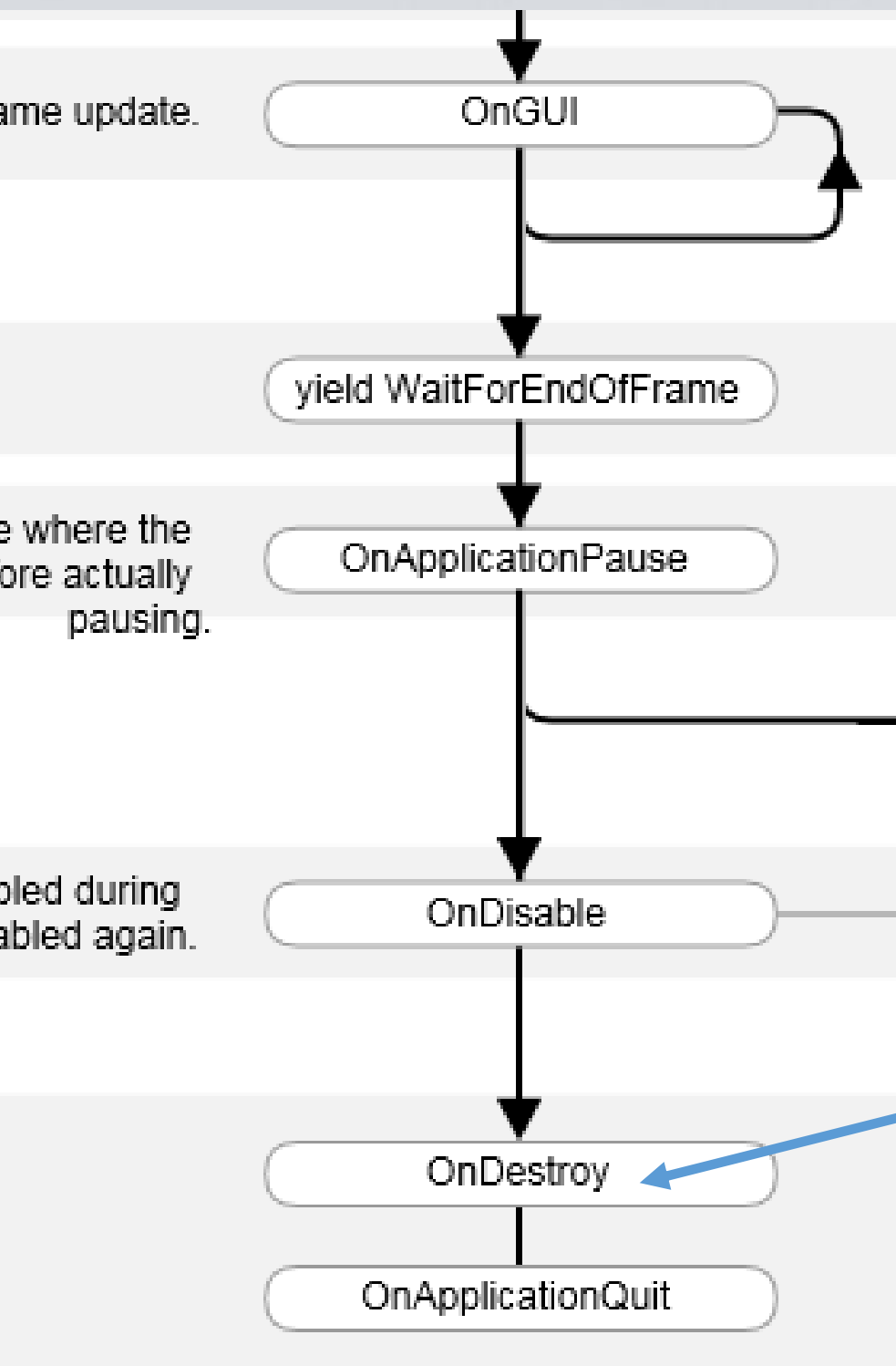
Se invoca cuando el objeto se destruye  
Se utiliza para

- Realizar cualquier tipo de limpieza
- Retirar el objeto del juego actual

Si el objeto vuelve a ser invocado para la siguiente fase del juego, nivel,... hay que ejecutar de nuevo *OnEnable*

# Unity Scripts (VIII)

OnGUI(), OnDisable(), OnDestroy()



```
void OnDestroy() { ... }
```

Sólo se puede invocar con  
objetos del juego que hayan  
estado previamente activos



# Bucle principal interno de Unity



## Initialization

## Physics

## Input events

## Game logic

## Scene rendering

## Gizmo rendering

## GUI rendering

End of frame

## Pausing

Disable/enable

## Decommissioning

# Bibliografía

Learning C# by Developing Games with Unity 3D Beginner's Guide. Terry Norton. Packt Publishing. ISBN 978-1-84969-658-6

Manuales en línea de Unity 3D



Documentación generada por  
Dr. Ramón Mollá Vayá  
Sección de Informática Gráfica  
Departamento de Sistemas Informáticos y Computación  
Universidad Politécnica de Valencia

### **Reconocimiento-NoComercial-CompartirIgual 2.5**

#### **Usted es libre de:**

copiar, distribuir y comunicar públicamente la obra  
hacer obras derivadas bajo las condiciones siguientes:



**Reconocimiento.** Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador.



**No comercial.** No puede utilizar esta obra para fines comerciales.



**Compartir bajo la misma licencia.** Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.

Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor

**Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.**