

Coding Standards

File Formatting


Indentation

Code levels should be indented by using the tab key in order to offset the code.

Libraries

Libraries should be organized at the top of the script.

Example:

```
 using System.Collections;  
    using System.Collections.Generic;  
    using UnityEngine;  
    using System;
```

Spacing

Developers should follow the below spacing guidelines:

- Should have at least one blank line below their libraries before writing code.
- Should have at least one blank line below their curly braces after a class/function before writing code.
- Place one blank line to separate public member variables from private member

Naming Conventions

Classes

Classes should be named using an UpperCamelCase scheme, in this the first letter of each word should be capitalized in order to signify another word.

Example:

```
 public class PlayerScript : MonoBehaviour
```

Functions

Function names should utilize the PascalCase naming scheme where the first letter of each word is capitalized. There should be no special characters or numbers present in the function name.

Example:

```
void OnApplicationPause()  
{  
    worldCursorPoint = transform.position;  
    screenCursorPoint = Camera.main.WorldToScreenPoint(worldCursorPoint);  
}
```

Variables

Variables will be represented by a normal camelCase structure.

Example:

damagePlayer

Comments

Functions/ Classes

Function comments should be placed above the said function and in a paragraph style comment explaining what the function below does.

Additional Comments

Comments should be denoted with a //

Other Formatting

Files

The following general rules should be followed when formatting your code:

- Max line length should not exceed more than ~120 characters
- Curly braces should be placed on the line after function, class, or other declarations

Classes

- Public variables should appear before private variables in classes • Public functions should appear before private functions in classes
- Unity functions should appear before custom functions