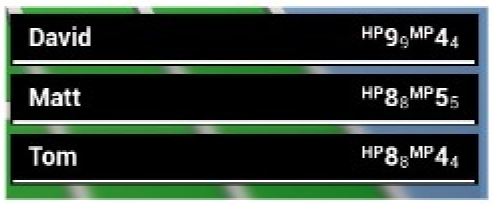
Feature - RPG Party Interface
BP_UnitBase - Character
BP_CombatComponent - Actor Component
WBP_BattleHUD - Widget
WBP_PartyInterface - Widget
WBP_PartyInterfaceSlot - Widget
BP_BattleTurnMode - Game Mode Base
BP_TurnController - Player Controller



BP_UnitBase

Involved Variables
Name - Text
HP_Max - Float
HP_Current - Float
MP_Max
MP_Current - Float

Involved Functions
Start Up Function - Event BeginPlay

WBP_BattleHUD

Involved Variables
WBP_PartyInterface - Object Reference

Involved Functions
Add Unit to HUD Function - Custom Function

BP_CombatComponent

Involved Variables
DexMod - Float
ConMod - Float
WisMod - Float
Initiative - Float
PlayerLevel - Float

WBP_PartyInterface

Involved Variables
PartyUnitList - Object Reference

BP_TurnController

Involved Variables
Battle HUD - Widget Object Reference

Involved Functions
Start Up Function - Event BeginPlay

WBP_PartyInterfaceSlot

Involved Variables
HPCurrent - Object Reference
HPMax - Object Reference
MPCurrent - Object Reference
MPMax - Object Reference
NameText - Object Reference
Unit - Object Reference

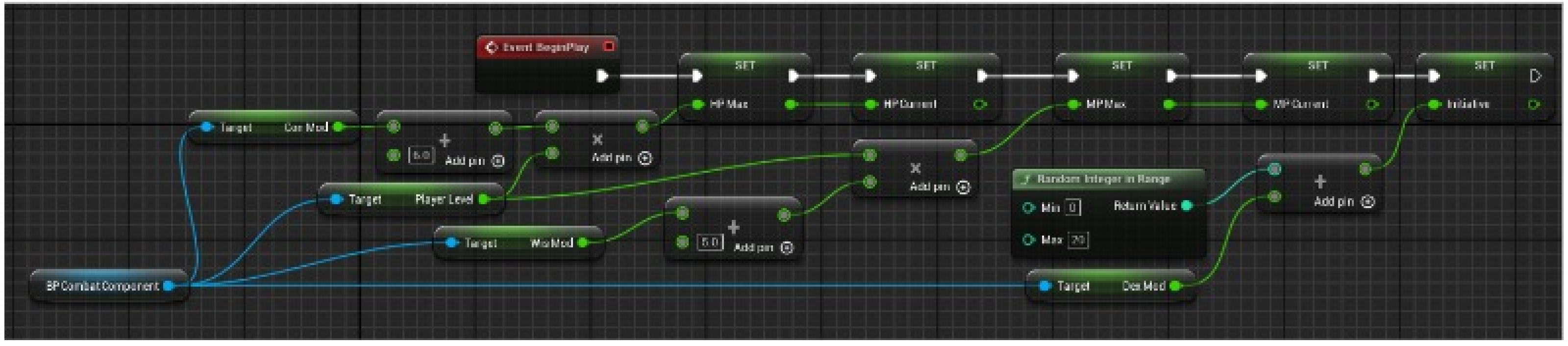
Involved Functions
Start Up Function - Event Pre Construct

BP_BattleTurnMode

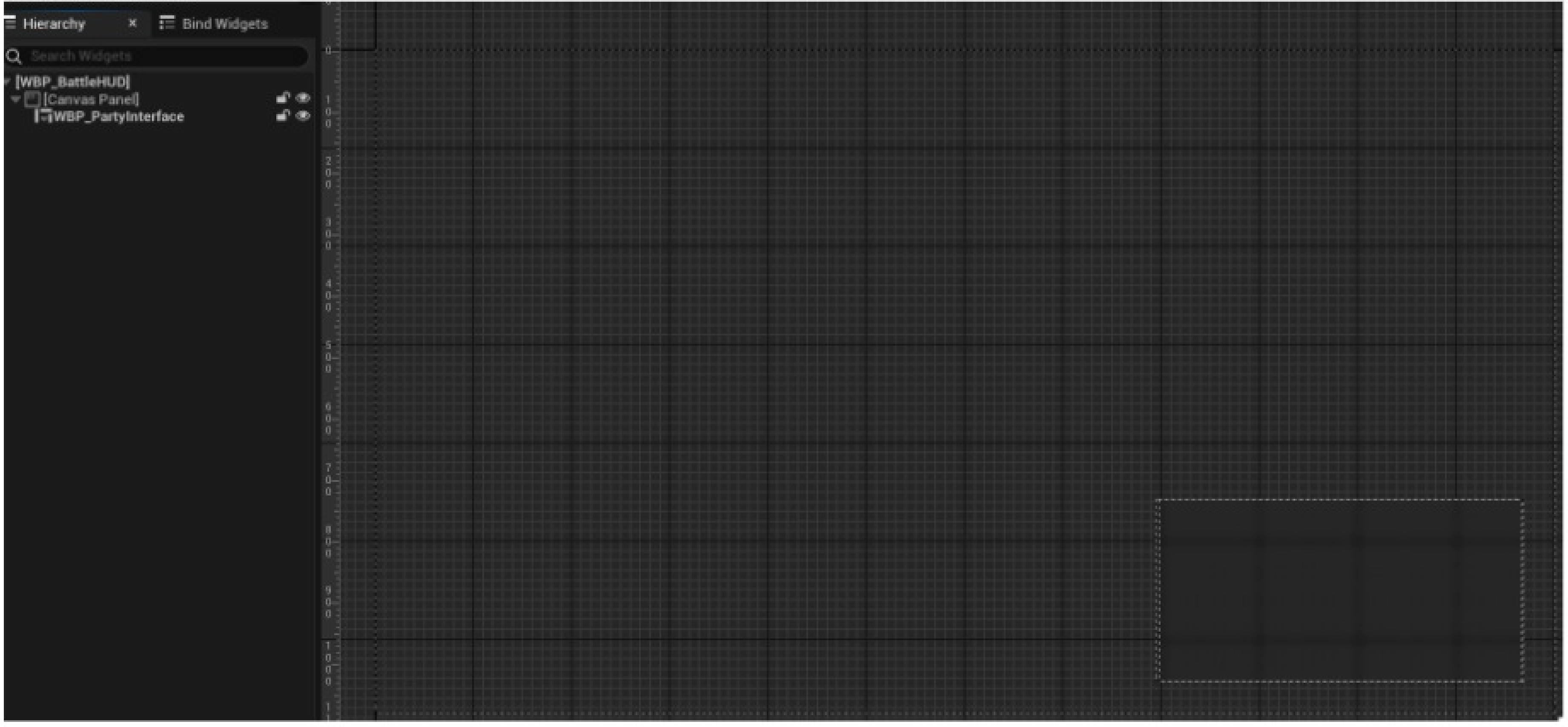
Involved Variables
PartyUnits - Object Reference Array
EnemyUnits - Object Reference Array
Battle HUD - Widget Object Reference

Involved Functions
Start Up Function - Event BeginPlay

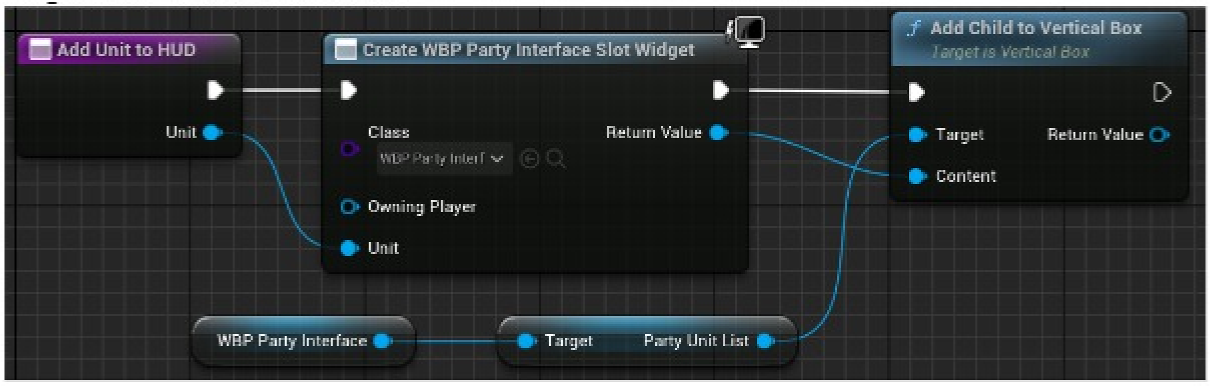
BP_UnitBase - Start Up Function



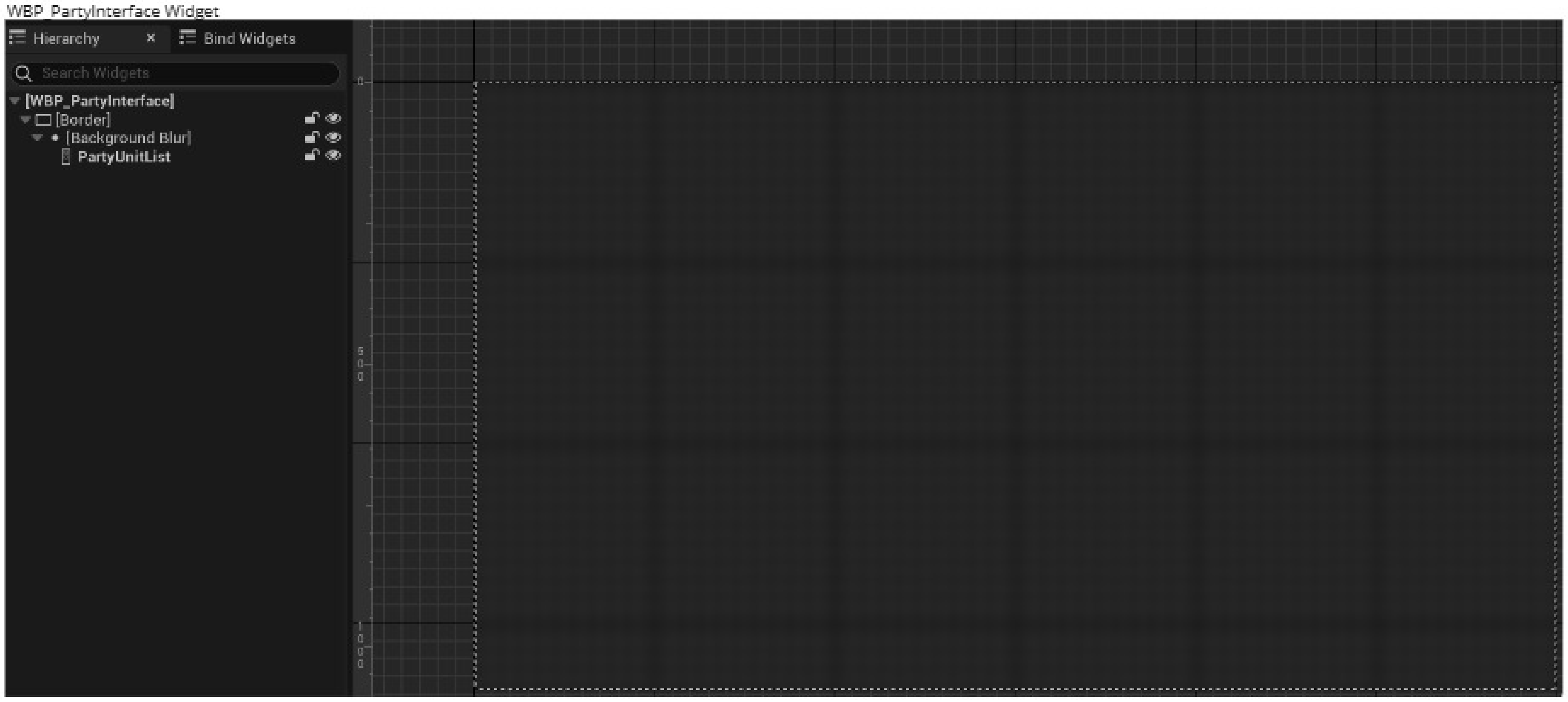
WBP_BattleHUD Widget



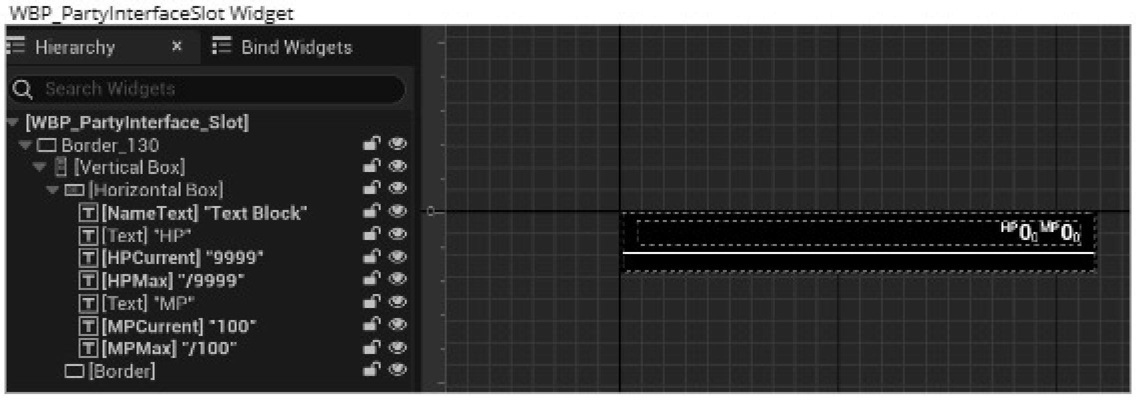
WBP_BattleHUD - Add Unit to HUD Function



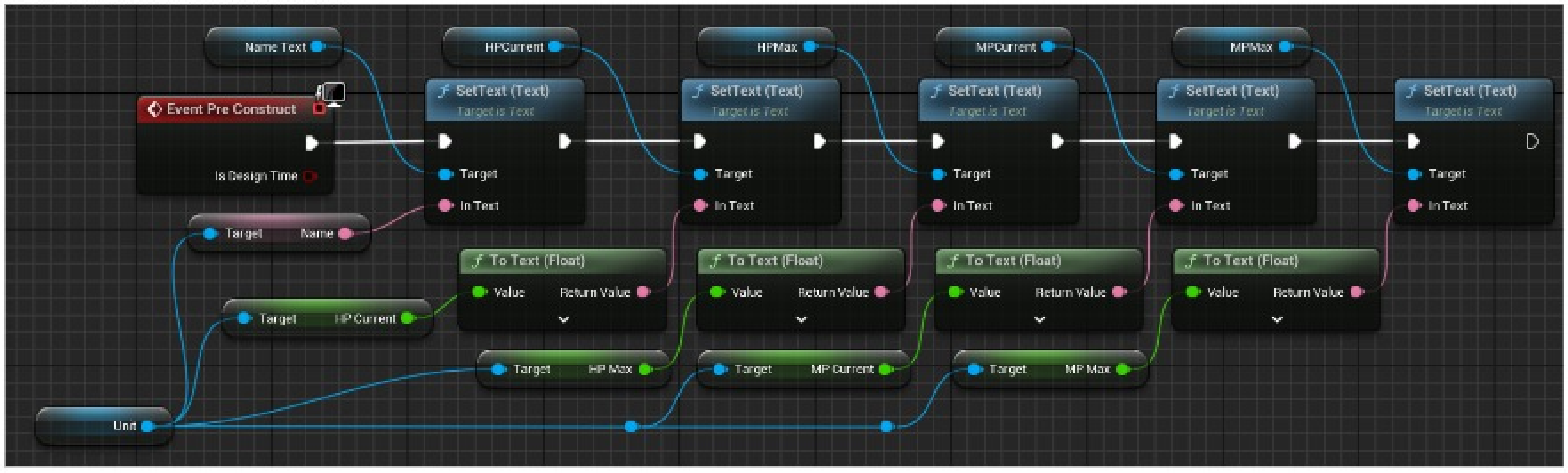
WBP_PartyInterface Widget



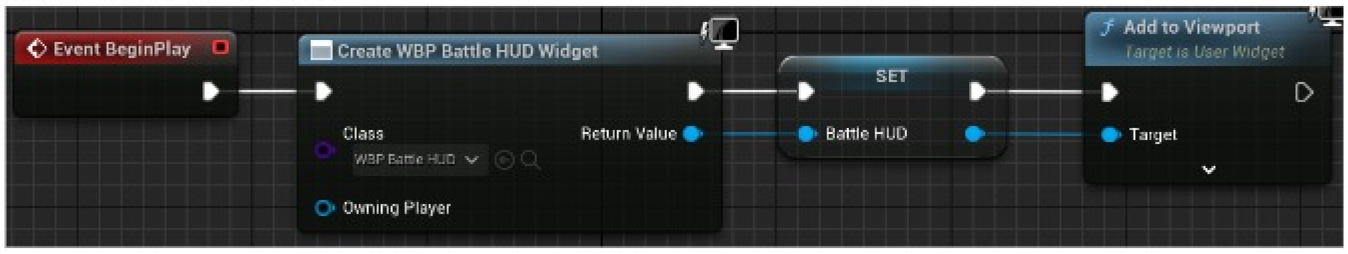
WBP_PartyInterfaceSlot Widget



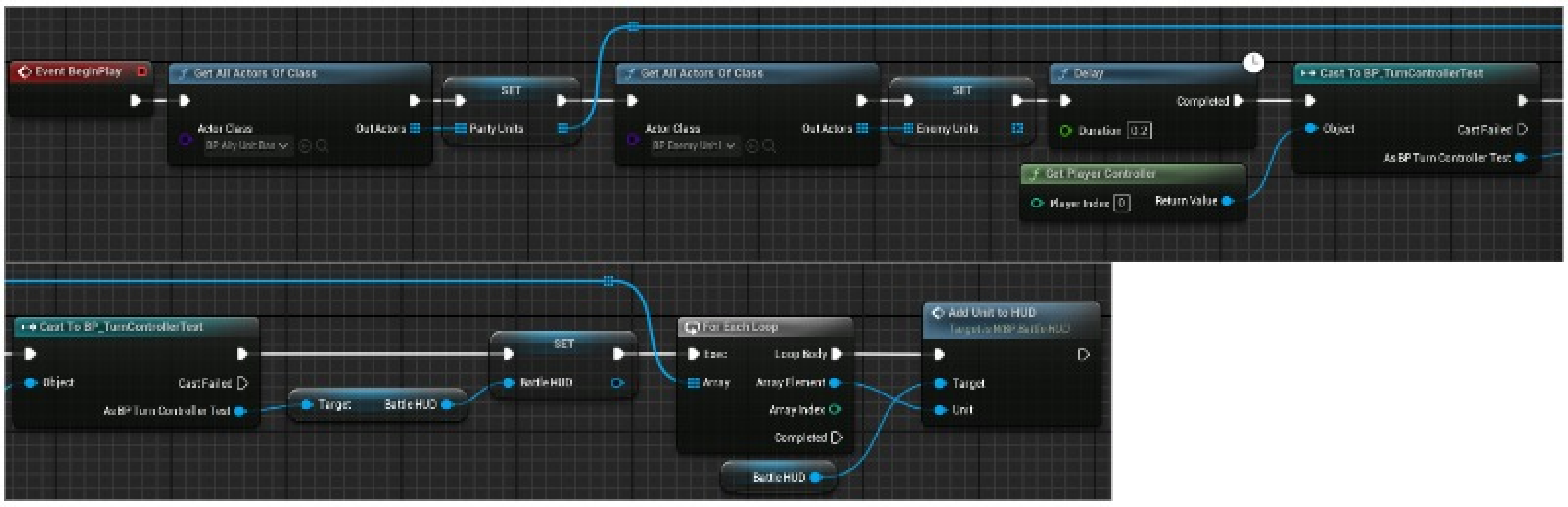
WBP_PartyInterfaceSlot Start Up Function



BP_TurnController Start Up Function



BP_BattleTurnMode Start Up Function



Pseudo Code

BP_UnitBase Start Up Function

```
On Event BeginPlay
   Set float variable HP_Max equal to BP_CombatComponent variable ConMod plus six, then multiplied by BP_CombatComponent PlayerLevel value
   Set float variable HP_Current equal to HP_Max
   Set float variable MP_Max equal to BP_CombatComponent variable WisMod plus five, then multiplied by BP_CombatComponent PlayerLevel value
   Set float variable MP_Current equal to MP_Max
   Set float variable Initiative equal to BP_CombatComponent variable DexMod added to a random integer from 1 to 20.
```

WBP_BattleHUD Add Unit to HUD Function

```
Custom function Add Unit to HUD with Input Object Reference variable Unit.
   Create Widget WBP_PartyInterfaceSlot with Input Unit
   Add Child to Vertical Box with Content the Return Value of created Widget, and Target the PartyUnitList Object Reference of WBP_PartyInterface
```

WBP_PartyInterfaceSlot Start Up Function

```
On Event Pre Construct
   Set Text of Object Reference Name Text to Name of Object Reference Unit
   Set Text of Object Reference HPCurrent Text to value of HP_Current Float of Object Reference Unit
   Set Text of Object Reference HPMax Text to value of HP_Max Float of Object Reference Unit
   Set Text of Object Reference MPCurrent Text to value of MP_Current Float of Object Reference Unit
   Set Text of Object Reference MPMax Text to value of MP_Max Float of Object Reference Unit
```

BP_TurnController Start Up Function

```
On Event BeginPlay
   Create Widget WBP_BattleHUD
   Set Return Value as variable BattleHUD Object Reference
   Add to Viewport
```

BP_BattleTurnMode Start Up Function

```
On Event BeginPlay
   Get all Actors of Class BP_AllyUnitBase and make output a variable Object Reference Array called PartyUnits
   Get all Actors of Class BP_EnemyUnitBase and make output a variable Object Reference Array called EnemyUnits
   Delay 0.2 seconds
   Cast to BP_TurnController with object Player Controller
   As BP_TurnController, create variable BattleHUD using BattleHUD variable from BP_TurnController.
   For Each item in Array PartyUnits, call function Add Unit to HUD with Target BattleHUD and Unit input of each Array Element
```

Summary and Explanation

BP_UnitBase Start Up Function

This function triggers once at the start for each unit in play. BP_UnitBase is a parent with children BP_AllyUnitBase and BP_EnemyUnitBase from which each character in play is derived. The purpose of this function is to determine each character's stats using the BP_CombatComponent that is attached. In this example, the BP_CombatComponent uses Dungeons and Dragons style stats, though this could easily be changed by renaming the variables and changing their values.

First, the function starts with getting the ConMod value from BP_CombatComponent and adding six before multiplying that value by the PlayerLevel. This is a simplification of how D&D handles calculating hit points which then sets the variable HP_Max's value. The HP_Current value is then set to be equal to the HP_Max as the understanding is that each character will begin a battle stage with full health. Similarly, it determines the MP_Max value using the BP_CombatComponent's WisMod value added to 5 and multiplied by the PlayerLevel. This is a step away from D&D mechanics and could easily be changed to Spell Slots, though MP is generally more common in RPGs. The MP_Current is then set to equal MP_Max as players should also start each level with maximum magic points. Lastly, the function determines the unit's Initiative value by taking the BP_CombatComponent's DexMod value and adding it to a random integer from 1 to 20. This is again how D&D handles initiative, by rolling a 20-sided die and adding the player's dexterity modifier.

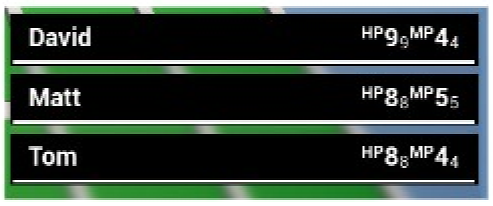
WBP_BattleHUD Add Unit to HUD Function
This simple function serves to retrieve information for each individual unit that calls the function and then create a unique slot for it in the party interface. It does so by requiring an input from an Object Reference, titled Unit, and then creating a WBP_PartyInterfaceSlot widget with its information. It then takes this created slot widget and adds it to the PartyUnitList Vertical Box object in the WBP_PartyInterface widget.

WBP_PartyInterfaceSlot Start Up Function
This function uses information from the Unit object reference to change the text of different values as well as determining what name should display. First it sets the NameText component with the Name information from Unit. It then fills out HPCurrent, HPMax, MPCurrent, and MPMax using the similar values stored in Unit.

BP_TurnController Start Up Function
The purpose of this little function is to add the BattleHUD window to the player's screen at the start of the combat stage. It creates a widget for WBP_BattleHUD and stores it as a useful variable before adding it to the Viewport.

BP_BattleTurnMode Start Up Function
This function serves to gather the required information and then feed it into the Add Unit to HUD function and calling it to build the party interface with the individual slots. First is gets all actors of the BP_AllyUnitBase and it's children, creating an Array variable called PartyUnits with its output. It then does the same with BP_EnemyUnitBase, creating an array EnemyUnits. Here we have a short 0.2 seconds delay to allow the arrays and the BP_TurnController to finish before we cast to the BP_TurnController. We get its BattleHUD variable and create a variable inside BattleTurnMode with it. The function then enters a For Each Loop using the PartyUnits array. For each item in the array, meaning for each friendly party unit, it calls the Add Unit to HUD function with Target of the BattleHUD variable we just stored and each array element feeding into the Unit object reference slot.

With these steps completed, the party interface feature is finished and will create a window detailing friendly units in play as shown below.



David $^{HP}9_{9}{}^{MP}4_{4}$

Matt $^{HP}8_{8}{}^{MP}5_{5}$

Tom $^{HP}8_{8}{}^{MP}4_{4}$