

## Project Calamity

Generated by Doxygen 1.9.5



---

<b>1 Project Calamity</b>	<b>1</b>
<b>2 Namespace Index</b>	<b>3</b>
2.1 Package List . . . . .	3
<b>3 Hierarchical Index</b>	<b>5</b>
3.1 Class Hierarchy . . . . .	5
<b>4 Class Index</b>	<b>7</b>
4.1 Class List . . . . .	7
<b>5 Namespace Documentation</b>	<b>9</b>
5.1 PC Namespace Reference . . . . .	9
5.2 PC.Combat Namespace Reference . . . . .	9
5.3 PC.Entities Namespace Reference . . . . .	9
5.4 PC.Input Namespace Reference . . . . .	10
5.5 PC.Stats Namespace Reference . . . . .	10
5.6 PC.UI Namespace Reference . . . . .	10
5.7 PC.VFX Namespace Reference . . . . .	11
5.7.1 Detailed Description . . . . .	11
<b>6 Class Documentation</b>	<b>13</b>
6.1 PC.UI.AlphaPMS Class Reference . . . . .	13
6.2 PC.UI.BetaPMS Class Reference . . . . .	13
6.3 PC.UI.ButtonAction Struct Reference . . . . .	13
6.4 PC.Combat.CharacterCombat Class Reference . . . . .	13
6.4.1 Member Function Documentation . . . . .	14
6.4.1.1 Attack() . . . . .	14
6.5 PC.Stats.CharacterStats Class Reference . . . . .	14
6.5.1 Member Function Documentation . . . . .	14
6.5.1.1 Die() . . . . .	14
6.6 PC.UI.Container Class Reference . . . . .	14
6.6.1 Member Function Documentation . . . . .	15
6.6.1.1 Start() . . . . .	15
6.7 PC.UI.ContainerBase Class Reference . . . . .	15
6.7.1 Member Function Documentation . . . . .	16
6.7.1.1 EmptyCell() . . . . .	16
6.7.1.2 GetCell() . . . . .	17
6.7.1.3 GetCellIndex() . . . . .	17
6.7.1.4 GetItemAt() . . . . .	17
6.7.1.5 InitBackground() . . . . .	18
6.7.1.6 InitContents() . . . . .	18
6.7.1.7 OnPointerEnter() . . . . .	18
6.7.1.8 OnPointerExit() . . . . .	18

---

6.7.1.9 PlaceItemAt()	19
6.7.1.10 RemoveItemAt()	19
6.7.1.11 SetCell()	19
6.7.1.12 SetItemParent()	20
6.7.1.13 TakeItemAt()	20
6.7.1.14 TransferItem()	20
6.8 PC.UI.DevConsoleMenu Class Reference	20
6.8.1 Member Function Documentation	21
6.8.1.1 AwakeExtension()	21
6.8.1.2 CloseExtension()	21
6.8.1.3 OpenExtension()	21
6.9 PC.Combat.EnemyAnimationController Class Reference	22
6.10 PC.Combat.EnemyCombat Class Reference	22
6.10.1 Member Function Documentation	22
6.10.1.1 AttackTarget()	22
6.10.1.2 SetTags()	23
6.11 PC.Entities.EnemyController Class Reference	23
6.12 PC.Stats.EnemyStats Class Reference	23
6.12.1 Member Function Documentation	23
6.12.1.1 Die()	23
6.13 PC.Entities.Gun Class Reference	23
6.13.1 Member Function Documentation	24
6.13.1.1 Reload()	24
6.13.1.2 Shoot()	24
6.14 PC.Entities.GunSO Class Reference	24
6.15 PC.UI.HorizontalLayoutGroupHandler Class Reference	25
6.16 PC.Input.InputModule Class Reference	25
6.17 PC.UI.Interactable Class Reference	26
6.18 PC.UI.InventoryMenu Class Reference	26
6.18.1 Member Function Documentation	27
6.18.1.1 AwakeExtension()	27
6.18.1.2 CloseExtension()	27
6.18.1.3 OpenExtension()	27
6.18.1.4 TryPickingUpItem()	27
6.18.1.5 TryReleasingItem()	27
6.19 PC.UI.Item Class Reference	28
6.19.1 Member Function Documentation	29
6.19.1.1 GetOriginCellIndex()	29
6.19.1.2 Init()	29
6.19.1.3 MakeCopy()	29
6.19.1.4 TransferTo()	30
6.20 PC.UI.ItemContainerInfo Class Reference	30

---

6.20.1 Constructor & Destructor Documentation	30
6.20.1.1 ItemContainerInfo()	30
6.20.2 Member Function Documentation	31
6.20.2.1 Create()	31
6.21 PC.UI.ItemSO Class Reference	31
6.22 PC.UI.ItemType Class Reference	31
6.22.1 Member Function Documentation	33
6.22.1.1 belongsTo()	33
6.22.1.2 Equals()	33
6.22.1.3 Lookup()	33
6.22.1.4 operator!=() [1/2]	34
6.22.1.5 operator!=() [2/2]	34
6.22.1.6 operator==() [1/2]	34
6.22.1.7 operator==() [2/2]	34
6.23 PC.UI.ItemTypes Class Reference	35
6.23.1 Member Function Documentation	35
6.23.1.1 belongsTo()	35
6.24 PC.UI.LayoutMaintainPreferredHeight Class Reference	35
6.25 PC.UI.MapMenu Class Reference	36
6.25.1 Member Function Documentation	36
6.25.1.1 AwakeExtension()	36
6.25.1.2 CloseExtension()	36
6.25.1.3 OpenExtension()	36
6.26 PC.UI.MenuBase Class Reference	37
6.26.1 Member Function Documentation	38
6.26.1.1 AwakeExtension()	38
6.26.1.2 CloseExtension()	38
6.26.1.3 OpenExtension()	38
6.27 PC.UI.MenusController Class Reference	38
6.28 PC.UI.PauseMenu Class Reference	39
6.28.1 Member Function Documentation	39
6.28.1.1 AwakeExtension()	40
6.28.1.2 CloseExtension()	40
6.28.1.3 OpenExtension()	40
6.29 PC.Entities.PlayerAnimatorController Class Reference	40
6.30 PC.Combat.PlayerCombat Class Reference	41
6.30.1 Member Function Documentation	41
6.30.1.1 AttackTarget()	41
6.31 PC.Entities.PlayerController Class Reference	41
6.31.1 Member Function Documentation	42
6.31.1.1 OnAnimatorIK()	42
6.32 PC.Entities.PlayerControllerBase Class Reference	42

---

6.32.1 Member Function Documentation	42
6.32.1.1 GetMovement()	42
6.32.1.2 GroundedCheck()	43
6.33 PC.UI.PlayerManagementStationMenu Class Reference	43
6.33.1 Member Function Documentation	43
6.33.1.1 AwakeExtension()	43
6.33.1.2 CloseExtension()	43
6.33.1.3 OpenExtension()	43
6.34 PC.Entities.PlayerManager Class Reference	44
6.35 PC.Stats.PlayerStats Class Reference	44
6.35.1 Member Function Documentation	44
6.35.1.1 Die()	44
6.36 PC.UI.PMSBase Class Reference	44
6.37 PC.UI.PMSMenu Class Reference	45
6.37.1 Member Function Documentation	45
6.37.1.1 AwakeExtension()	45
6.37.1.2 CloseExtension()	46
6.37.1.3 Init() [1/2]	46
6.37.1.4 Init() [2/2]	47
6.37.1.5 OpenExtension()	47
6.38 PC.UI.PopupWindow Class Reference	47
6.38.1 Member Function Documentation	48
6.38.1.1 AddButton()	48
6.38.1.2 Create() [1/2]	48
6.38.1.3 Create() [2/2]	49
6.38.1.4 Init()	49
6.39 PC.UI.ScrollRectNoMouseDrag Class Reference	50
6.40 PC.UI.Slot Class Reference	50
6.40.1 Member Function Documentation	50
6.40.1.1 GetCellIndex()	50
6.40.1.2 GetItem()	51
6.40.1.3 PlaceItemAt()	51
6.40.1.4 RemoveItemAt()	51
6.40.1.5 Start()	52
6.41 PC.Stats.Stat Class Reference	52
6.41.1 Member Function Documentation	52
6.41.1.1 GetValue()	52
6.42 PC.Combat.TouchSensor Class Reference	53
6.43 PC.UI.VerticalLayoutGroupHandler Class Reference	53
6.44 PC.VFX.WFX_LightFlicker Class Reference	53
6.44.1 Member Function Documentation	53
6.44.1.1 Flicker()	53







# Chapter 1

## Project Calamity

Dead Space and the Alien franchise were pioneers in the sci-fi horror game genre with the last of these games being Alien Isolation released in 2014. There have been several titles of the same ilk released since then, but none have reached the same level of popularity.

In an effort to reclaim and reinterpret the magic of the originals and take advantage of this hole in the market, a new slate of space station horror shooters were announced at the 2022 Summer Game Fest event. Our team used this opportunity to develop our own demo in the genre.

### Warning

This game is not suitable for children or those who are easily scared.



# Chapter 2

## Namespace Index

### 2.1 Package List

Here are the packages with brief descriptions (if available):

<a href="#">PC</a> . . . . .	9
<a href="#">PC.Combat</a> . . . . .	9
<a href="#">PC.Entities</a> . . . . .	9
<a href="#">PC.Input</a> . . . . .	10
<a href="#">PC.Stats</a> . . . . .	10
<a href="#">PC.UI</a> . . . . .	10
<a href="#">PC.VFX</a> . . . . .	11



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

BaselInputModule	
PC.InputInputModule . . . . .	25
PC.UI.ButtonAction . . . . .	13
IPointerEnterHandler	
PC.UI.ContainerBase . . . . .	15
PC.UI.Container . . . . .	14
PC.UI.Slot . . . . .	50
IPointerExitHandler	
PC.UI.ContainerBase . . . . .	15
PC.UI.ItemContainerInfo . . . . .	30
PC.UI.ItemType . . . . .	31
MonoBehaviour	
PC.Combat.CharacterCombat . . . . .	13
PC.Combat.EnemyCombat . . . . .	22
PC.Combat.PlayerCombat . . . . .	41
PC.Combat.EnemyAnimationController . . . . .	22
PC.Combat.TouchSensor . . . . .	53
PC.Entities.EnemyController . . . . .	23
PC.Entities.Gun . . . . .	23
PC.Entities.PlayerAnimatorController . . . . .	40
PC.Entities.PlayerControllerBase . . . . .	42
PC.Entities.PlayerController . . . . .	41
PC.Entities.PlayerManager . . . . .	44
PC.Stats.CharacterStats . . . . .	14
PC.Stats.EnemyStats . . . . .	23
PC.Stats.PlayerStats . . . . .	44
PC.UI.ContainerBase . . . . .	15
PC.UI.HorizontalLayoutGroupHandler . . . . .	25
PC.UI.Interactable . . . . .	26
PC.UI.PMSBase . . . . .	44
PC.UI.AlphaPMS . . . . .	13
PC.UI.BetaPMS . . . . .	13
PC.UI.Item . . . . .	28
PC.UI.ItemTypes . . . . .	35
PC.UI.LayoutMaintainPreferredHeight . . . . .	35

---

PC.UI.MenuBase . . . . .	37
PC.UI.DevConsoleMenu . . . . .	20
PC.UI.InventoryMenu . . . . .	26
PC.UI.MapMenu . . . . .	36
PC.UI.PMSMenu . . . . .	45
PC.UI.PauseMenu . . . . .	39
PC.UI.PlayerManagementStationMenu . . . . .	43
PC.UI.MenusController . . . . .	38
PC.UI.PopupWindow . . . . .	47
PC.UI.VerticalLayoutGroupHandler . . . . .	53
PC.VFX.WFX_LightFlicker . . . . .	53
ScriptableObject	
PC.Entities.GunSO . . . . .	24
PC.UI.ItemSO . . . . .	31
ScrollRect	
PC.UI.ScrollRectNoMouseDrag . . . . .	50
PC.Stats.Stat . . . . .	52

# Chapter 4

## Class Index

### 4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">PC.UI.AlphaPMS</a>	13
<a href="#">PC.UI.BetaPMS</a>	13
<a href="#">PC.UI.ButtonAction</a>	13
<a href="#">PC.Combat.CharacterCombat</a>	
Defined the combat behavior specific to the player character.	13
<a href="#">PC.Stats.CharacterStats</a>	
This class is used to store general stats and damage behavior common to all game characters.	14
<a href="#">PC.UI.Container</a>	14
<a href="#">PC.UI.ContainerBase</a>	15
<a href="#">PC.UI.DevConsoleMenu</a>	20
<a href="#">PC.Combat.EnemyAnimationController</a>	
Controls how the enemy animations react to information in the environment.	22
<a href="#">PC.Combat.EnemyCombat</a>	
Defines the combat behavior specific to our enemies.	22
<a href="#">PC.Entities.EnemyController</a>	
<a href="#">EnemyController</a> is a class that handles the enemy's AI. The enemy is programmed with a navmeshagent to follow and attack the player when the player is in a specific range.	23
<a href="#">PC.Stats.EnemyStats</a>	
This class is used to store the enemy's stats and log its death.	23
<a href="#">PC.Entities.Gun</a>	
Defines the general behavior of the player's gun.	23
<a href="#">PC.Entities.GunSO</a>	
This class defines various parameters that control the gun's functionality and behavior. For instance, the gun's damage, range, and fire rate default values are defined here.	24
<a href="#">PC.UI.HorizontalLayoutGroupHandler</a>	25
<a href="#">PC.Input.InputModule</a>	25
<a href="#">PC.UI.Interactable</a>	26
<a href="#">PC.UI.InventoryMenu</a>	26
<a href="#">PC.UI.Item</a>	28
<a href="#">PC.UI.ItemContainerInfo</a>	30
<a href="#">PC.UI.ItemSO</a>	31
<a href="#">PC.UI.ItemType</a>	31
<a href="#">PC.UI.ItemTypes</a>	35
<a href="#">PC.UI.LayoutMaintainPreferredHeight</a>	35

<a href="#">PC.UI.MapMenu</a>	36
<a href="#">PC.UI.MenuBase</a>	37
<a href="#">PC.UI.MenusController</a>	38
<a href="#">PC.UI.PauseMenu</a>	39
<a href="#">PC.Entities.PlayerAnimatorController</a>	
This class is responsible for controlling the player's animations. For our purposes, it simply regularly updates the variables used by the animator to determine which animation to play.	40
<a href="#">PC.Combat.PlayerCombat</a>	
Defines the combat behavior specific to our player.	41
<a href="#">PC.Entities.PlayerController</a>	
Defines functionality and attributes specific to our player character.	41
<a href="#">PC.Entities.PlayerControllerBase</a>	
Defined functionality and attributes common to all player controllers such as ground checking and translating inputs to movement information.	42
<a href="#">PC.UI.PlayerManagementStationMenu</a>	43
<a href="#">PC.Entities.PlayerManager</a>	
Creates a global instance of the player for easy reference by other scripts. Handles interface between player health and UI.	44
<a href="#">PC.Stats.PlayerStats</a>	
This class is used to store the player's stats and log its death to the console.	44
<a href="#">PC.UI.PMSBase</a>	44
<a href="#">PC.UI.PMSMenu</a>	45
<a href="#">PC.UI.PopupWindow</a>	47
<a href="#">PC.UI.ScrollRectNoMouseDrag</a>	50
<a href="#">PC.UI.Slot</a>	50
<a href="#">PC.Stats.Stat</a>	
Base class for all stats used to track character parameters.	52
<a href="#">PC.Combat.TouchSensor</a>	
Records any contact between the player and the enemy. This allows the enemy to apply damage to the player on contact.	53
<a href="#">PC.UI.VerticalLayoutGroupHandler</a>	53
<a href="#">PC.VFX.WFX_LightFlicker</a>	
Controls the flickering of the light for the muzzle flash.	53



## Chapter 5

# Namespace Documentation

### 5.1 PC Namespace Reference

#### Namespaces

- namespace [VFX](#)

### 5.2 PC.Combat Namespace Reference

#### Classes

- class [CharacterCombat](#)  
*Defined the combat behavior specific to the player character.*
- class [EnemyAnimationController](#)  
*Controls how the enemy animations react to information in the environment.*
- class [EnemyCombat](#)  
*Defines the combat behavior specific to our enemies.*
- class [PlayerCombat](#)  
*Defines the combat behavior specific to our player.*
- class [TouchSensor](#)  
*Records any contact between the player and the enemy. This allows the enemy to apply damage to the player on contact.*

### 5.3 PC.Entities Namespace Reference

#### Classes

- class [EnemyController](#)  
*[EnemyController](#) is a class that handles the enemy's AI.  
The enemy is programmed with a navmeshagent to follow and attack the player when the player is in a specific range.*
- class [Gun](#)  
*Defines the general behavior of the player's gun.*
- class [GunSO](#)

*This class defines various parameters that control the gun's functionality and behavior. For instance, the gun's damage, range, and fire rate default values are defined here.*

- class [PlayerAnimatorController](#)

*This class is responsible for controlling the player's animations. For our purposes, it simply regularly updates the variables used by the animator to determine which animation to play.*

- class [PlayerController](#)

*Defines functionality and attributes specific to our player character.*

- class [PlayerControllerBase](#)

*Defined functionality and attributes common to all player controllers such as ground checking and translating inputs to movement information.*

- class [PlayerManager](#)

*Creates a global instance of the player for easy reference by other scripts. Handles interface between player health and UI.*

## 5.4 PC.Input Namespace Reference

### Classes

- class [InputModule](#)

## 5.5 PC.Stats Namespace Reference

### Classes

- class [CharacterStats](#)

*This class is used to store general stats and damage behavior common to all game characters.*

- class [EnemyStats](#)

*This class is used to store the enemy's stats and log its death.*

- class [PlayerStats](#)

*This class is used to store the player's stats and log its death to the console.*

- class [Stat](#)

*Base class for all stats used to track character parameters.*

## 5.6 PC.UI Namespace Reference

### Classes

- class [AlphaPMS](#)
- class [BetaPMS](#)
- struct [ButtonAction](#)
- class [Constants](#)
- class [Container](#)
- class [ContainerBase](#)
- class [DevConsoleMenu](#)
- class [HorizontalLayoutGroupHandler](#)
- class [Interactable](#)
- class [InventoryMenu](#)

- class [Item](#)
- class [ItemContainerInfo](#)
- class [ItemSO](#)
- class [ItemType](#)
- class [ItemTypes](#)
- class [LayoutMaintainPreferredHeight](#)
- class [MapMenu](#)
- class [MenuBase](#)
- class [MenusController](#)
- class [PauseMenu](#)
- class [PlayerManagementStationMenu](#)
- class [PMSBase](#)
- class [PMSMenu](#)
- class [PopupWindow](#)
- class [ScrollRectNoMouseDrag](#)
- class [Slot](#)
- class [VerticalLayoutGroupHandler](#)

## 5.7 PC.VFX Namespace Reference

### Classes

- class [WFX\\_LightFlicker](#)  
*Controls the flickering of the light for the muzzle flash.*

### 5.7.1 Detailed Description

Rapidly sets a light on/off.

(c) 2015, Jean Moreno



## Chapter 6

# Class Documentation

### 6.1 PC.UI.AlphaPMS Class Reference

#### Public Attributes

- MarketModule **MarketModule** = null
- StashModule **StashModule** = new StashModule()
- TransportationModule **TransportationModule** = null
- MemoryModule **MemoryModule** = null

### 6.2 PC.UI.BetaPMS Class Reference

#### Additional Inherited Members

### 6.3 PC.UI.ButtonAction Struct Reference

#### Public Member Functions

- **ButtonAction** (string label, UnityEngine.Events.UnityAction action=null)

#### Public Attributes

- string **Label**
- UnityEngine.Events.UnityAction **Action**

### 6.4 PC.Combat.CharacterCombat Class Reference

Defined the combat behavior specific to the player character.

## Public Member Functions

- void [Attack](#) ([CharacterStats](#) targetStats)  
*Attack the target. This is called when the player's gun gets a hit on a damageable object.*

### 6.4.1 Member Function Documentation

#### 6.4.1.1 Attack()

```
void PC.Combat.CharacterCombat.Attack (
    CharacterStats targetStats )
```

##### Parameters

<i>targetStats</i>	The stats object for the target being hit. This controls the amount of damage it receives.
--------------------	--

## 6.5 PC.Stats.CharacterStats Class Reference

This class is used to store general stats and damage behavior common to all game characters.

### Public Member Functions

- void **TakeDamage** ()  
*The general damage function. This is called by other scripts any time the relevant entity needs to lose health points.*
- virtual void [Die](#) ()  
*Logs an entities death to the console. Specific death behavior and animations will be defined in scripts specific to a character.*

### 6.5.1 Member Function Documentation

#### 6.5.1.1 Die()

```
virtual void PC.Stats.CharacterStats.Die ( ) [virtual]
```

Reimplemented in [PC.Stats.EnemyStats](#), and [PC.Stats.PlayerStats](#).

## 6.6 PC.UI.Container Class Reference

### Protected Member Functions

- override void [Start](#) ()

## Private Attributes

- [Item](#) `_itemPrefab` = null
- [ItemSO](#)[] `_items` = null
- [Vector2Int](#)[] `_itemPositions` = null

## Additional Inherited Members

### 6.6.1 Member Function Documentation

#### 6.6.1.1 Start()

```
override void PC.UI.Container.Start ( ) [protected], [virtual]
```

Reimplemented from [PC.UI.ContainerBase](#).

## 6.7 PC.UI.ContainerBase Class Reference

### Public Member Functions

- void [OnPointerEnter](#) (PointerEventData eventData)  
*Is called upon a cursor entering the container's bounds.*
- void [OnPointerExit](#) (PointerEventData eventData)  
*Is called upon a cursor exiting the container's bounds.*
- void [SetItemParent](#) (RectTransform itemRectTransform)  
*Sets the parent of an item to the content parent object of this container.*
- virtual [Vector2Int](#) [GetCellIndex](#) (Vector2 mousePos)  
*Gets the cell index relative to the container using the given mouse position.*
- [Item](#) [GetItemAt](#) (Vector2Int cellIndex)  
*Gets the item at the given cell index.*
- virtual bool [RemoveItemAt](#) (Vector2Int cellIndex)  
*Removes the item at a given cellIndex from the container.*
- virtual bool [PlaceItemAt](#) ([Item](#) item, Vector2Int cellIndex)  
*Places an item at a given cellIndex in the container.*
- virtual [Item](#) [TakeItemAt](#) (Vector2Int cellIndex)  
*Takes an item from a container*
- void [TransferItem](#) (Vector2Int sourceCellIndex, [Container](#) targetContainer, Vector2Int targetCellIndex)  
*Transfers an item from one container to another.*

## Protected Member Functions

- virtual void **Awake** ()
- virtual void **Start** ()
- void **InitContents** ()
  - Initializes the contents of the container.*
- void **InitContents** (Vector2 size)
  - Initializes the contents of the container.*
- void **InitBackground** (Vector2 size)
  - Initializes the background of the container.*
- **Item** **GetCell** (Vector2Int cellIndex)
  - Check the contents of the container.*
- void **SetCell** (Vector2Int cellIndex, **Item** item)
  - Sets the contents of the container.*
- void **EmptyCell** (Vector2Int cellIndex)
  - Sets the cell at the given index to null.*
- bool **IsCellOutOfRange** (Vector2Int cellIndex)
- bool **IsCellEmpty** (Vector2Int cellIndex)
- bool **IsCellEmptyExcludingSelfOrCopy** (**Item** item, Vector2Int cellIndex)
- bool **IsCellOccupied** (Vector2Int cellIndex)
- bool **IsCellOccupiedExcludingSelfOrCopy** (**Item** item, Vector2Int cellIndex)
- bool **IsCellOccupiedBySelfOrCopy** (**Item** item, Vector2Int cellIndex)
- bool **IsCellOccupiedNotBySelfOrCopy** (**Item** item, Vector2Int cellIndex)

## Protected Attributes

- RectTransform **\_contentsParent** = null
- int **cellWidth** = 10
- int **cellHeight** = 10
- RectTransform **\_rectTransform** = null
- RectTransform **\_anchor** = null
- RectTransform **\_cellBackgroundPrefab** = null
- RectTransform **\_cellBackgroundParent** = null
- Vector2 **\_mousePos** = Vector2.zero
- Vector2 **\_currentPosition** = Vector2.zero
- Vector2Int **\_currentCellIndex** = Vector2Int.zero

## Properties

- RectTransform **ContentsParent** [get]
- **Item**[,] **\_contents** [get, set]

## Private Attributes

- **Item**[,] **h\_contents** = null

## 6.7.1 Member Function Documentation

### 6.7.1.1 EmptyCell()

```
void PC.UI.ContainerBase.EmptyCell (
    Vector2Int cellIndex ) [protected]
```



**Parameters**

<i>cellIndex</i>	The cell index you want to set to null.
------------------	---

**6.7.1.2 GetCell()**

```
Item PC.UI.ContainerBase.GetCell (
    Vector2Int cellIndex ) [protected]
```

**Parameters**

<i>cellIndex</i>	The cell index you want to check.
------------------	-----------------------------------

**Returns**

The item at the given cell index if an item was present or null if there was no item.

**6.7.1.3 GetCellIndex()**

```
virtual Vector2Int PC.UI.ContainerBase.GetCellIndex (
    Vector2 mousePos ) [virtual]
```

**Parameters**

<i>mousePos</i>	Position of the mouse.
-----------------	------------------------

**Returns**

Cell index relative to the container.

Reimplemented in [PC.UI.Slot](#).

**6.7.1.4 GetItemAt()**

```
Item PC.UI.ContainerBase.GetItemAt (
    Vector2Int cellIndex )
```

**Parameters**

<i>cellIndex</i>	The cell index you want to check.
------------------	-----------------------------------

**Returns**

The item at the given cell index if an item was present or null if there was no item.

**6.7.1.5 InitBackground()**

```
void PC.UI.ContainerBase.InitBackground (
    Vector2 size ) [protected]
```

**Parameters**

<i>size</i>	The area of the background.
-------------	-----------------------------

**6.7.1.6 InitContents()**

```
void PC.UI.ContainerBase.InitContents (
    Vector2 size ) [protected]
```

**Parameters**

<i>size</i>	The area of the contents.
-------------	---------------------------

**6.7.1.7 OnPointerEnter()**

```
void PC.UI.ContainerBase.OnPointerEnter (
    PointerEventData eventData )
```

**Parameters**

<i>eventData</i>	The event data of the pointer.
------------------	--------------------------------

**6.7.1.8 OnPointerExit()**

```
void PC.UI.ContainerBase.OnPointerExit (
    PointerEventData eventData )
```

**Parameters**

<i>eventData</i>	The event data of the pointer.
------------------	--------------------------------

### 6.7.1.9 PlaceItemAt()

```
virtual bool PC.UI.ContainerBase.PlaceItemAt (
    Item item,
    Vector2Int cellIndex ) [virtual]
```

#### Parameters

<i>item</i>	The item you want to place.
<i>cellIndex</i>	The cell index you want to place the item at.

#### Returns

Returns true if the operation was successful, otherwise false.

Reimplemented in [PC.UI.Slot](#).

### 6.7.1.10 RemoveItemAt()

```
virtual bool PC.UI.ContainerBase.RemoveItemAt (
    Vector2Int cellIndex ) [virtual]
```

#### Parameters

<i>cellIndex</i>	The cell index of the item you want to remove.
------------------	--

#### Returns

Returns true if the operation was successful, otherwise false.

Reimplemented in [PC.UI.Slot](#).

### 6.7.1.11 SetCell()

```
void PC.UI.ContainerBase.SetCell (
    Vector2Int cellIndex,
    Item item ) [protected]
```

#### Parameters

<i>cellIndex</i>	The cell index you want to store an item at.
<i>item</i>	The item you want to store.

### 6.7.1.12 SetItemParent()

```
void PC.UI.ContainerBase.SetItemParent (
    RectTransform itemRectTransform )
```

#### Parameters

<i>itemRectTransform</i>	The given item you want to set the parent of.
--------------------------	---

### 6.7.1.13 TakeItemAt()

```
virtual Item PC.UI.ContainerBase.TakeItemAt (
    Vector2Int cellIndex ) [virtual]
```

#### Parameters

<i>cellIndex</i>	The cell index of the item you want to take from the container.
------------------	---

#### Returns

The item taken from the container

### 6.7.1.14 TransferItem()

```
void PC.UI.ContainerBase.TransferItem (
    Vector2Int sourceCellIndex,
    Container targetContainer,
    Vector2Int targetCellIndex )
```

#### Parameters

<i>sourceCellIndex</i>	The cell index of the item you want to transfer.
<i>targetContainer</i>	The container you want to transfer the item to.
<i>targetCellIndex</i>	The cell index you want to transfer the item to.

## 6.8 PC.UI.DevConsoleMenu Class Reference

### Protected Member Functions

- override void [AwakeExtension](#) ()

DO NOT HIDE/OVERRIDE [MenuBase.Awake\(\)](#) USING THE NEW OR OVERRIDE KEYWORDS!!! [AwakeExtension\(\)](#) is called at the end of [MenuBase.Awake\(\)](#). If you need any code to run in [MonoBehaviour.Awake\(\)](#) put it inside [AwakeExtension\(\)](#).

- override void [OpenExtension](#) ()  
Gets called at the end of [Open\(\)](#).
- override void [CloseExtension](#) ()  
Gets called at the end of [Close\(\)](#).

## Private Member Functions

- void [Update](#) ()

## Private Attributes

- [RectTransform\\_content](#) = null
- [RectTransform\\_logItemPrefab](#) = null

## Additional Inherited Members

### 6.8.1 Member Function Documentation

#### 6.8.1.1 AwakeExtension()

```
override void PC.UI.DevConsoleMenu.AwakeExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

#### 6.8.1.2 CloseExtension()

```
override void PC.UI.DevConsoleMenu.CloseExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

#### 6.8.1.3 OpenExtension()

```
override void PC.UI.DevConsoleMenu.OpenExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

## 6.9 PC.Combat.EnemyAnimationController Class Reference

Controls how the enemy animations react to information in the environment.

### Public Member Functions

- void **Die** ()

*Plays the enemy death animation when the animator has recorded the enemy as dead. The animator gets this information from the enemy controller.*

## 6.10 PC.Combat.EnemyCombat Class Reference

Defines the combat behavior specific to our enemies.

### Public Member Functions

- void **AttackTarget** (Transform target)

*Activates the enemy's attack when the player is touched.*

- void **SetTags** (Transform curr)

*Sets the tag of all child objects to "Enemy" so that collisions with appendages can be identified. Calls this recursively on all children.*

### Public Attributes

- bool **\_touchingPlayer** = false

*Records whether or not the enemy and player colliders overlap.*

### 6.10.1 Member Function Documentation

#### 6.10.1.1 AttackTarget()

```
void PC.Combat.EnemyCombat.AttackTarget (  
    Transform target )
```

#### Parameters

<i>target</i>
---------------

### 6.10.1.2 SetTags()

```
void PC.Combat.EnemyCombat.SetTags (
    Transform curr )
```

#### Parameters

<i>curr</i>	The transform of the object currently being expected.
-------------	---

## 6.11 PC.Entities.EnemyController Class Reference

[EnemyController](#) is a class that handles the enemy's AI.

The enemy is programmed with a navmeshagent to follow and attack the player when the player is in a specific range.

### Private Member Functions

- void **FaceTarget** ()

*FaceTarget is a method that rotates the enemy towards the player during a chase.*

## 6.12 PC.Stats.EnemyStats Class Reference

This class is used to store the enemy's stats and log its death.

### Public Member Functions

- override void [Die](#) ()

*Logs the enemy's death and can specify other death behavior not related to animations which are handled in the animation controller.*

### 6.12.1 Member Function Documentation

#### 6.12.1.1 Die()

```
override void PC.Stats.EnemyStats.Die ( ) [virtual]
```

Reimplemented from [PC.Stats.CharacterStats](#).

## 6.13 PC.Entities.Gun Class Reference

Defines the general behavior of the player's gun.

## Private Member Functions

- async void [Reload](#) (InputAction.CallbackContext obj)  
*Reloads the gun's magazine*
- async void [Shoot](#) (InputAction.CallbackContext obj)  
*Deals with all functionality related shooting. Includes audio, muzzle flash, recoil, raycast shooting and target damage.*

### 6.13.1 Member Function Documentation

#### 6.13.1.1 Reload()

```
async void PC.Entities.Gun.Reload (
    InputAction.CallbackContext obj ) [private]
```

##### Parameters

<i>obj</i>	Context for the InputActions asset. Passed automatically after Reload is subscribed to the relevant input.
------------	--

#### 6.13.1.2 Shoot()

```
async void PC.Entities.Gun.Shoot (
    InputAction.CallbackContext obj ) [private]
```

##### Parameters

<i>obj</i>	Context for the InputActions asset. Passed automatically after Shoot is subscribed to the relevant input.
------------	---

## 6.14 PC.Entities.GunSO Class Reference

This class defines various parameters that control the gun's functionality and behavior. For instance, the gun's damage, range, and fire rate default values are defined here.

### Public Attributes

- float **Range** = 100f  
*The range within the gun can deal damage.*
- float **Damage** = 20f  
*The damage the gun deals to the target.*
- int **MagazineSize** = 30  
*The number of rounds in the gun's magazine.*



- float **FireRate** = 10f  
*Max fire rate of the gun.*
- float **ReloadTime** = 3.0f  
*Time to reload.*
- float **VerticalRecoil** = 2f  
*Vertical recoil component magnitude.*
- float **HorizontalRecoil** = 0.5f  
*Horizontal recoil component magnitude.*

## 6.15 PC.UI.HorizontalLayoutGroupHandler Class Reference

### Private Member Functions

- void **Awake** ()
- void **Update** ()

### Private Attributes

- RectTransform **\_rectTransform**
- List< RectTransform > **\_children** = new List<RectTransform>()

## 6.16 PC.Input.InputModule Class Reference

### Public Member Functions

- override void **Process** ()

### Protected Member Functions

- override void **Awake** ()

### Properties

- static [InputModule](#) **instance** [get]
- static InputActions **InputActions** [get]

### Static Private Attributes

- static InputActions **h\_inputActions**
- static bool **\_awakeCalledFlag** = false
- static [InputModule](#) **\_instance** = null

## 6.17 PC.UI.Interactable Class Reference

## 6.18 PC.UI.InventoryMenu Class Reference

### Public Member Functions

- void **OpenWithStash** ()  
*Opens the inventory with the stash on the right. So you can transfer items between the stash and the inventory.*

### Protected Member Functions

- override void **AwakeExtension** ()  
*DO NOT HIDE/OVERRIDE [MenuBase.Awake\(\)](#) USING THE NEW OR OVERRIDE KEYWORDS!!! [AwakeExtension\(\)](#) is called at the end of [MenuBase.Awake\(\)](#). If you need any code to run in `MonoBehaviour.Awake()` put it inside [AwakeExtension\(\)](#).*
- override void **OpenExtension** ()  
*Gets called at the end of [Open\(\)](#).*
- override void **CloseExtension** ()  
*Gets called at the end of [Close\(\)](#).*

### Properties

- static [ContainerBase](#) **CurrentContainer** [set]
- static [InventoryMenu](#) **instance** [get, set]
- [ContainerBase](#) **\_currentContainer** [get, set]
- [Item](#) **\_currentItemSource** [get, set]

### Private Member Functions

- void **Update** ()
- void **ClampCurrentItemToCursor** ()  
*Clamps the current item to the cursor position.*
- void **TryPickingUpItem** (Vector2Int cellIndex)  
*Tries to pick up an item from the current container.*
- void **TryReleasingItem** (Vector2Int cellIndex)  
*Tries to release the current item to the current container.*
- void **TryRotatingItem** ()  
*Tries to rotate the current item.*

### Private Attributes

- [ContainerBase](#) **h\_currentContainer** = null
- [Item](#) **h\_currentItemSource** = null
- [Item](#) **\_currentItemCopy** = null
- `GameObject` **\_stash** = null

## Static Private Attributes

- static List< string > **\_hierarchyPaths** = new List<string>()
- static [InventoryMenu](#) **h\_instance** = null

## 6.18.1 Member Function Documentation

### 6.18.1.1 AwakeExtension()

```
override void PC.UI.InventoryMenu.AwakeExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

### 6.18.1.2 CloseExtension()

```
override void PC.UI.InventoryMenu.CloseExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

### 6.18.1.3 OpenExtension()

```
override void PC.UI.InventoryMenu.OpenExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

### 6.18.1.4 TryPickingUpItem()

```
void PC.UI.InventoryMenu.TryPickingUpItem (
    Vector2Int cellIndex ) [private]
```

#### Parameters

<i>cellIndex</i>	The cell index of the item to pick up.
------------------	--

### 6.18.1.5 TryReleasingItem()

```
void PC.UI.InventoryMenu.TryReleasingItem (
```

```
Vector2Int cellIndex ) [private]
```

#### Parameters

<code>cellIndex</code>	The cell index to release the item to.
------------------------	--

## 6.19 PC.UI.Item Class Reference

### Public Member Functions

- [Item Init](#) ([ItemSO](#) itemSO)  
*Initializes the item.*
- void **SetContainer** ([ContainerBase](#) container, Vector2Int originCellIndex)  
*Caches the container the item is currently in. Sets the item's parent to the given container's content GameObject. Sets the item's position to the given cell index relative to the container's content GameObject.*
- void **RemoveContainer** ()  
*Removes the item's container reference and sets its parent to null.*
- Vector2Int [GetOriginCellIndex](#) ()  
*Gets the origin cell index of the item, relative to its container's content 2D array.*
- void **Rotate** ()  
*Rotates the item or rather toggles the item's rotation between 0 and 90 degrees Useful for packing items together more efficiently*
- [Item MakeCopy](#) ()  
*Duplicates the item as another item.*
- bool **TransferTo** ([ContainerBase](#) container, Vector2Int cellIndex)  
*Transfers the item to a given container at a given cell index.*
- void **Destroy** ()  
*Destroys the item.*

### Public Attributes

- bool **isRotated** = false
- uint **cellWidth**
- uint **cellHeight**
- RectTransform **RectTransform** = null
- [ItemType](#) **type**

### Properties

- Vector2Int **OriginCellIndex** [get]
- [Item Copy](#) [get]
- [Item Source](#) [get]

### Private Member Functions

- void **Awake** ()
- void **SetSize** ()  
*Sets the RectTransform size of the item based on the cellWidth and cellHeight*
- void **SetImages** ()  
*Sets the item's images based on the item's [ItemSO](#).*

## Private Attributes

- `ItemSO _itemSO` = null
- `ContainerBase _currentContainer` = null
- `Image _backgroundImage`
- `Image _contentImage`
- `TMP_Text _nicknameLabel`
- `RectTransform _contentRectTransform` = null
- `Vector2Int _originCellIndex` = `Vector2Int.zero`
- `Item _copy` = null
- `Item _source` = null

## 6.19.1 Member Function Documentation

### 6.19.1.1 GetOriginCellIndex()

```
Vector2Int PC.UI.Item.GetOriginCellIndex ( )
```

#### Returns

The origin cell index of the item.

### 6.19.1.2 Init()

```
Item PC.UI.Item.Init (
    ItemSO itemSO )
```

#### Parameters

<code>itemSO</code>	The itemSO to set the item's details to.
---------------------	--

#### Returns

The item.

### 6.19.1.3 MakeCopy()

```
Item PC.UI.Item.MakeCopy ( )
```

#### Returns

The duplicate item.

### 6.19.1.4 TransferTo()

```
bool PC.UI.Item.TransferTo (
    ContainerBase container,
    Vector2Int cellIndex )
```

#### Parameters

<i>container</i>	The container to transfer the item to.
<i>cellIndex</i>	The cell index to transfer the item to.

#### Returns

True if the transfer was successful, false otherwise.

## 6.20 PC.UI.ItemContainerInfo Class Reference

### Static Public Member Functions

- static [ItemContainerInfo Create](#) ([Container](#) container, [Vector2Int](#) index)  
*Creates an [ItemContainerInfo](#) object containing info about a given item's state in its source container*

### Properties

- [Item](#) **item** = null [get, private set]
- [Vector2Int](#) **sourceIndex** = [Vector2Int.zero](#) [get, private set]
- [Container](#) **sourceContainer** = null [get, private set]
- [RectTransform](#) **rectTransform** = null [get, private set]
- bool **wasRotated** = false [get, private set]

### Private Member Functions

- [ItemContainerInfo](#) ([Container](#) currentContainer, [Vector2Int](#) currentIndex)  
*Constructor for the [ItemContainerInfo](#) class*

### 6.20.1 Constructor & Destructor Documentation

#### 6.20.1.1 ItemContainerInfo()

```
PC.UI.ItemContainerInfo.ItemContainerInfo (
    Container currentContainer,
    Vector2Int currentIndex ) [private]
```

## Parameters

<i>container</i>	The source container the item is currently in
<i>index</i>	The source index of the item in the source container

## Returns

An [ItemContainerInfo](#) object containing info about the item's state in its source container

## 6.20.2 Member Function Documentation

### 6.20.2.1 Create()

```
static ItemContainerInfo PC.UI.ItemContainerInfo.Create (
    Container container,
    Vector2Int index ) [static]
```

## Parameters

<i>container</i>	The source container the item is currently in
<i>index</i>	The source index of the item in the source container

## Returns

An [ItemContainerInfo](#) object containing info about the item's state in its source container

## 6.21 PC.UI.ItemSO Class Reference

### Public Attributes

- uint **cellWidth** = 1
- uint **cellHeight** = 1
- Color **backgroundColor** = Color.white
- Sprite **itemIcon** = null
- new string **name**
- string **nickname**
- string **description**
- float **mass**
- uint **value**
- [ItemType.Value](#) type

## 6.22 PC.UI.ItemType Class Reference

### Public Types

- enum [Value](#) : System.UInt32  
*The type of an item.*

## Public Member Functions

- bool `belongsTo` (`ItemType` item)  
*Checks if a type belongs to another type.*
- override bool `Equals` (object a)  
*Checks if an enum Type is equal to an instance of an `ItemType`.*

## Static Public Member Functions

- static implicit `operator uint` (`ItemType` item)  
*Implicitly converts an `ItemType` to a `uint`.*
- static bool `operator==` (`ItemType` a, `ItemType` b)  
*Checks if an instance of an `ItemType` is equal to another instance of an `ItemType`.*
- static bool `operator==` (`ItemType` a, `Value` b)  
*Checks if an instance of an `ItemType` is equal to an enum Type.*
- static bool `operator!=` (`ItemType` a, `ItemType` b)  
*Checks if two instances of `ItemType` are equal.*
- static bool `operator!=` (`ItemType` a, `Value` b)  
*Checks if a enum Type is not equal to a instance of an `ItemType`.*
- static `ItemType Lookup` (`Value` value)  
*Returns the type of an item.*

## Properties

- `uint data` [get, set]
- `string dataAsHex` [get]
- `byte layer1` [get]
- `byte layer2` [get]
- `byte layer3` [get]
- `byte layer4` [get]
- `int lastLayerIndex` [get]

## Private Member Functions

- `ItemType` (`Value` val)

## Static Private Member Functions

- static `Dictionary< Value, ItemType > InitLookup` ()

## Private Attributes

- `uint h_data` = 0x00000000
- `byte[] layers` = new byte[4]



## Static Private Attributes

- const uint **LAYER\_1\_MASK** = 0xFF000000
- const uint **LAYER\_2\_MASK** = 0x00FF0000
- const uint **LAYER\_3\_MASK** = 0x0000FF00
- const uint **LAYER\_4\_MASK** = 0x000000FF
- static readonly Dictionary< Value, ItemType > **dict** = InitLookup()

## 6.22.1 Member Function Documentation

### 6.22.1.1 belongsTo()

```
bool PC.UI.ItemType.belongsTo (
    ItemType item )
```

#### Parameters

<i>item</i>	The type to check.
-------------	--------------------

#### Returns

True if the type belongs to the other type, else false.

### 6.22.1.2 Equals()

```
override bool PC.UI.ItemType.Equals (
    object a )
```

#### Parameters

<i>a</i>	The value of the item.
----------	------------------------

### 6.22.1.3 Lookup()

```
static ItemType PC.UI.ItemType.Lookup (
    Value value ) [static]
```

#### Parameters

<i>value</i>	The value of the item.
--------------	------------------------

**Returns**

The type of the item.

**6.22.1.4 operator"!=(()) [1/2]**

```
static bool PC.UI.ItemType.operator!=(
    ItemType a,
    ItemType b ) [static]
```

**Parameters**

<i>a</i>	The first given instance of the <a href="#">ItemType</a> .
<i>b</i>	The second given instance of the <a href="#">ItemType</a> .

**6.22.1.5 operator"!=(()) [2/2]**

```
static bool PC.UI.ItemType.operator!=(
    ItemType a,
    Value b ) [static]
```

**Parameters**

<i>a</i>	The given enum type.
<i>b</i>	The given instance of the <a href="#">ItemType</a> .

**6.22.1.6 operator==(()) [1/2]**

```
static bool PC.UI.ItemType.operator==(
    ItemType a,
    ItemType b ) [static]
```

**Parameters**

<i>a</i>	The first given instance of the <a href="#">ItemType</a> .
<i>b</i>	The second given instance of the <a href="#">ItemType</a> .

**6.22.1.7 operator==(()) [2/2]**

```
static bool PC.UI.ItemType.operator==(
```

```
ItemType a,  
Value b ) [static]
```

#### Parameters

<i>a</i>	The given instance of the <a href="#">ItemType</a> .
<i>b</i>	The given enum type.

## 6.23 PC.UI.ItemTypes Class Reference

### Public Member Functions

- bool [belongsTo](#) (ItemType item)  
*Checks if a type belongs to a list of types.*

### Private Attributes

- List< [ItemType](#) > **types** = new List<ItemType>()

### 6.23.1 Member Function Documentation

#### 6.23.1.1 belongsTo()

```
bool PC.UI.ItemTypes.belongsTo (  
    ItemType item )
```

#### Parameters

<i>item</i>	The type to check.
-------------	--------------------

#### Returns

True if the type belongs to the list of types, else false.

## 6.24 PC.UI.LayoutMaintainPreferredHeight Class Reference

### Private Member Functions

- void **Awake** ()
- void **Update** ()

## Private Attributes

- RectTransform `_rectTransform`
- HorizontalOrVerticalLayoutGroup `_layoutGroup`
- float `_preferredHeight` = 0f

## 6.25 PC.UI.MapMenu Class Reference

### Protected Member Functions

- override void [AwakeExtension](#) ()  
*DO NOT HIDE/OVERRIDE [MenuBase.Awake\(\)](#) USING THE NEW OR OVERRIDE KEYWORDS!!! [AwakeExtension\(\)](#) is called at the end of [MenuBase.Awake\(\)](#). If you need any code to run in [MonoBehaviour.Awake\(\)](#) put it inside [AwakeExtension\(\)](#).*
- override void [OpenExtension](#) ()  
*Gets called at the end of [Open\(\)](#).*
- override void [CloseExtension](#) ()  
*Gets called at the end of [Close\(\)](#).*

### Additional Inherited Members

#### 6.25.1 Member Function Documentation

##### 6.25.1.1 AwakeExtension()

```
override void PC.UI.MapMenu.AwakeExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

##### 6.25.1.2 CloseExtension()

```
override void PC.UI.MapMenu.CloseExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

##### 6.25.1.3 OpenExtension()

```
override void PC.UI.MapMenu.OpenExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

## 6.26 PC.UI.MenuBase Class Reference

### Public Member Functions

- void **Open** ()  
*DO NOT OVERRIDE/HIDE [MenuBase.Open\(\)](#) USING THE NEW OR OVERRIDE KEYWORDS!!! Override the [OpenExtension\(\)](#) function instead. Opens the menu. Closes all other menus. Enables base menu input while disabling all others. Unlocks the cursor. Calls [OpenExtension\(\)](#) at end end of this function. [OpenExtension\(\)](#) by default does nothing. Needs to be overridden if you want to add to the functionality of [Open\(\)](#).*
- void **Close** ()  
*DO NOT OVERRIDE/HIDE [MenuBase.Close\(\)](#) USING THE NEW OR OVERRIDE KEYWORDS!!! Override the [CloseExtension\(\)](#) function instead. Closes the menu. Locks the cursor. Calls [CloseExtension\(\)](#) at end end of this function. [CloseExtension\(\)](#) by default does nothing. Needs to be overridden if you want to add to the functionality of [Close\(\)](#).*

### Protected Member Functions

- void **Awake** ()  
*DO NOT HIDE/OVERRIDE THIS FUNCTION USING THE NEW OR OVERRIDE KEYWORDS!!! Override the [AwakeExtension\(\)](#) function instead.*
- abstract void [AwakeExtension](#) ()  
*DO NOT HIDE/OVERRIDE [MenuBase.Awake\(\)](#) USING THE NEW OR OVERRIDE KEYWORDS!!! [AwakeExtension\(\)](#) is called at the end of [MenuBase.Awake\(\)](#). If you need any code to run in [MonoBehaviour.Awake\(\)](#) put it inside [AwakeExtension\(\)](#).*
- abstract void [OpenExtension](#) ()  
*Gets called at the end of [Open\(\)](#).*
- abstract void [CloseExtension](#) ()  
*Gets called at the end of [Close\(\)](#).*

### Properties

- static InputActions **\_inputActions** [get]

### Private Member Functions

- void **Disable** ()  
*Hides the menu from the scene.*

### Static Private Member Functions

- static void **DisableAll** ()  
*Hides ALL menus from the scene.*

### Static Private Attributes

- static readonly List< [MenuBase](#) > **menus** = new List<[MenuBase](#)>()

## 6.26.1 Member Function Documentation

### 6.26.1.1 AwakeExtension()

```
abstract void PC.UI.MenuBase.AwakeExtension ( ) [protected], [pure virtual]
```

Implemented in [PC.UI.DevConsoleMenu](#), [PC.UI.InventoryMenu](#), [PC.UI.MapMenu](#), [PC.UI.PauseMenu](#), [PC.UI.PlayerManagementState](#) and [PC.UI.PMSMenu](#).

### 6.26.1.2 CloseExtension()

```
abstract void PC.UI.MenuBase.CloseExtension ( ) [protected], [pure virtual]
```

Implemented in [PC.UI.DevConsoleMenu](#), [PC.UI.InventoryMenu](#), [PC.UI.MapMenu](#), [PC.UI.PauseMenu](#), [PC.UI.PlayerManagementState](#) and [PC.UI.PMSMenu](#).

### 6.26.1.3 OpenExtension()

```
abstract void PC.UI.MenuBase.OpenExtension ( ) [protected], [pure virtual]
```

Implemented in [PC.UI.DevConsoleMenu](#), [PC.UI.InventoryMenu](#), [PC.UI.MapMenu](#), [PC.UI.PauseMenu](#), [PC.UI.PlayerManagementState](#) and [PC.UI.PMSMenu](#).

## 6.27 PC.UI.MenusController Class Reference

### Properties

- static [PauseMenu](#) **PauseMenu** [get]
- static [InventoryMenu](#) **InventoryMenu** [get]
- static [DevConsoleMenu](#) **DevConsoleMenu** [get]
- static [MapMenu](#) **MapMenu** [get]
- static [PMSMenu](#) **PMSMenu** [get]

### Private Member Functions

- void **Awake** ()

## Private Attributes

- [PauseMenu](#) `_pauseMenu` = null
- [InventoryMenu](#) `_inventoryMenu` = null
- [DevConsoleMenu](#) `_devConsoleMenu` = null
- [MapMenu](#) `_mapMenu` = null
- [PMSMenu](#) `_pmsMenu` = null

## Static Private Attributes

- static [MenusController](#) `_instance` = null

## 6.28 PC.UI.PauseMenu Class Reference

### Public Member Functions

- void [ExitGame](#) ()

### Static Public Attributes

- static bool `activated` = false

### Protected Member Functions

- override void [AwakeExtension](#) ()  
*DO NOT HIDE/OVERRIDE [MenuBase.Awake\(\)](#) USING THE NEW OR OVERRIDE KEYWORDS!!! [AwakeExtension\(\)](#) is called at the end of [MenuBase.Awake\(\)](#). If you need any code to run in [MonoBehaviour.Awake\(\)](#) put it inside [AwakeExtension\(\)](#).*
- override void [OpenExtension](#) ()  
*Gets called at the end of [Open\(\)](#).*
- override void [CloseExtension](#) ()  
*Gets called at the end of [Close\(\)](#).*

## Private Attributes

- Button `_closePauseMenuButton` = null
- Button `_exitGameButton` = null

## Additional Inherited Members

### 6.28.1 Member Function Documentation

### 6.28.1.1 AwakeExtension()

```
override void PC.UI.PauseMenu.AwakeExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

### 6.28.1.2 CloseExtension()

```
override void PC.UI.PauseMenu.CloseExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

### 6.28.1.3 OpenExtension()

```
override void PC.UI.PauseMenu.OpenExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

## 6.29 PC.Entities.PlayerAnimatorController Class Reference

This class is responsible for controlling the player's animations. For our purposes, it simply regularly updates the variables used by the animator to determine which animation to play.

### Properties

- static InputActions **inputActions** [get]

### Private Member Functions

- void **Start** ()
- void **Update** ()

### Private Attributes

- Animator **\_animator** = null

### Static Private Attributes

- const int **UNARMED** = 0
- const int **ASSAULT\_RIFLE** = 1
- const int **BEAM\_GUN** = 2
- const int **GATLIN\_GUN** = 3
- const int **MISSILE\_LAUNCHER** = 4
- const int **IDLE** = 0
- const int **WALKING** = 1
- const int **RUNNING** = 2



## 6.30 PC.Combat.PlayerCombat Class Reference

Defines the combat behavior specific to our player.

### Public Member Functions

- void [AttackTarget](#) (RaycastHit hit)

*Finds the stats component of the target object and uses it to apply damage. This is called from Gun.cs*

### 6.30.1 Member Function Documentation

#### 6.30.1.1 AttackTarget()

```
void PC.Combat.PlayerCombat.AttackTarget (
    RaycastHit hit )
```

#### Parameters

<i>hit</i>	The target object his by the gun's raycast.
------------	---

## 6.31 PC.Entities.PlayerController Class Reference

Defines functionality and attributes specific to our player character.

### Protected Member Functions

- override void **Look** ()  
*Turns the Player object along x mouse input and turns camera along y mouse input.*
- override void **Move** ()  
*Moves the Player object by input from user and also applies gravity.*
- override void **SetupInput** ()  
*Initializes the InputActions object, which handles all input from the user.*
- override void **CheckForInteractions** ()  
*Displays interaction message if player is close enough to a terminal*
- override void **Interact** ()  
*Triggers menu when interacting with player management stations*
- void [OnAnimatorIK](#) (int layerIndex)  
*Modifies player animations to adjust hand position on weapon via inverse kinematics*

## Private Member Functions

- void **EquipPrimaryFirearm** ()  
*Equips the player's primary firearm from their inventory*
- void **EquipSecondaryFirearm** ()  
*Equips the player's secondary firearm from their inventory*

### 6.31.1 Member Function Documentation

#### 6.31.1.1 OnAnimatorIK()

```
void PC.Entities.PlayerController.OnAnimatorIK (
    int layerIndex ) [protected]
```

##### Parameters

<i>layerIndex</i>	Index of layer on which the IK solver is called.
-------------------	--

## 6.32 PC.Entities.PlayerControllerBase Class Reference

Defined functionality and attributes common to all player controllers such as ground checking and translating inputs to movement information.

## Private Member Functions

- Vector3 **GetMovement** ()  
*Retrieves the player's current movement vector*
- bool **GroundedCheck** ()  
*Checks if the player is grounded*

### 6.32.1 Member Function Documentation

#### 6.32.1.1 GetMovement()

```
Vector3 PC.Entities.PlayerControllerBase.GetMovement ( ) [private]
```

##### Returns

Movement vector already adjusted for speed

### 6.32.1.2 GroundedCheck()

```
bool PC.Entities.PlayerControllerBase.GroundedCheck ( ) [private]
```

#### Returns

True if the player is touching the ground, otherwise false.

## 6.33 PC.UI.PlayerManagementStationMenu Class Reference

### Protected Member Functions

- override void [AwakeExtension](#) ()  
*DO NOT HIDE/OVERRIDE [MenuBase.Awake\(\)](#) USING THE NEW OR OVERRIDE KEYWORDS!!! [AwakeExtension\(\)](#) is called at the end of [MenuBase.Awake\(\)](#). If you need any code to run in [MonoBehaviour.Awake\(\)](#) put it inside [AwakeExtension\(\)](#).*
- override void [OpenExtension](#) ()  
*Gets called at the end of [Open\(\)](#).*
- override void [CloseExtension](#) ()  
*Gets called at the end of [Close\(\)](#).*

### Additional Inherited Members

#### 6.33.1 Member Function Documentation

##### 6.33.1.1 AwakeExtension()

```
override void PC.UI.PlayerManagementStationMenu.AwakeExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

##### 6.33.1.2 CloseExtension()

```
override void PC.UI.PlayerManagementStationMenu.CloseExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

##### 6.33.1.3 OpenExtension()

```
override void PC.UI.PlayerManagementStationMenu.OpenExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

## 6.34 PC.Entities.PlayerManager Class Reference

Creates a global instance of the player for easy reference by other scripts. Handles interface between player health and [UI](#).

### Public Attributes

- GameObject **player** = null  
*The global instance of the player.*
- GameObject **healthCount** = null  
*The player's current health.*

### Static Public Attributes

- static [PlayerManager](#) **instance**

## 6.35 PC.Stats.PlayerStats Class Reference

This class is used to store the player's stats and log its death to the console.

### Public Member Functions

- override void [Die](#) ()  
*Logs the enemy's death and can specify other death behavior not related to animations which are handled in the animation controller.*

### 6.35.1 Member Function Documentation

#### 6.35.1.1 [Die\(\)](#)

```
override void PC.Stats.PlayerStats.Die ( ) [virtual]
```

Reimplemented from [PC.Stats.CharacterStats](#).

## 6.36 PC.UI.PMSBase Class Reference

### Public Attributes

- QuestsModule **QuestsModule** = null
- CraftingModule **CraftingModule** = null
- SuitModule **SuitModule** = null
- WeaponUpgradingModule **WeaponUpgradingModule** = null

## 6.37 PC.UI.PMSMenu Class Reference

### Public Member Functions

- void [Init \(BetaPMS PMS\)](#)  
*Initializes the menu with the given betaPMS.*
- void [Init \(AlphaPMS PMS\)](#)  
*Initializes the menu with the given alphaPMS.*

### Protected Member Functions

- override void [AwakeExtension \(\)](#)  
*DO NOT HIDE/OVERRIDE [MenuBase.Awake\(\)](#) USING THE NEW OR OVERRIDE KEYWORDS!!! [AwakeExtension\(\)](#) is called at the end of [MenuBase.Awake\(\)](#). If you need any code to run in [MonoBehaviour.Awake\(\)](#) put it inside [AwakeExtension\(\)](#).*
- override void [OpenExtension \(\)](#)  
*Gets called at the end of [Open\(\)](#).*
- override void [CloseExtension \(\)](#)  
*Gets called at the end of [Close\(\)](#).*

### Private Attributes

- `TMP_Text _label = null`
- `Button _questsModuleButton = null`
- `Button _marketModuleButton = null`
- `Button _stashModuleButton = null`
- `Button _craftingModuleButton = null`
- `Button _suitModuleButton = null`
- `Button _weaponUpgradingModuleButton = null`
- `Button _transportationModuleButton = null`
- `Button _memoryModuleButton = null`

### Additional Inherited Members

#### 6.37.1 Member Function Documentation

##### 6.37.1.1 AwakeExtension()

override void `PC.UI.PMSMenu.AwakeExtension ( )` [protected], [virtual]

Implements [PC.UI.MenuBase](#).

### 6.37.1.2 CloseExtension()

```
override void PC.UI.PMSMenu.CloseExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

### 6.37.1.3 Init() [1/2]

```
void PC.UI.PMSMenu.Init (
    AlphaPMS PMS )
```

## Parameters

<i>PMS</i>	The alphaPMS to initialize the menu with.
------------	---

**6.37.1.4 Init()** [2/2]

```
void PC.UI.PMSMenu.Init (
    BetaPMS PMS )
```

## Parameters

<i>PMS</i>	The betaPMS to initialize the menu with.
------------	--

**6.37.1.5 OpenExtension()**

```
override void PC.UI.PMSMenu.OpenExtension ( ) [protected], [virtual]
```

Implements [PC.UI.MenuBase](#).

## 6.38 PC.UI.PopupWindow Class Reference

### Static Public Member Functions

- static GameObject [Create](#) (GameObject parent, string title, string message, [ButtonAction](#) buttonAction, params [ButtonAction](#)[] additionalButtonActions)  
*Create a popup window.*
- static GameObject [Create](#) (GameObject parent, float width, float height, string title, string message, [ButtonAction](#) buttonAction, params [ButtonAction](#)[] additionalButtonActions)  
*Create a popup window.*

### Private Member Functions

- void [AddButton](#) ([ButtonAction](#) buttonAction)  
*Add a button to this popup window.*
- void [Init](#) (float width, float height, string title, string message, [ButtonAction](#) buttonAction, params [ButtonAction](#)[] additionalButtonActions)  
*Initialize the popup window.*
- void [Close](#) ()  
*Close the popup window.*

## Private Attributes

- `TMP_Text _title`
- `TMP_Text _message`
- `Transform _buttonPrefab`
- `Transform _buttonContent`
- `readonly List< Button > _buttons = new List<Button>()`

## Static Private Attributes

- `const string _popupWindowPrefabPath = "Prefabs/UI/PR_PopupWindow"`
- `const float _defaultPopupWindowWidth = 300f`
- `const float _defaultPopupWindowHeight = 200f`
- `const float _minPopupWindowWidth = 100f`
- `const float _minPopupWindowHeight = 75f`

## 6.38.1 Member Function Documentation

### 6.38.1.1 AddButton()

```
void PC.UI.PopupWindow.AddButton (
    ButtonAction buttonAction ) [private]
```

#### Parameters

<i>buttonAction</i>	The label and action to perform when the button is clicked.
---------------------	---

### 6.38.1.2 Create() [1/2]

```
static GameObject PC.UI.PopupWindow.Create (
    GameObject parent,
    float width,
    float height,
    string title,
    string message,
    ButtonAction buttonAction,
    params ButtonAction[] additionalButtonActions ) [static]
```

#### Parameters

<i>parent</i>	The parent transform to attach the popup window to.
<i>width</i>	The desired width of the popup window.
<i>height</i>	The desired height of the popup window.
<i>title</i>	The title of the popup window.



## Parameters

<i>message</i>	The message of the popup window.
<i>buttonAction</i>	The required single button of the popup window.
<i>additionalButtonActions</i>	The optional multiple buttons of the popup window.

## Returns

The GameObject created with the given parameters.

**6.38.1.3 Create() [2/2]**

```
static GameObject PC.UI.PopupWindow.Create (
    GameObject parent,
    string title,
    string message,
    ButtonAction buttonAction,
    params ButtonAction[] additionalButtonActions ) [static]
```

<param name="parent"The parent transform to attach the popup window to.

## Parameters

<i>title</i>	The title of the popup window.
<i>message</i>	The message of the popup window.
<i>buttonAction</i>	The required single button of the popup window.
<i>additionalButtonActions</i>	The optional multiple buttons of the popup window.

## Returns

The GameObject created with the given parameters.

**6.38.1.4 Init()**

```
void PC.UI.PopupWindow.Init (
    float width,
    float height,
    string title,
    string message,
    ButtonAction buttonAction,
    params ButtonAction[] additionalButtonActions ) [private]
```

## Parameters

<i>width</i>	The desired width of the popup window.
--------------	--

## Parameters

<i>height</i>	The desired height of the popup window.
<i>title</i>	The title of the popup window.
<i>message</i>	The message of the popup window.
<i>buttonAction</i>	The required single button of the popup window.
<i>additionalButtonActions</i>	The optional multiple buttons of the popup window.

## 6.39 PC.UI.ScrollRectNoMouseDrag Class Reference

### Public Member Functions

- override void **OnBeginDrag** (PointerEventData eventData)
- override void **OnDrag** (PointerEventData eventData)
- override void **OnEndDrag** (PointerEventData eventData)

## 6.40 PC.UI.Slot Class Reference

### Public Member Functions

- override Vector2Int **GetCellIndex** (Vector2 mousePos)  
*Gets the cell index relative to the container using the given mouse position.*
- override bool **RemoveItemAt** (Vector2Int cellIndex)  
*Removes the item at a given cellIndex from the container.*
- override bool **PlaceItemAt** (Item item, Vector2Int cellIndex)  
*Places an item at a given cellIndex in the container.*
- **Item GetItem** ()  
*Retrieves the item inside the slot.*

### Protected Member Functions

- override void **Start** ()

### Private Attributes

- **ItemType.Value i\_allowedType** = ItemType.Value.NONE
- **ItemType \_allowedType**

### Additional Inherited Members

#### 6.40.1 Member Function Documentation

##### 6.40.1.1 GetCellIndex()

```
override Vector2Int PC.UI.Slot.GetCellIndex (
    Vector2 mousePos ) [virtual]
```

## Parameters

<i>mousePos</i>	Position of the mouse.
-----------------	------------------------

## Returns

Cell index relative to the container.

Reimplemented from [PC.UI.ContainerBase](#).

**6.40.1.2 GetItem()**

```
Item PC.UI.Slot.GetItem ( )
```

## Returns

Returns the item inside the slot.

**6.40.1.3 PlaceItemAt()**

```
override bool PC.UI.Slot.PlaceItemAt (
    Item item,
    Vector2Int cellIndex ) [virtual]
```

## Parameters

<i>item</i>	The item you want to place.
<i>cellIndex</i>	The cell index you want to place the item at.

## Returns

Returns true if the operation was successful, otherwise false.

Reimplemented from [PC.UI.ContainerBase](#).

**6.40.1.4 RemoveItemAt()**

```
override bool PC.UI.Slot.RemoveItemAt (
    Vector2Int cellIndex ) [virtual]
```

### Parameters

<code>cellIndex</code>	The cell index of the item you want to remove.
------------------------	--

### Returns

Returns true if the operation was successful, otherwise false.

Reimplemented from [PC.UI.ContainerBase](#).

#### 6.40.1.5 Start()

```
override void PC.UI.Slot.Start ( ) [protected], [virtual]
```

Reimplemented from [PC.UI.ContainerBase](#).

## 6.41 PC.Stats.Stat Class Reference

Base class for all stats used to track character parameters.

### Public Member Functions

- int [GetValue](#) ()  
*Getter for the stat value.*

### Private Attributes

- int `_baseValue` = 0  
*The base value of the stat.*

### 6.41.1 Member Function Documentation

#### 6.41.1.1 GetValue()

```
int PC.Stats.Stat.GetValue ( )
```

### Returns

integer stat value

## 6.42 PC.Combat.TouchSensor Class Reference

Records any contact between the player and the enemy. This allows the enemy to apply damage to the player on contact.

## 6.43 PC.UI.VerticalLayoutGroupHandler Class Reference

### Private Member Functions

- void **Awake** ()
- void **Update** ()

### Private Attributes

- RectTransform **\_rectTransform**
- List< RectTransform > **\_children** = new List<RectTransform>()

## 6.44 PC.VFX.WFX\_LightFlicker Class Reference

Controls the flickering of the light for the muzzle flash.

### Private Member Functions

- IEnumerator **Flicker** ()

*Flickers the light component of this object. Object is meant to be a background light to muzzle flash*

### Private Attributes

- float **\_flickerTime** = 0.05f

### 6.44.1 Member Function Documentation

#### 6.44.1.1 Flicker()

```
IEnumerator PC.VFX.WFX_LightFlicker.Flicker ( ) [private]
```

Returns



# Index

- AddButton
  - PC.UI.PopupWindow, [48](#)
- Attack
  - PC.Combat.CharacterCombat, [14](#)
- AttackTarget
  - PC.Combat.EnemyCombat, [22](#)
  - PC.Combat.PlayerCombat, [41](#)
- AwakeExtension
  - PC.UI.DevConsoleMenu, [21](#)
  - PC.UI.InventoryMenu, [27](#)
  - PC.UI.MapMenu, [36](#)
  - PC.UI.MenuBase, [38](#)
  - PC.UI.PauseMenu, [39](#)
  - PC.UI.PlayerManagementStationMenu, [43](#)
  - PC.UI.PMSMenu, [45](#)
- belongsTo
  - PC.UI.ItemType, [33](#)
  - PC.UI.ItemTypes, [35](#)
- CloseExtension
  - PC.UI.DevConsoleMenu, [21](#)
  - PC.UI.InventoryMenu, [27](#)
  - PC.UI.MapMenu, [36](#)
  - PC.UI.MenuBase, [38](#)
  - PC.UI.PauseMenu, [40](#)
  - PC.UI.PlayerManagementStationMenu, [43](#)
  - PC.UI.PMSMenu, [45](#)
- Create
  - PC.UI.ItemContainerInfo, [31](#)
  - PC.UI.PopupWindow, [48, 49](#)
- Die
  - PC.Stats.CharacterStats, [14](#)
  - PC.Stats.EnemyStats, [23](#)
  - PC.Stats.PlayerStats, [44](#)
- EmptyCell
  - PC.UI.ContainerBase, [16](#)
- Equals
  - PC.UI.ItemType, [33](#)
- Flicker
  - PC.VFX.WFX\_LightFlicker, [53](#)
- GetCell
  - PC.UI.ContainerBase, [17](#)
- GetCellIndex
  - PC.UI.ContainerBase, [17](#)
  - PC.UI.Slot, [50](#)
- GetItem
  - PC.UI.Slot, [51](#)
- GetItemAt
  - PC.UI.ContainerBase, [17](#)
- GetMovement
  - PC.Entities.PlayerControllerBase, [42](#)
- GetOriginCellIndex
  - PC.UI.Item, [29](#)
- GetValue
  - PC.Stats.Stat, [52](#)
- GroundedCheck
  - PC.Entities.PlayerControllerBase, [42](#)
- Init
  - PC.UI.Item, [29](#)
  - PC.UI.PMSMenu, [46, 47](#)
  - PC.UI.PopupWindow, [49](#)
- InitBackground
  - PC.UI.ContainerBase, [18](#)
- InitContents
  - PC.UI.ContainerBase, [18](#)
- ItemContainerInfo
  - PC.UI.ItemContainerInfo, [30](#)
- Lookup
  - PC.UI.ItemType, [33](#)
- MakeCopy
  - PC.UI.Item, [29](#)
- OnAnimatorIK
  - PC.Entities.PlayerController, [42](#)
- OnPointerEnter
  - PC.UI.ContainerBase, [18](#)
- OnPointerExit
  - PC.UI.ContainerBase, [18](#)
- OpenExtension
  - PC.UI.DevConsoleMenu, [21](#)
  - PC.UI.InventoryMenu, [27](#)
  - PC.UI.MapMenu, [36](#)
  - PC.UI.MenuBase, [38](#)
  - PC.UI.PauseMenu, [40](#)
  - PC.UI.PlayerManagementStationMenu, [43](#)
  - PC.UI.PMSMenu, [47](#)
- operator!=
  - PC.UI.ItemType, [34](#)
- operator==
  - PC.UI.ItemType, [34](#)
- PC, [9](#)
- PC.Combat, [9](#)
- PC.Combat.CharacterCombat, [13](#)

- Attack, [14](#)
- PC.Combat.EnemyAnimationController, [22](#)
- PC.Combat.EnemyCombat, [22](#)
  - AttackTarget, [22](#)
  - SetTags, [22](#)
- PC.Combat.PlayerCombat, [41](#)
  - AttackTarget, [41](#)
- PC.Combat.TouchSensor, [53](#)
- PC.Entities, [9](#)
- PC.Entities.EnemyController, [23](#)
- PC.Entities.Gun, [23](#)
  - Reload, [24](#)
  - Shoot, [24](#)
- PC.Entities.GunSO, [24](#)
- PC.Entities.PlayerAnimatorController, [40](#)
- PC.Entities.PlayerController, [41](#)
  - OnAnimatorIK, [42](#)
- PC.Entities.PlayerControllerBase, [42](#)
  - GetMovement, [42](#)
  - GroundedCheck, [42](#)
- PC.Entities.PlayerManager, [44](#)
- PC.Input, [10](#)
- PC.Input.InputModule, [25](#)
- PC.Stats, [10](#)
- PC.Stats.CharacterStats, [14](#)
  - Die, [14](#)
- PC.Stats.EnemyStats, [23](#)
  - Die, [23](#)
- PC.Stats.PlayerStats, [44](#)
  - Die, [44](#)
- PC.Stats.Stat, [52](#)
  - GetValue, [52](#)
- PC.UI, [10](#)
- PC.UI.AlphaPMS, [13](#)
- PC.UI.BetaPMS, [13](#)
- PC.UI.ButtonAction, [13](#)
- PC.UI.Container, [14](#)
  - Start, [15](#)
- PC.UI.ContainerBase, [15](#)
  - EmptyCell, [16](#)
  - GetCell, [17](#)
  - GetCellIndex, [17](#)
  - GetItemAt, [17](#)
  - InitBackground, [18](#)
  - InitContents, [18](#)
  - OnPointerEnter, [18](#)
  - OnPointerExit, [18](#)
  - PlaceltemAt, [19](#)
  - RemoveltemAt, [19](#)
  - SetCell, [19](#)
  - SetItemParent, [20](#)
  - TakeItemAt, [20](#)
  - TransferItem, [20](#)
- PC.UI.DevConsoleMenu, [20](#)
  - AwakeExtension, [21](#)
  - CloseExtension, [21](#)
  - OpenExtension, [21](#)
- PC.UI.HorizontalLayoutGroupHandler, [25](#)
- PC.UI.Interactable, [26](#)
- PC.UI.InventoryMenu, [26](#)
  - AwakeExtension, [27](#)
  - CloseExtension, [27](#)
  - OpenExtension, [27](#)
  - TryPickingUpItem, [27](#)
  - TryReleasingItem, [27](#)
- PC.UI.Item, [28](#)
  - GetOriginCellIndex, [29](#)
  - Init, [29](#)
  - MakeCopy, [29](#)
  - TransferTo, [29](#)
- PC.UI.ItemContainerInfo, [30](#)
  - Create, [31](#)
  - ItemContainerInfo, [30](#)
- PC.UI.ItemSO, [31](#)
- PC.UI.ItemType, [31](#)
  - belongsTo, [33](#)
  - Equals, [33](#)
  - Lookup, [33](#)
  - operator!=, [34](#)
  - operator==, [34](#)
- PC.UI.ItemTypes, [35](#)
  - belongsTo, [35](#)
- PC.UI.LayoutMaintainPreferredHeight, [35](#)
- PC.UI.MapMenu, [36](#)
  - AwakeExtension, [36](#)
  - CloseExtension, [36](#)
  - OpenExtension, [36](#)
- PC.UI.MenuBase, [37](#)
  - AwakeExtension, [38](#)
  - CloseExtension, [38](#)
  - OpenExtension, [38](#)
- PC.UI.MenusController, [38](#)
- PC.UI.PauseMenu, [39](#)
  - AwakeExtension, [39](#)
  - CloseExtension, [40](#)
  - OpenExtension, [40](#)
- PC.UI.PlayerManagementStationMenu, [43](#)
  - AwakeExtension, [43](#)
  - CloseExtension, [43](#)
  - OpenExtension, [43](#)
- PC.UI.PMSBase, [44](#)
- PC.UI.PMSMenu, [45](#)
  - AwakeExtension, [45](#)
  - CloseExtension, [45](#)
  - Init, [46, 47](#)
  - OpenExtension, [47](#)
- PC.UI.PopupWindow, [47](#)
  - AddButton, [48](#)
  - Create, [48, 49](#)
  - Init, [49](#)
- PC.UI.ScrollRectNoMouseDrag, [50](#)
- PC.UI.Slot, [50](#)
  - GetCellIndex, [50](#)
  - GetItem, [51](#)
  - PlaceltemAt, [51](#)
  - RemoveltemAt, [51](#)



Start, [52](#)  
PC.UI.VerticalLayoutGroupHandler, [53](#)  
PC.VFX, [11](#)  
PC.VFX.WFX\_LightFlicker, [53](#)  
    Flicker, [53](#)  
PlaceItemAt  
    PC.UI.ContainerBase, [19](#)  
    PC.UI.Slot, [51](#)  
  
Reload  
    PC.Entities.Gun, [24](#)  
RemoveItemAt  
    PC.UI.ContainerBase, [19](#)  
    PC.UI.Slot, [51](#)  
  
SetCell  
    PC.UI.ContainerBase, [19](#)  
SetItemParent  
    PC.UI.ContainerBase, [20](#)  
SetTags  
    PC.Combat.EnemyCombat, [22](#)  
Shoot  
    PC.Entities.Gun, [24](#)  
Start  
    PC.UI.Container, [15](#)  
    PC.UI.Slot, [52](#)  
  
TakeItemAt  
    PC.UI.ContainerBase, [20](#)  
TransferItem  
    PC.UI.ContainerBase, [20](#)  
TransferTo  
    PC.UI.Item, [29](#)  
TryPickingUpItem  
    PC.UI.InventoryMenu, [27](#)  
TryReleasingItem  
    PC.UI.InventoryMenu, [27](#)