**TextColor – не трогать**

**TextColor – ожидает создания**

**TextColor – под вопросом**

**UIManagerComponent (UIManager) –** менеджер UI окон (виджетов). Хранит список окон, которые наследуются от **UIWindowBase.**

**Core/WidgetLogic/BaseLogic:**

* **UI\_BaseWindow –** родительский элемент от которого наследуются созданные UI окна. Хранит ссылку на **UIManager**.   
  Стиль именования новых окон: **UI\_[Имя]\_[Тип]Window**. Например: **UI\_MainMenu\_MainWindow, UI\_SettingsGraphics\_MainWindow,  
  UI\_MainMenu\_PauseWindow,  
  UI\_SettingsGraphics\_PauseWindow**
* **HUD\_Base** - родительский элемент от которого наследуются созданные HUD окна. (Heads Up Display (окно, на котором располагается различная информация, во время игрового процесса (жизнь игрока, количество ресурсов и т.д.)))  
  Стиль именования новых **HUDов**: **HUD\_[Имя]**. Например: **HUD\_Player, HUD\_Information**

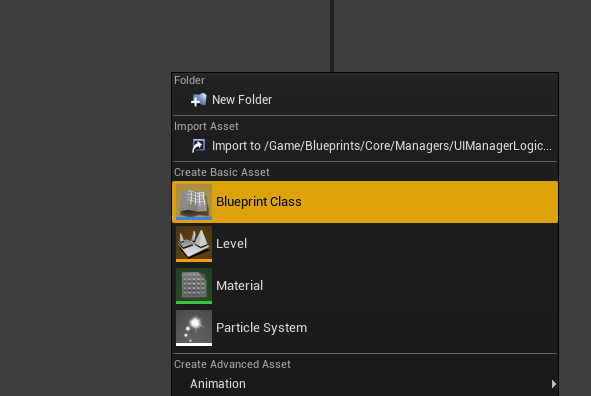
**Core/UIManagerLogic/Enum:**

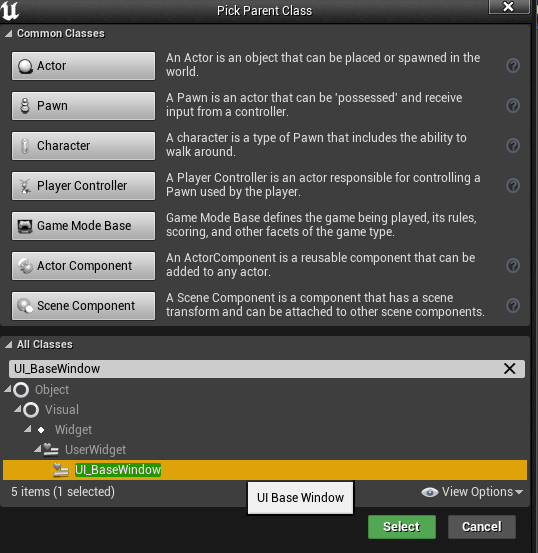
* **HUDTypes -** тут писать типы HUDов, которые будут создаваться
* **WindowTypes –** тут писать типы окон, которые будут создаваться (Например, MainMenu, Settings и т.д.)

**Core/UIManagerLogic/Structure:**

* **HUD\_Structure -** используется для массива HUDов (HUDs), чтоб назначить HUD в **BP\_MyGameMode.**
* **UI\_Window\_Structure –** используется для массива окон (Windows), чтоб назначить окна в **BP\_MyGameMode.**

**Процесс создания UI окна**

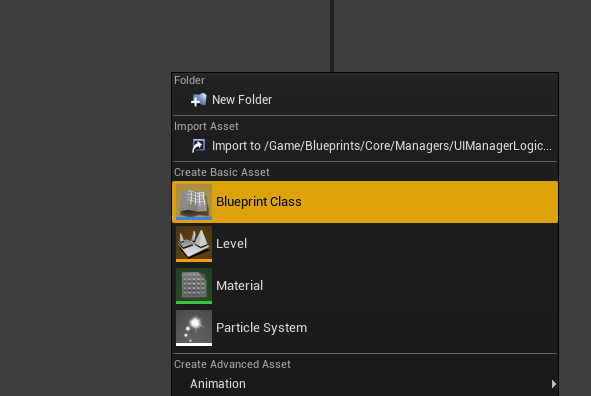
****

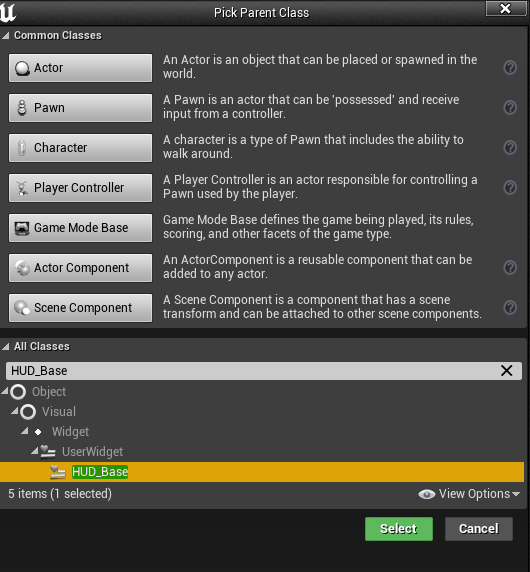


Создать **Blueprint Class**, от **UI\_Base\_Window.** Назвать, следуя стилю именования новых окон:

«Стиль именования новых окон: **UI\_[Имя]\_Window**. Например: **UI\_MainMenu\_Window, UI\_Settings\_Window»**

**Процесс создания HUD**

****



Создать **Blueprint Class**, от **HUD\_Base.** Назвать, следуя стилю именования новых HUDов:

«Стиль именования новых **HUDов**: **HUD\_[Имя]**. Например: **HUD\_Player, HUD\_Information»**

**Методы:**

**Category [Windows]:**

**public:**

* **InitializeWindows** - создает все окна из массива **Windows** и добавляет их в словарь **WindowsGroup** чтоб можно было открывать окна.
* **ShowWindowNew** - показывает окно за указанным типом. Добавляет это окно в конец стека.
* **ShowWindowToggle** - показывает окно за указанным типом. Заменяет последнее окно стека на текущее открытое. Использовать для окон, в которых нужно типа просто обновить содержимое.  
  Например: сверху несколько кнопок, а нужно заменять только область под кнопками.
* **CloseWindow -** закрывает все окна, очищает стек окон.
* **BackWindow** - возвращается на предыдущее окно в стеке.
* **GetWindow** - получает окно из словаря **WindowsGroup** по указанному типу.

**private:**

* **GetLastWindowOfStack** - получает последнее, добавленное в стек, окно. Приватный метод, доступен только в пределах менеджера.
* **SetLastWindowOfStack** - заменяет последний элемент в стеке (заменяет последнее добавленное окно на текущее).

**Category [HUDs]:**

**public:**

* **InitializeHUDs** - создает все окна из массива **HUDs** и добавляет их в словарь **HUDsGroup** чтоб можно было открыть нужный HUD.
* **GetHUD** - получает HUD по указанному типу.
* **ShowHUD** - показывает HUD за указанным типом.
* **CloseHUD** - закрывает HUD за указанным типом.

**Переменные:**

**Category [Windows]:**

**public:**

* **Windows** - помещаем созданные UI окна для инициализации.

**private:**

* **WindowsGroup** - словарь (ключ/значение (ТипОкна/Окно)), в котором хранятся созданные UI окна после инициализации.
* **WindowsStack** - стек окон, который увеличивается по мере открытия UI окон, уменьшается, если возвращаемся назад по окнах, или очищается, если закрываем все окна.

**Category [HUDs]:**

**public:**

* **HUDs** - помещаем созданные **HUD** окна для инициализации.

**private:**

* **HUDsGroup** - словарь в котором хранятся HUD окна.

**Category [Default]:**

**public:**

* **!!!HUD\_Information** - HUD, в котором должна храниться различная информация (текущие таски, всплывающие окна и т.д.). (**Заменить на HUD\_Player. Чтоб все было на одном HUD**)