Event Functions

Unity has several helpful methods or “functions” as part of its MonoBehaviour code.
These methods help with:

- Initialisation
  - setting up a script when the game is loaded or started
- Updating
  - running code constantly, a “loop”
- GUI
  - handling input from buttons, the mouse etc
- Collision
  - GameObjects colliding or overlapping each other in the game

These are only a small amount of the event functions. Please consult the Unity Manual’s MonoBehaviour section for more.

Initialisation Event Functions

These deal with either:

- When the scene is loaded, or
- When the Component is enabled

Name	Description	Example

Awake()	Runs when the scene is loaded.
	Runs only once.
		void Awake() {
		
		}

Start()	Runs when the Component is enabled, just before the Update loop is started.
	Runs only once.
		void Start() {
		
		}

OnEnable()	Runs every time the Component is enabled.
	If the component disabled and enabled again, this will run again also.
		void OnEnable() {
		
		}

OnDisable()	Runs every time the Component is disabled. If the component is enabled, then disabled again, this will run also.	void OnDisable() {
		
		}

Update Event Functions

These methods run constantly, up to as fast as the game possible can, maybe 30-120 times per second!.
Update events are known as the “Game Loop”, where the core game code keeps track of what's happening in the scene, from things moving to scores going up/down.

Name	Description	Example

Update()	The most common type of Update event, this will run as fast as the game can, often at 30-120 times per second	void Update() {
		
		}

FixedUpdate()	This will run at a set speed. The speed is set in the Unity Project Settings.
	Go to Top Menu: Edit > Project Settings > Time.
	The “Fixed Timestep” value is how often per second the FixedUpdate() method will run.
		void FixedUpdate() {
		
		}

Physics Event Functions

These methods are used to handle what to do if the GameObject hits another GameObject.
Both the GameObjects involved need to have a Collider Component attached in order to use these methods

2D Collisions

There are separate methods for handling collisions and triggers from 2D Colliders.
Please refer to MonoBehaviour in the manual

The Collision methods are to handle physics collision. Active when the Colliders “Is Trigger” is set to **False**

Name	Description	Example

OnCollisionEnter2D()	When another GameObject first hits this one, this method runs.
	If the two are separated and collide again, the method runs again
		void OnCollisionEnter2D(Collision2D collision) {
		
		}

OnCollisionStay2D()	Runs every frame while staying in contact with another GameObject.	void OnCollisionStay2D(Collision2D collision) {
		
		}

OnCollisionExit2D()	Runs when the GameObject stops being in contact with this one.	void OnCollisionExit2D(Collision2D collision) {
		
		}

The Trigger methods are active when the Colliders “Is Trigger” is set to **True**

Name	Description	Example

OnTriggerEnter2D()	When another GameObject first enters this one, this method runs. If the two are separated and enter each other again, the method runs again.	void OnTriggerEnter2D(Collider2D other) {
		
		}

OnTriggerStay2D()	Runs every frame while the other GameObject is overlapping this one.	void OnTriggerStay2D(Collider2D other) {
		
		}

OnTriggerExit2D()	Runs when the GameObject stops overlapping this one.	void OnTriggerExit2D(Collider2D other) {
		
		}

GUI Event Functions

Deal with drawing GUI elements on the screen during gameplay.
Also deals with some aspects of Mouse Input

Name	Description	Example

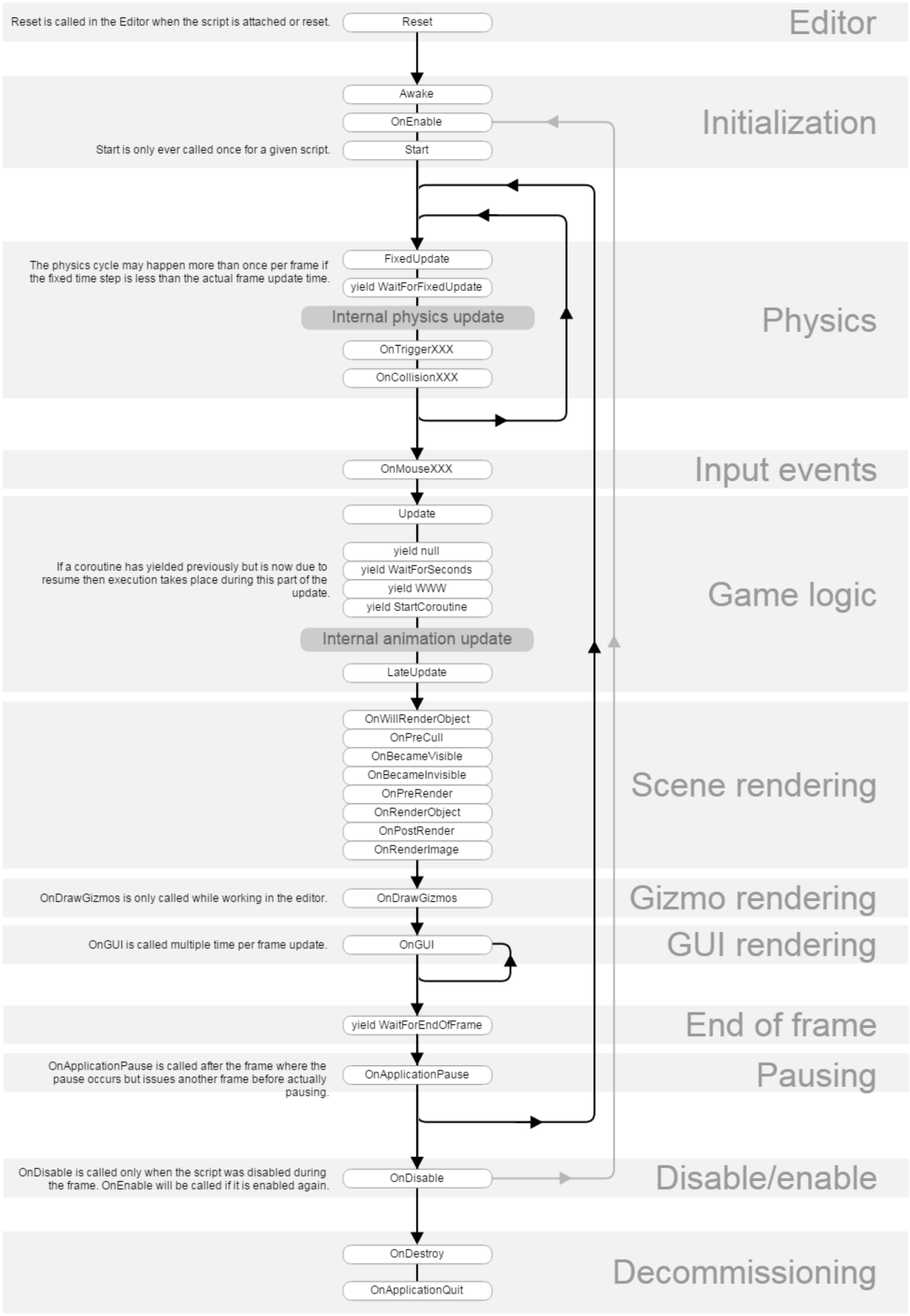
OnGUI()	Runs every frame. Code from the GUI library will run in this method only. This may be removed in later versions of Unity because of the GUI update in Unity version 4.6. More details here	void OnGUI() {
		
		}

OnMouseEnter()	Runs when the mouse enters the GameObject’s Collider.	void OnMouseEnter() {
		
		}

OnMouseDown()	Will run if the user left clicks the GameObject’s Collider.	void OnMouseDown() {
		
		}

A chart showing the order of execution of the event functions
[Execution Order of Event Functions - Manual]

All these Event Functions can be accessed from Monobehaviour
[Monobehaviour in the Unity API manual]




INTERACTIVE
SYSTEMS STUDIO
CGD