**2. HealhProgressBar. Blueprint Widget**

1. Добавляем функцию для получения процентов в HealthComponent и проверку для MaxHealth

2. Нужно создать пользовательский интерфейс – как это делаем (что за компонент создаем)? Какой редактор открывается, как называется, его структура справа и слева? Специальной тип элемента для группировки других.

3. Добавляем элемент для количества жизней. Из чего состоит? Как настроили цвет? Что есть у многих параметров в виджете? У кого создаем? Что автоматически создалось? На какие две части делится UMG? Как настраиваем функцию?

4. В какой класс можно добавить виджет? В какой логичнее всего? Как это сделать?

1. Для начала добавим дополнительную функцию в HealthComponent, которая будет возвращать количество процентов текущего здоровья от максимального. А так же проверку нашего MaxHealth на BeginPlay:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

2. Для создания пользовательского интерфейса в новой папке UI через ПКМ->UserInterface->WidjetBlueprint создаем виджет WBP\_PlayerHUD.

Нам открывается редактор интерфейсов UE UMG – Unreal Motion Graphics:

Изображение выглядит как текст, монитор, электроника, снимок экрана

Автоматически созданное описание

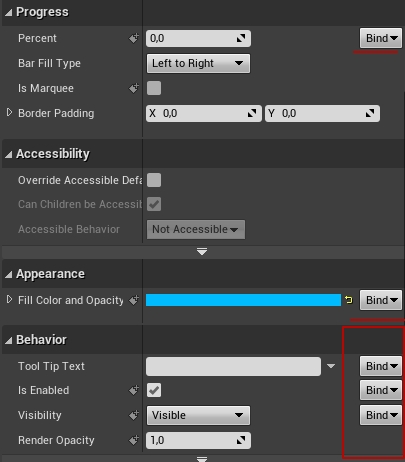
Слева находится окно Palette – здесь находятся все доступные нам элементы UI.

Ниже находится окно Hierarchy – каким образом UI компоненты сгруппированы друг относительно друга. Корневой компонент – Canvas Panel.

Вытащим элемент Horizontal Box – он служит для выравнивания и группировки других элементов в виде строки. Через окно Hierarchy мы и можем туда перетащить нужные нам элементы. Таким образом они становятся дочерними элементами и их трансформация теперь меняется в зависимости от трансформации родителя.

3. Выведем количество жизней. Добавим элемент с названием ProgressBar. Состоит из двух картинок – фона и второй, которая отвечает за заполнение. В панели Details поменяли у него в параметре Style->FillImage->Tint цвет.

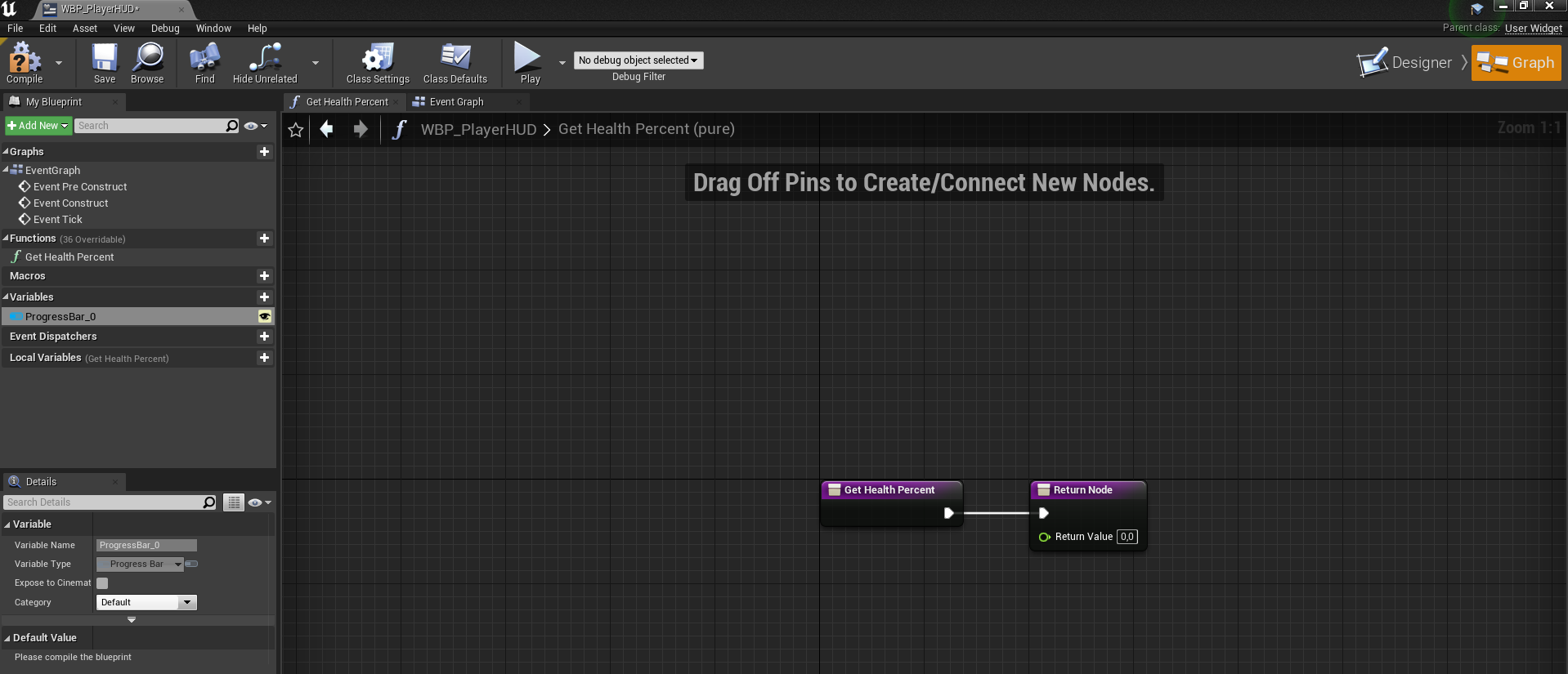
Еще один важный концепт, который нам нужно знать – это биндинг проперти (он есть напротив множества параметров). Это позволяет создать функцию Update для конкретного проперти в UI.

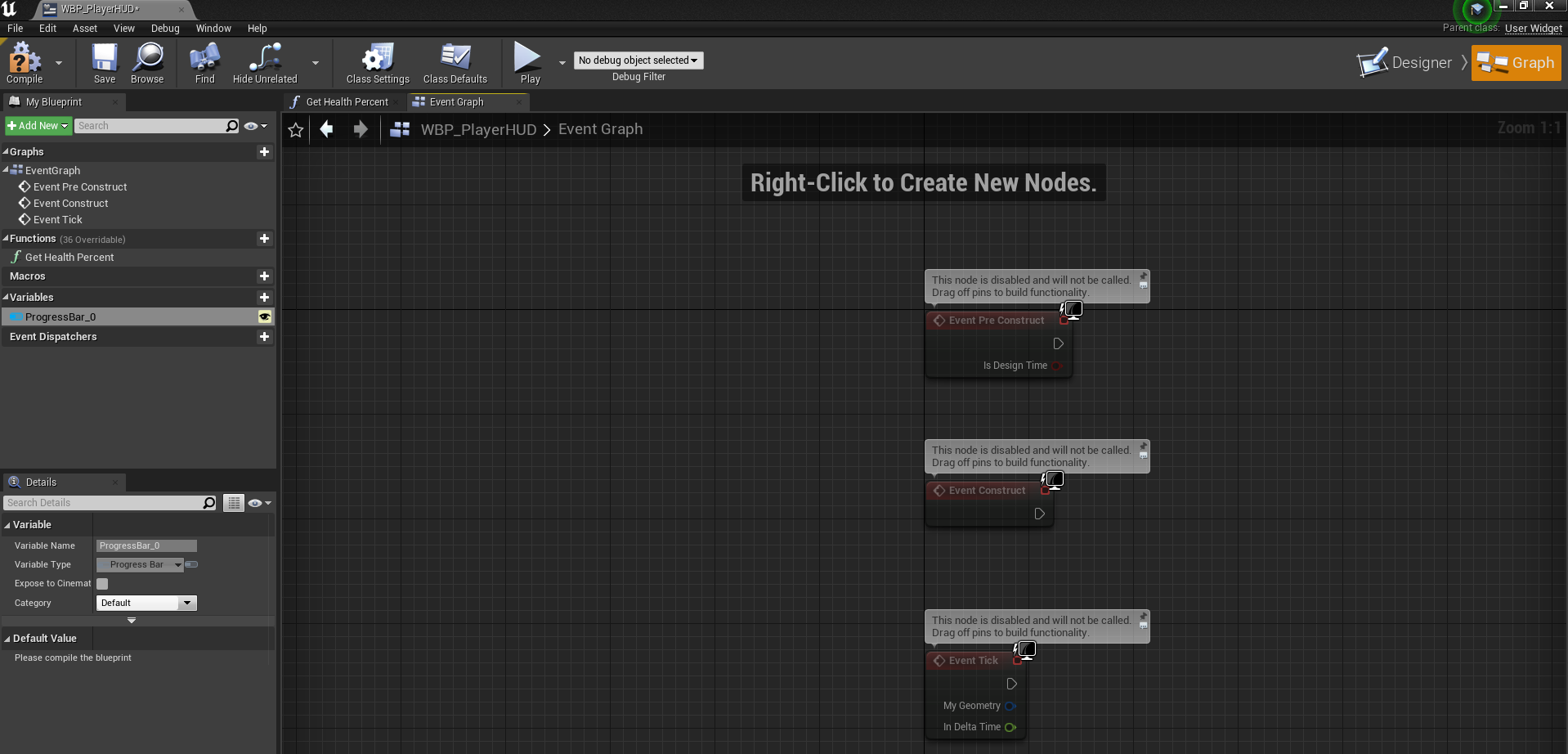


Нажав на Bind->CreateBinding – у нас автоматически создалась функция GetPercent и редактор интерфейса имеет два окна – Designer – где мы изначально настраивали положение Progress Bar, и Graph – который является аналогом EventGraph блюпринта, то есть мы так же можем изменять интерфейс в зависимости от событий в игре.

Изображение выглядит как текст, внутренний, часы, экран

Автоматически созданное описание





Переименуем функцию и добавим ей функционал. Мы забиндили функцию на проперти Percent – это означает, что при любом изменении возвращаемого значения, мы получим обновленный Percent в UI.

Получим доступ к нашему персонажу. Для этого в виджетах существует специальная функция Get Owning Player Pawn, возвращающая указатель на пауна, которого мы **контролируем** в данный момент. Так как все наши компоненты у персонажа protected, надо вызвать шаблонную функцию Get Component By Class и выбрать наш класс. Оттуда мы уже вызываем нашу функцию GetHealthPercent из C++:

Изображение выглядит как текст, внутренний

Автоматически созданное описание

4. Добавить данный виджет во вьюпорт можно в абсолютно любом классе – персонаже, GameMode, контроллере и т.д.

Но логичнее всего это делать в специальном классе HUD (STUGameHUD), где мы рисовали наш интерфейс – он будет контроллером наших интерфейсов. Для этого создадим от него BP-класс. И переходим в ивент граф.

На BeginPlay создадим наш виджет с помощью ноды Create Widjet. В параметре Class выбрали наш Widjet. После этого на созданном объекте вызвать Add To Viewport. Не забываем поменять в WorldSettings наш HUD-класс.

Изображение выглядит как текст, внутренний, снимок экрана

Автоматически созданное описание