horizontal line

Coding Standards

# Naming

## Classes

|  |  |
| --- | --- |
| public class **Weapon** : MonoBehaviour {} | * Use Pascal Case * Class Names are Nouns |

## Properties

|  |  |
| --- | --- |
| public class Weapon : MonoBehaviour  {  public int **Damage** { get; set; }  } | * Use Pascal Case * Nouns or Adjectives |

## Fields

|  |  |
| --- | --- |
| public class Weapon : MonoBehaviour  {  private int **\_health;**  } | * camelCase * \_underscore * Nouns or Adjectives |

## Methods

|  |  |
| --- | --- |
| public class Weapon : MonoBehaviour  {  public int **FireWeapon()** {}  } | * Use Pascal Case * Verbs |

## Method Parameters

|  |  |
| --- | --- |
| public class Player : MonoBehaviour  {  public int TakeDamage(**int amount**)  {}  } | * camelCase * Nouns or Adjectives |

## 

# Naming Continued...

## Constants

|  |  |
| --- | --- |
| public class Player : MonoBehaviour  {  public const int **MAX\_PLAYERS** = 3;  } | * SCREAMING\_CAPS * Nouns or Adjectives |

## Events

|  |  |
| --- | --- |
| public class Player : MonoBehaviour  {  public event Action<int> OnTookHit;  } | * Pascal Case * ‘On Something Happened’ |

# Abbreviations

* Avoid abbreviations except in cases where it’s a domain specific common term.
  + Example: Prefer **playerWeapon** over *pWeap*
  + Exception Example: Prefer **GPU** over *GraphicalProcessingUnit*
* Avoid single character names except as loop iterators with ‘i’
  + If multiple iterators are needed, give them proper names