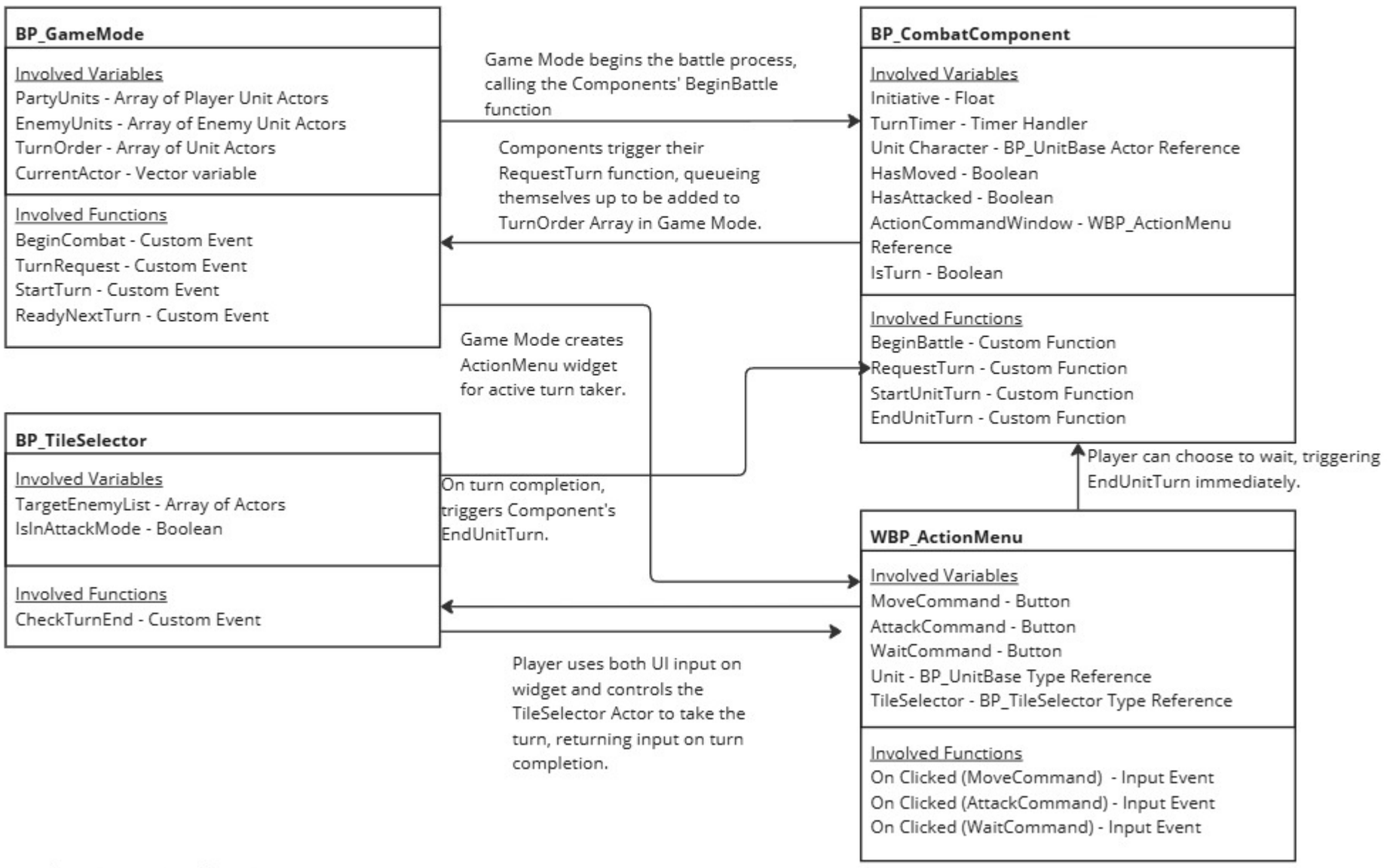
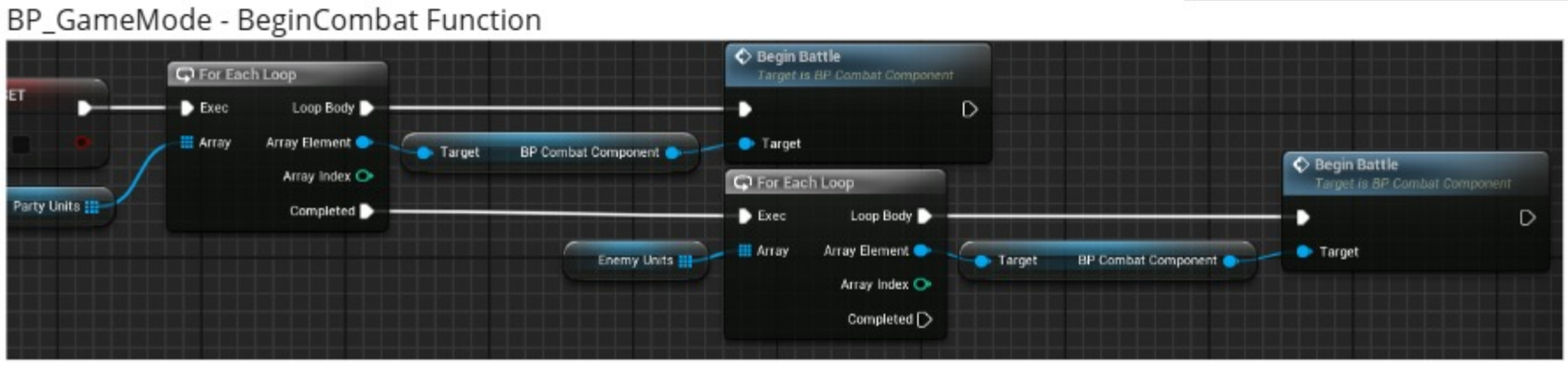
Feature - Turn Implementation
BP_GameMode - Game Mode
BP_CombatComponent - Actor Component
WBP_ActionMenu - Widget
BP_TileSelector - Actor

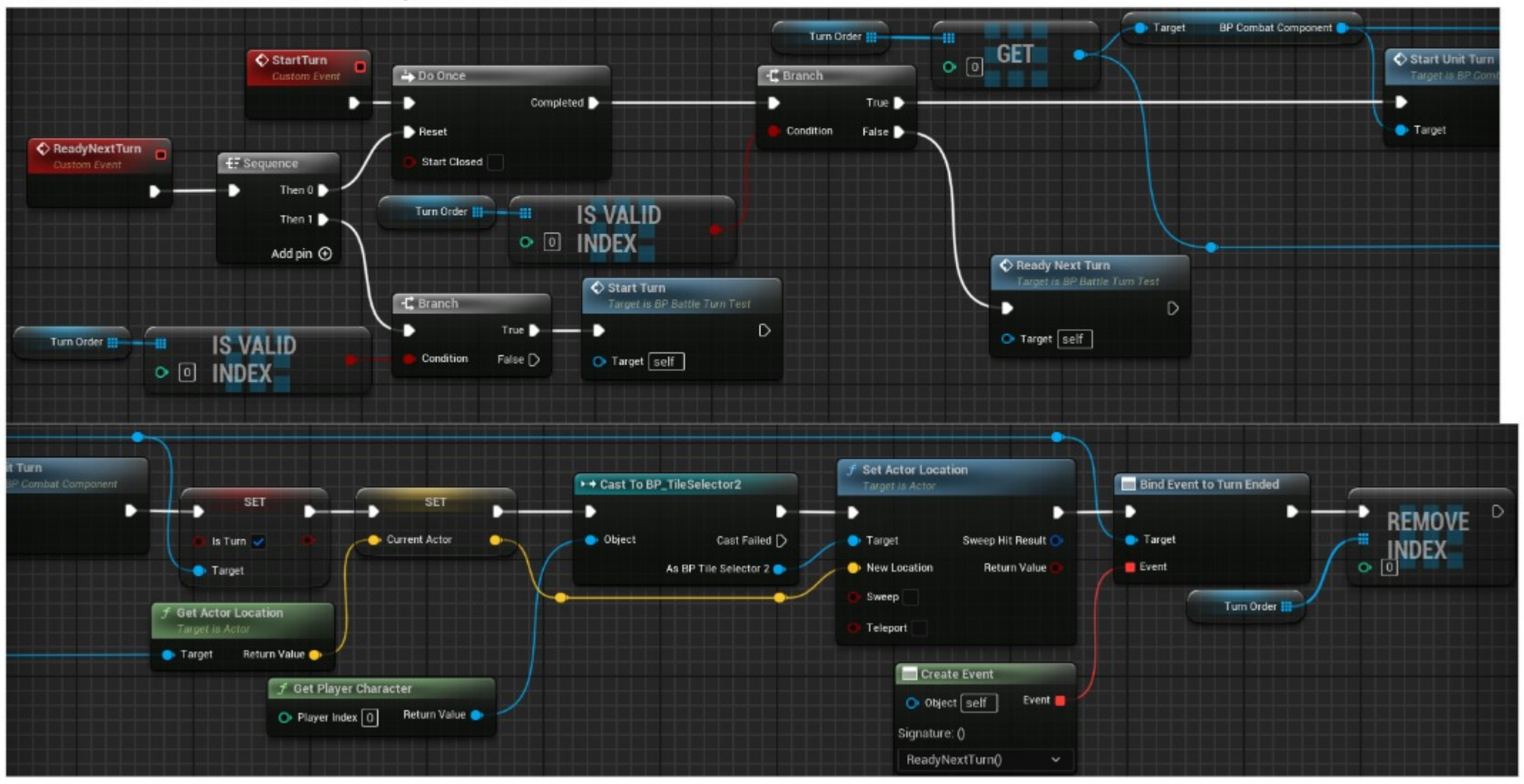


BP_GameMode - BeginCombat Function

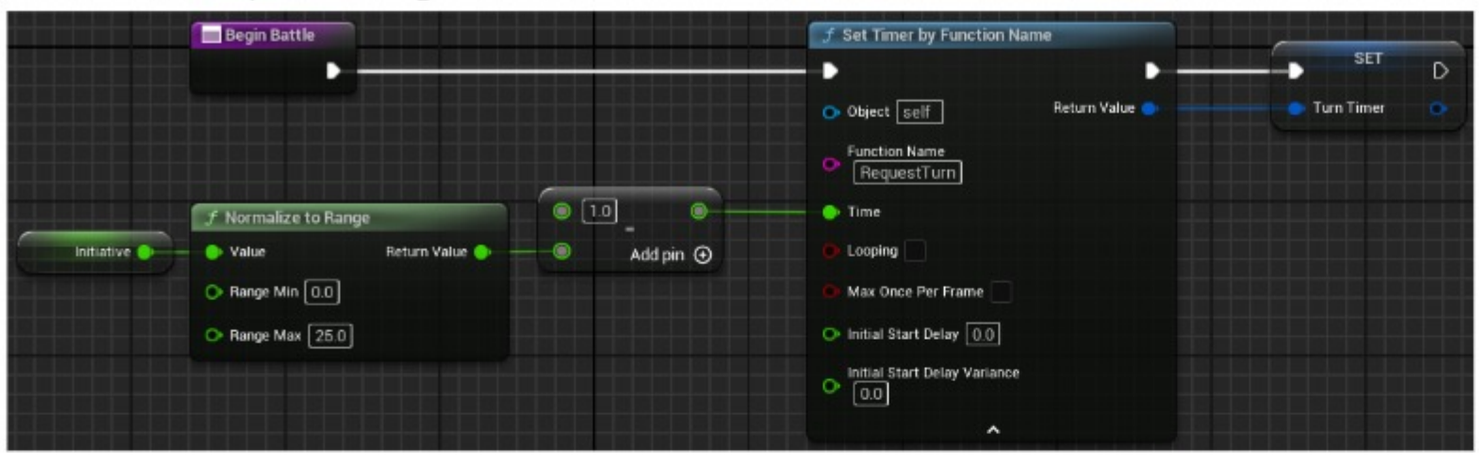


BP_GameMode - TurnRequest Function

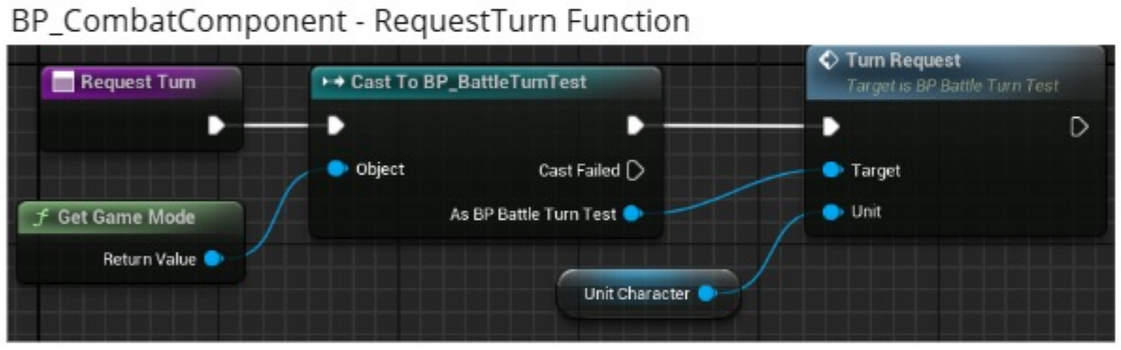
BP_GameMode - StartTurn and ReadyNextTurn Functions



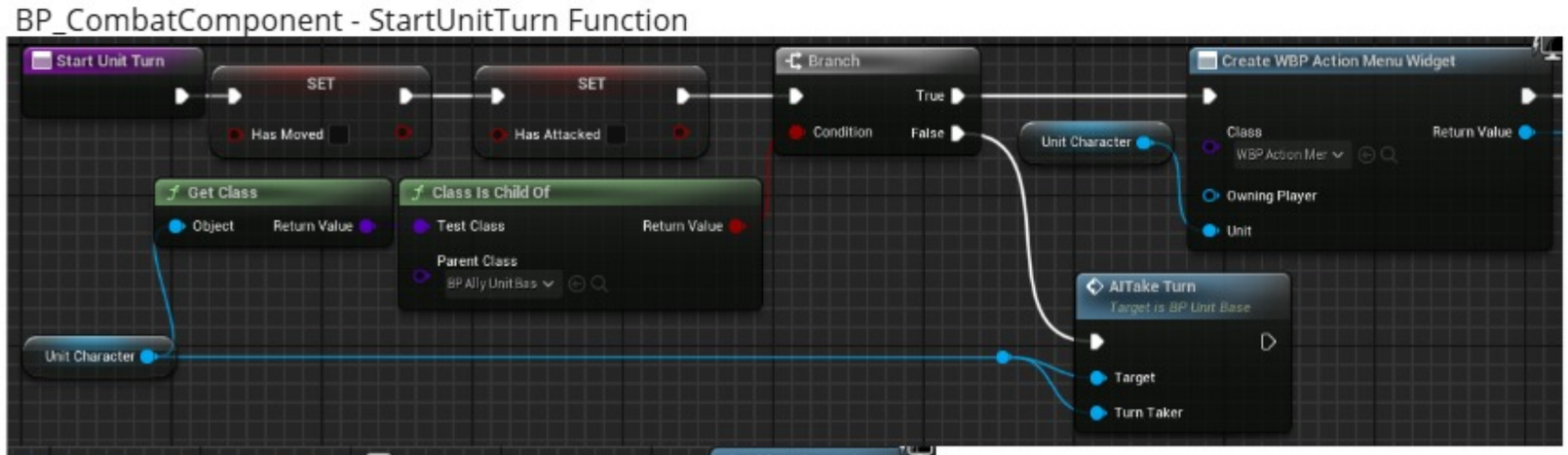
BP_CombatComponent - BeginBattle Function



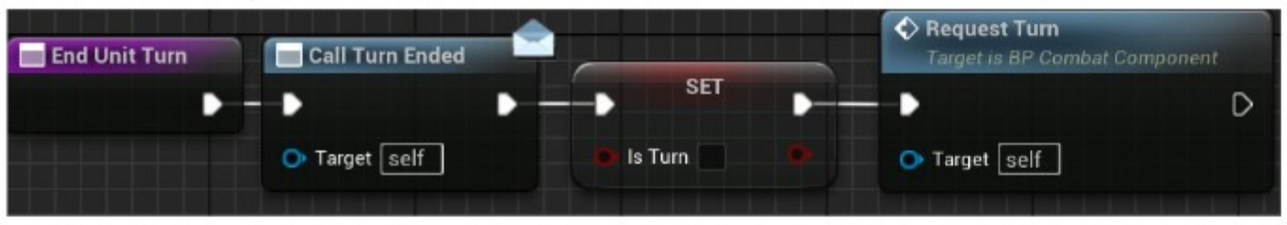
BP_CombatComponent - RequestTurn Function



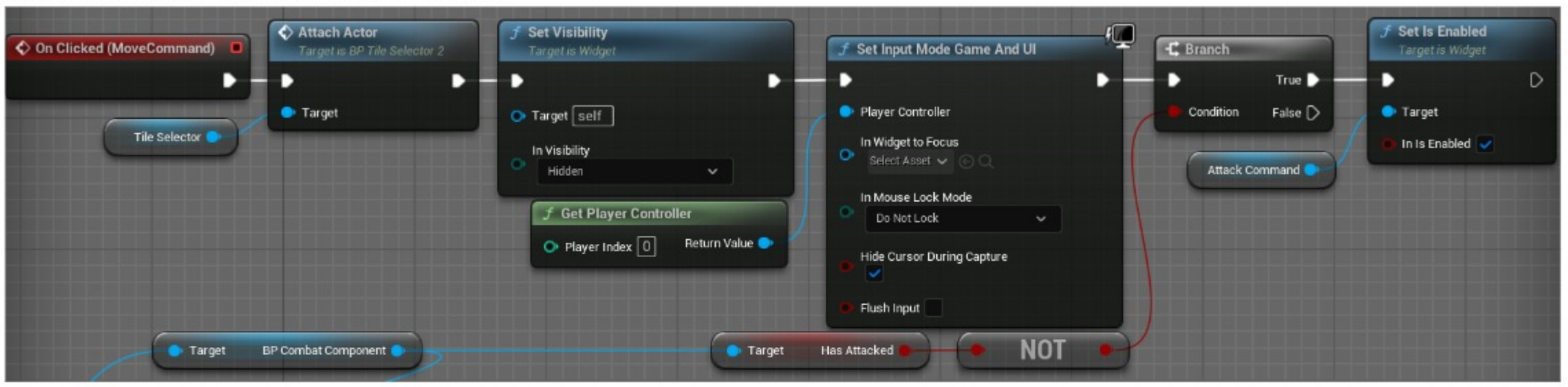
BP_CombatComponent - StartUnitTurn Function



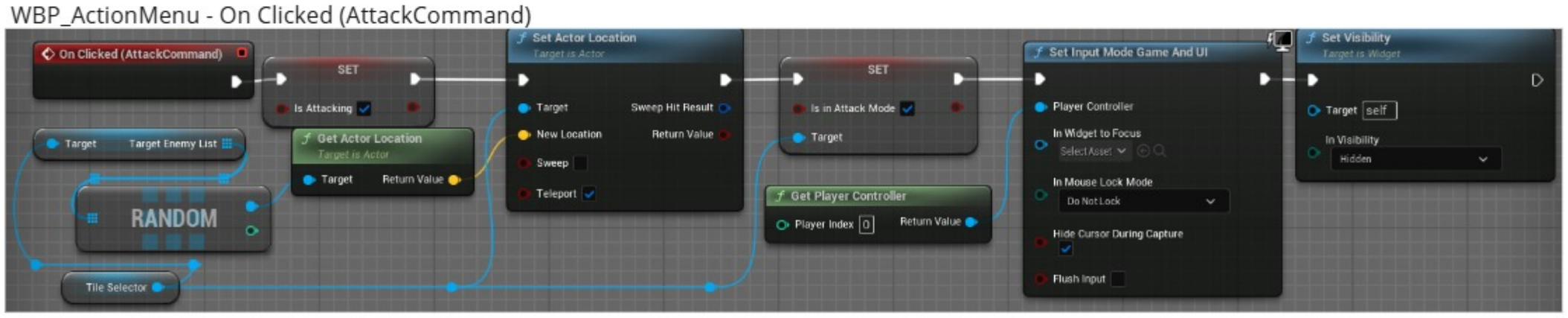
BP_CombatComponent - EndUnitTurn Function



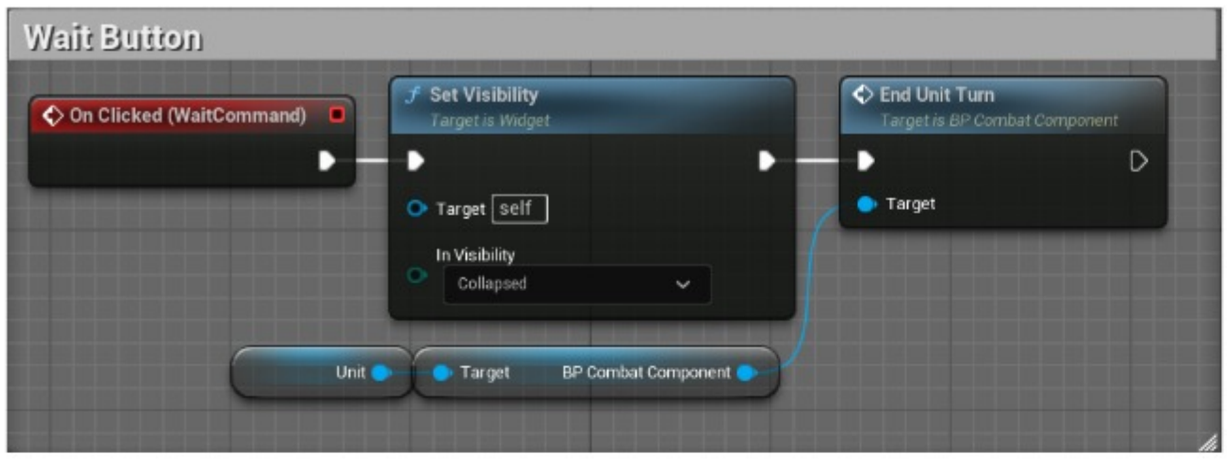
WBP_ActionMenu - On Clicked (MoveCommand)



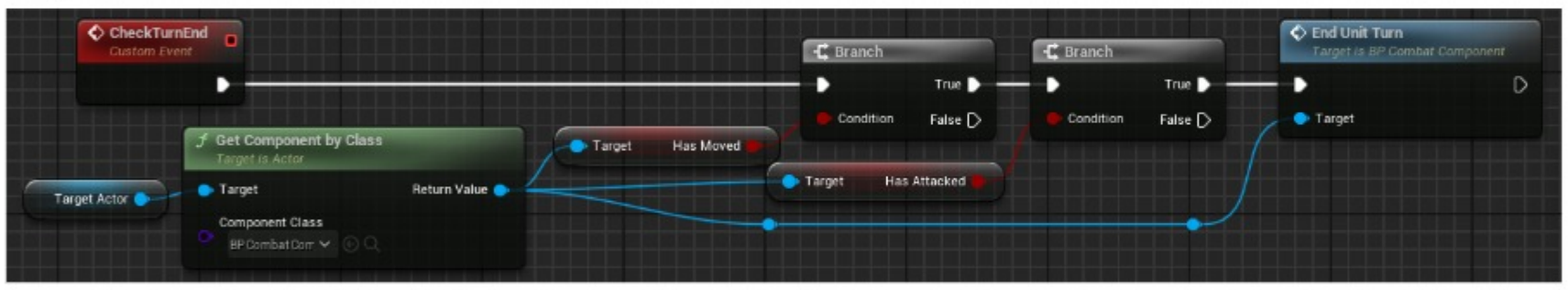
WBP_ActionMenu - On Clicked (AttackCommand)



WBP_ActionMenu - On Clicked (WaitCommand)



BP_TileSelector - CheckTurnEnd Function



Pseudo Code

BP_GameMode - BeginCombat Function

On Event called
- For each element in Array PartyUnits
  - Trigger BeginBattle Function in Target's BP_BattleComponent Component.
- On Completed loop
- For each element in Array EnemyUnits
  - Trigger BeginBattle Function in Target's BP_BattleComponent Component.

BP_GameMode - TurnRequest Function

On Event called with Input of Unit BP_UnitBase actor reference
- Add Unit to Array TurnOrder
- Trigger Event StartTurn

BP_GameMode - StartTurn Function

On Event called
- Do once
  - If the index of 0 of TurnOrder Array is valid
    - Call UnitStartTurn Function on BP_CombatComponent of Unit at index 0 of TurnOrder Array.
    - Set IsTurn Boolean on BP_CombatComponent of Unit to true.
    - Set CurrentActor Vector to Actor Location of Unit.
    - Cast to BP_TileSelector as a Player Character.
    - Set Actor Location of BP_TileSelector to CurrentActor Vector.
    - Bind Event to Turn Ended in BP_CombatComponent of Unit, using created Event with signature ReadyNextTurn()
    - Remove index 0 of TurnOrder Array.
  - If index of 0 is not valid
    - Call ReadyNextTurn Event

BP_GameMode - ReadyNextTurn Function

On Event called
- Enter Sequence
  - First, reset Do Once in StartTurn Function
  - Then, if 0 is valid index of TurnOrder Array
    - Call StartTurn Function.

BP_CombatComponent - BeginBattle Function

On Function called
- Set Timer by Function Name 'RequestTurn' with Time determined by Initiative Float Normalized to Range between 0 and 25 and then subtracted from 1.
- Set TurnTimer to Return value of Timer.

BP_CombatComponent - RequestTurn Function

On Function called
- Cast to BP_GameMode as Game Mode
- As BP_GameMode, call TurnRequest Function with input of UnitCharacter Actor reference.

BP_CombatComponent - StartUnitTurn Function
On Function called
  Set HasMoved Boolean to false
  Set HasAttacked Boolean to false
  If UnitCharacter Class is Child of BP_AllyUnitBase
    Create WBP_ActionMenu Widget for UnitCharacter.
    Set created Widget as ActionCommandWindow reference.
    Add ActionCommandWindow to Viewport.
  If not child of BP_AllyUnitBase
    Call AITakeTurn function of BP_UnitBase

BP_CombatComponent - EndUnitTurn Function
On Function called
  Call TurnEnded Notification
  Set IsTurn Boolean to false
  Call RequestTurn Function

WBP_ActionMenu - On Clicked (MoveCommand)
On Button clicked
  Attach Actor to TileSelector Actor reference
  Set visibility of self to Hidden
  Set Input Mode to Game and UI Only of Player Controller
  If HasAttacked Boolean of BP_CombatComponent is NOT true
    Set AttackCommand Button to Enabled.

WBP_ActionMenu - On Clicked (AttackCommand)
On Button clicked
  Set IsAttacking Boolean to true
  Set Actor Location of TileSelector to random actor from TargetEnemyList of BP_TileSelector as a teleport.
  Set IsInAttackMode of TileSelector to true.
  Set Input Mode to Game and UI Only of Player Controller
  Set visibility of self to Hidden

WBP_ActionMenu - On Clicked (WaitCommand)
On Button clicked
  Set visibility of self to Collapsed
  Call EndUnitTurn of Unit's BP_CombatComponent

BP_TileSelector - CheckTurnEnd Function
On Event called
  if HasMoved Boolean of TargetActor's BP_CombatComponent is true
    if HasAttacked Boolean of TargetActor's BP_CombatComponent is true
      Call EndUnitTurn of TargetActor's BP_CombatComponent

Summary and Explanation

BP_GameMode - BeginCombat Function

This function starts off the feature. When it is called, it runs through two For Loops to have the PartyUnits and EnemyUnits Arrays filled on BeginPlay each trigger their respective BeginBattle Function within their BP_CombatComponents. This simply awakens them and lets the Components know that a battle phase with turns has started. I find the advantage given to the player in this way is almost negligible keeping the fight fair, but also allows for more control of when turn based battle begins.

BP_GameMode - TurnRequest Function

The BP_CombatComponents will individually call this function according to their own timer's determined by their Initiative roll, explained later. This simple function simply adds each Unit to the TurnOrder Array in order of when they call it, and then triggers the StartTurn event, so the earlier a Unit calls this event, the earlier in the TurnOrder Array they will appear.

BP_GameMode - StartTurn Function

This function opens with a Do Once node, so it will only run for the first Unit in the TurnOrder Array at any time and can't have multiple simultaneous instances running, so that only one character can be taking a turn at a time. It then checks whether there is indeed a Unit at index 0 of the TurnOrder Array, and if so, triggers the Unit's BP_CombatComponent's StartUnitTurn Function. It then sets the IsTurn Boolean on the Component to true for any checks. It then sets the Vector CurrentActor to the Actor Location of the Unit currently taking its turn. It casts to the BP_TileSelector Actor and sets its location to the Unit taking its turn, making it clear to the player who they are currently meant to be controlling. It then Binds an Event to a Turn Ended Notification inside BP_CombatComponent with the signature ReadyNextTurn(), so that when that notification is triggered, ReadyNextTurn will start. It then removes the reference at index 0 of the TurnOrder Array so that another can take its place.
If the index of 0 is not found to be valid in the TurnOrder Array earlier in the function, it will call the ReadyNextTurn Event itself. The main purpose of this function is to ensure that only one character can be having a turn at any given time, and others will wait patiently for theirs to arrive by making use of the TurnOrder Array created in the TurnRequest Function.

BP_GameMode - ReadyNextTurn Function
This function is connected closely to the StartTurn function. It opens with a Sequence node to reset the Do Once node in the StartTurn function, opening the StartTurn function up to begin again. It then checks that the index 0 of TurnOrder has been replaced before launching the StartTurn function again. As the function name implies, it just readies the StartTurn function to run again properly, making sure the next unit in the TurnOrder array is ready to take their turn.

BP_CombatComponent - BeginBattle Function
BP_GameMode's BeginCombat function calls this one for each Unit involved in combat. This function handles the timing for each character for when they request a turn, determined by their Initiative roll. For this example, Initiative is determined by randomly getting a number between 0 and 20 for each character and then adding their dexterity modifier to that number, similar to Dungeons & Dragons. In this function, this number is then normalized between 0 and 25, returning a number between 0 and 1, which is then subtracted from 1 to determine the timing. By this logic, if a character scores a 20, they will trigger the RequestTurn function before a character that scores a 10. Finally, the function saves the return value of this timer as a handy variable to have on hand.

BP_CombatComponent - RequestTurn Function
This is a simple function where each Unit's BP_CombatComponent reaches out to the Game Mode to add themselves to the TurnOrder, determined by the previous function's timing. They provide a reference to themselves to be called when their turn arrives.

BP_CombatComponent - StartUnitTurn Function
The Game Mode calls this function within the StartTurn function, putting the ball back in the BP_CombatComponent's court when it's associated Unit's turn arrives. It starts by resetting the HasMoved and HasAttacked Booleans to false, indicating it has not yet done either of those this turn and opening the character up to do those actions. The function then checks if the character is a player unit, by checking if it is child of BP_AllyUnitBase, a parent of all player units. If not, the function will trigger a function to handle AI turn taking, not covered in this feature. If it is a player unit, this function will create the WBP_ActionMenu widget which is the interface for a player to take their turn. It saves the created widget as a variable to be referenced and then adds it to the player's viewport. From here, the player is able to interact with it to take their turn, which is elaborated on below.

BP_CombatComponent - EndUnitTurn Function
This function predictably ends a unit's turn. It triggers the TurnEnded notification which prompts the StartTurn function in the Game Mode to call the ReadyNextTurn function. It then sets its own IsTurn Boolean to false and triggers the RequestTurn function to re-add itself to the TurnOrder array, adding itself to the end of the turn queue.

WBP_ActionMenu - On Clicked (MoveCommand)
This function triggers when the player clicks on the 'Move' button in the ActionMenu widget. First, it attaches the selected Unit to the TileSelecter cursor Actor so it can be moved around. It then hides the ActionMenu widget and changes the Input Mode to Game and UI to allow the player to use the WASD keys to move the character around. Finally it resets the AttackCommand button if the player has not yet attacked, determined by the HasAttacked Boolean. This happens because the AttackCommand button is set to disable itself if there are no enemies within attack range, which can of course change when the character moves to a new position. The player is then free to move, and once their move is finished, the TileSelector is set to update the HasMoved Boolean to true at the end of the movement logic.

WBP_ActionMenu - On Clicked (AttackCommand)
Similarly, this function handles what happens when the player clicks on the Attack button in the ActionMenu widget. First it sets the IsAttacking Boolean to true and then it teleports the cursor to a random enemy within attacking range, a mechanic intended to improve user experience by expediting the attack process. It then changes the IsInAttackMode Boolean of the TileSelector to true, which is used within the TileSelector's attack handling logic. Finally, it does the same as the MoveCommand and sets the Input Mode to Game and UI so the player can use the WASD keys to move around if they don't want to attack the randomly selected target, or use the F key to confirm and select a target, and it hides the ActionMenu widget. Again, the TileSelector is set up to update the HasAttacked Boolean to true after attack logic has been completed and either re-enable the ActionMenu widget or trigger the CheckTurnEnd function below.

WBP_ActionMenu - On Clicked (WaitCommand)
This is the easiest button to set up, used when the player simply wants to skip their turn, or a portion of their turn. The player could chose to move and then wait without attacking, or attack without moving using the Wait button. This button only has to collapse the WBP_ActionMenu widget and then call the EndUnitTurn function explained above.

BP_TileSelector - CheckTurnEnd Function
This function is handled within the TileSelector which serves as the player-controlled cursor. After moving or attacking, it triggers this to check if both HasMoved and HasAttacked are true. If they are, it will automatically trigger the EndUnitTurn function without requiring further input from the player as logically there is nothing else to be done in that turn. At this point, the Unit's turn is completed and the ReadyNextTurn function can prepare the next unit in the TurnOrder array.