Developers @ TLL

**This document defines the coding standards and naming convention for The location Lab developers.**

Coding Standards

C# Coding standards and naming conventions

**[بسم الله الرحمن الرحيم](https://www.dawateislami.net/quran/surah-al-asr/ayat-0/translation-1/tafseer)**

[Table of Contents](https://www.dawateislami.net/quran/surah-al-asr/ayat-0/translation-1/tafseer)

[[Coding Standards 1](https://www.dawateislami.net/quran/surah-al-asr/ayat-0/translation-1/tafseer)](#_Toc64104809)

[[Naming: 1](https://www.dawateislami.net/quran/surah-al-asr/ayat-0/translation-1/tafseer)](#_Toc64104810)

[[Coding Style: 2](https://www.dawateislami.net/quran/surah-al-asr/ayat-0/translation-1/tafseer)](#_Toc64104811)

[[Coding Sequence: 3](https://www.dawateislami.net/quran/surah-al-asr/ayat-0/translation-1/tafseer)](#_Toc64104812)

[[Refrences: 3](https://www.dawateislami.net/quran/surah-al-asr/ayat-0/translation-1/tafseer)](#_Toc64104813)

**Unity/C# Coding Standards and Naming Convention**

**What is:** The document will provide you about the general guideline of coding and unity project organization.

**Why:** We are want to make our code based consistent and easily maintainable.

There are two part of the document, the one is dealing with **coding** the other one is dealing with project **organization.**

# Coding Standards

## Naming:

1. Never use Spaces in names.
2. Use **camelCase** for variable and method parameters

|  |
| --- |
| 1. **public** **GameObject** activeCam;//right 2. public **GameObject** ActiveCam;//wrong |

1. Use **PascalCase** for methods, namespace, class, struct, properties, Enum name.

|  |
| --- |
| 1. **public** **class** CamController : Page 2. { 3. //... 4. } |

1. Use **PascalCase** for custom file- and folder names, like this: UtilityClasses, ProjectIcons. Do not use spaces, underscores, or hyphens, with one exception (see Naming Different Aspects of the Same Thing).
2. Use namespace for Project specific classes.
3. User abbreviation for unity UI Controls in coding. Like use BtnVersionSwitch not ButtonVersionSwitch or use

|  |  |
| --- | --- |
| Abbreviations | Standard Control |
| btn | Button |
| cb | CheckBox |
| ddl | DropDownList |
| fu | FileUpload |
| img | Image |
| lbl | Label |
| pnl | Panel |
| txt | InputField |
| tbl | Table |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Abbreviations | Programming Constructs |
| Class | Cls (not recommended) |
| Delegate | Dlg |
| Event | Event |
| I | Interface |
|  |  |

|  |  |
| --- | --- |
| Abbreviations | Programming Constructs |
| Cam | Camera |
| PP | PostProcess |
|  |  |
| I | Interface |
|  |  |

1. Keep the most common descriptor on the left: postionX, not xPosition; BtnPause, not PauseBtn.
2. Some names form a sequence. Use numbers in these names, for example, PathNode0, PathNode1. Always start with 0, not 1.
3. Write descriptive name
4. Prefix temporary objects with a double underscore and add Backup string after it \_\_Player\_Backup. Like \_\_MainScene\_Backup or Gameobject Backup \_\_CameraController\_Backup or a method backup \_\_ActiveCamera\_Backup.
5. For inspector, Use Header[] for a collection of variable or a variable required more explanation.
6. Events are associated with actions. Therefore, events are name with verbs: TiggerOccured, ButtonClicked, SceneLoaded etc
7. If a method run in response of an event use EventHandler as suffix. ClickEventHandler, TriggerEventHandler

## Coding Style:

1. Must add summary of the class the show three things: Author, Purpose, Decencies, Create Date

|  |
| --- |
| 1. /// <summary> 2. /// Author: Ali 3. /// Create Date: 2021-02-02 4. /// Purpose: Control multiple camera and its switching 5. /// Dependencies (if any): CameraScript, CameraUI\_Link, ScriptReference 6. /// </summary> 7. public class CameraController : MonoBehaviour |

1. Must mention access modified even if the variable is private.
2. Don use public variable if they are not required to access from other classes. User Serialized filed.
3. User Singleton or Events for avoid dependencies on other scripts.

## Coding Sequence:

1. Declare member variable at the top of the class and if the member is static then, place it on top most.
2. User specific #Regions for placing the code. The sequence will be Vars Region, UnityEvents Region, Custom Methods, and Custom EventHandlers.

## References:

1. https://www.c-sharpcorner.com/UploadFile/8a67c0/C-Sharp-coding-standards-and-naming-conventions/