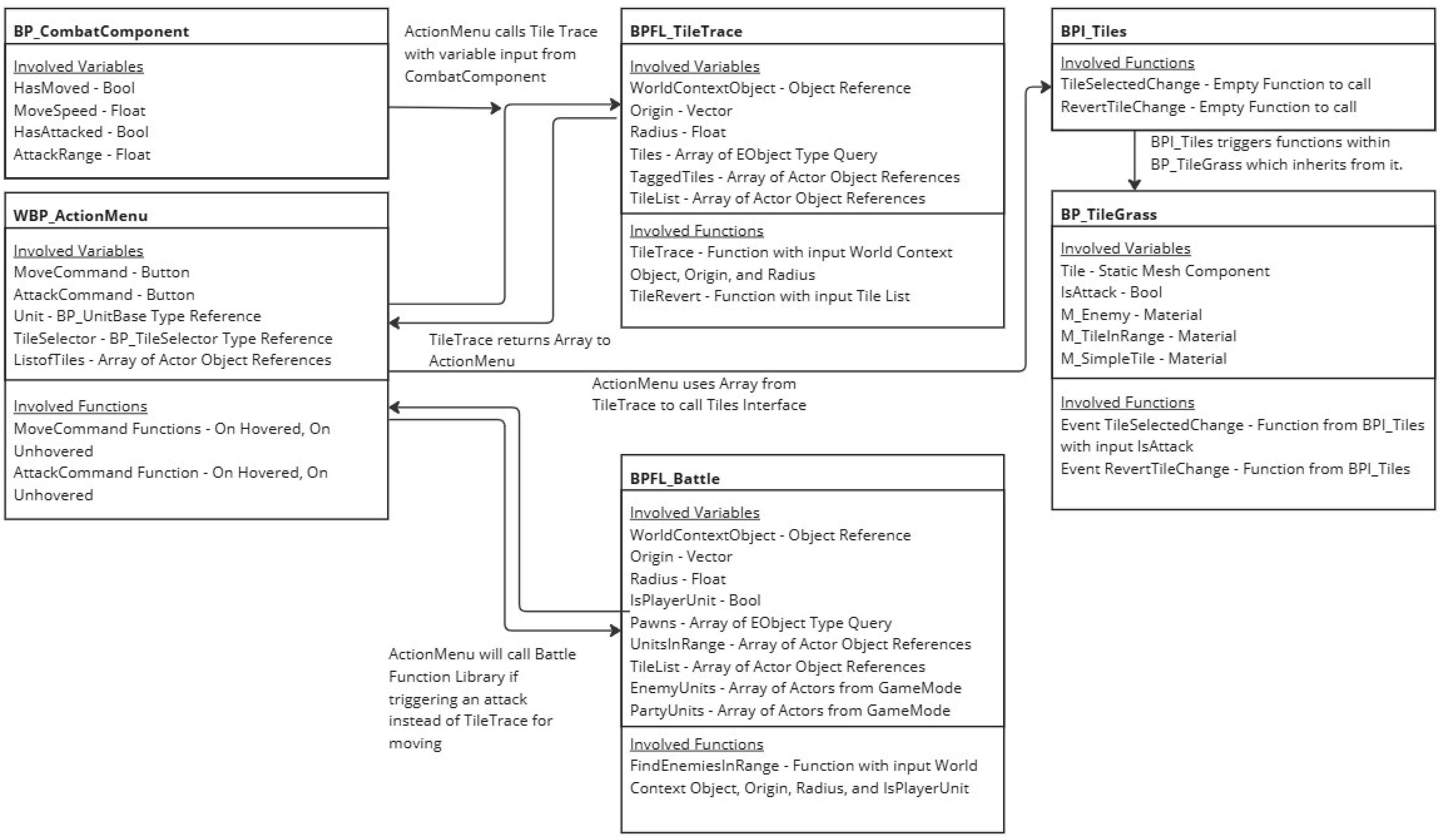
Feature - Finding Tiles within Move Range and Enemies within Attack Range
BP_TileGrass - Actor
BPI_Tiles - Blueprint Interface
BPFL_TileTrace - Blueprint Function Library
WBP_ActionMenu - Widget Blueprint
BP_CombatComponent - Actor Component



BP_CombatComponent

Involved Variables
HasMoved - Bool
MoveSpeed - Float
HasAttacked - Bool
AttackRange - Float

ActionMenu calls Tile Trace with variable input from CombatComponent

BPFL_TileTrace

Involved Variables
WorldContextObject - Object Reference
Origin - Vector
Radius - Float
Tiles - Array of EObject Type Query
TaggedTiles - Array of Actor Object References
TileList - Array of Actor Object References

Involved Functions
TileTrace - Function with input World Context Object, Origin, and Radius
TileRevert - Function with input Tile List

BPI_Tiles

Involved Functions
TileSelectedChange - Empty Function to call
RevertTileChange - Empty Function to call

BPI_Tiles triggers functions within BP_TileGrass which inherits from it.

WBP_ActionMenu

Involved Variables
MoveCommand - Button
AttackCommand - Button
Unit - BP_UnitBase Type Reference
TileSelector - BP_TileSelector Type Reference
ListofTiles - Array of Actor Object References

Involved Functions
MoveCommand Functions - On Hovered, On Unhovered
AttackCommand Function - On Hovered, On Unhovered

TileTrace returns Array to ActionMenu

ActionMenu uses Array from TileTrace to call Tiles Interface

BP_TileGrass

Involved Variables
Tile - Static Mesh Component
IsAttack - Bool
M_Enemy - Material
M_TileInRange - Material
M_SimpleTile - Material

Involved Functions
Event TileSelectedChange - Function from BPI_Tiles with input IsAttack
Event RevertTileChange - Function from BPI_Tiles

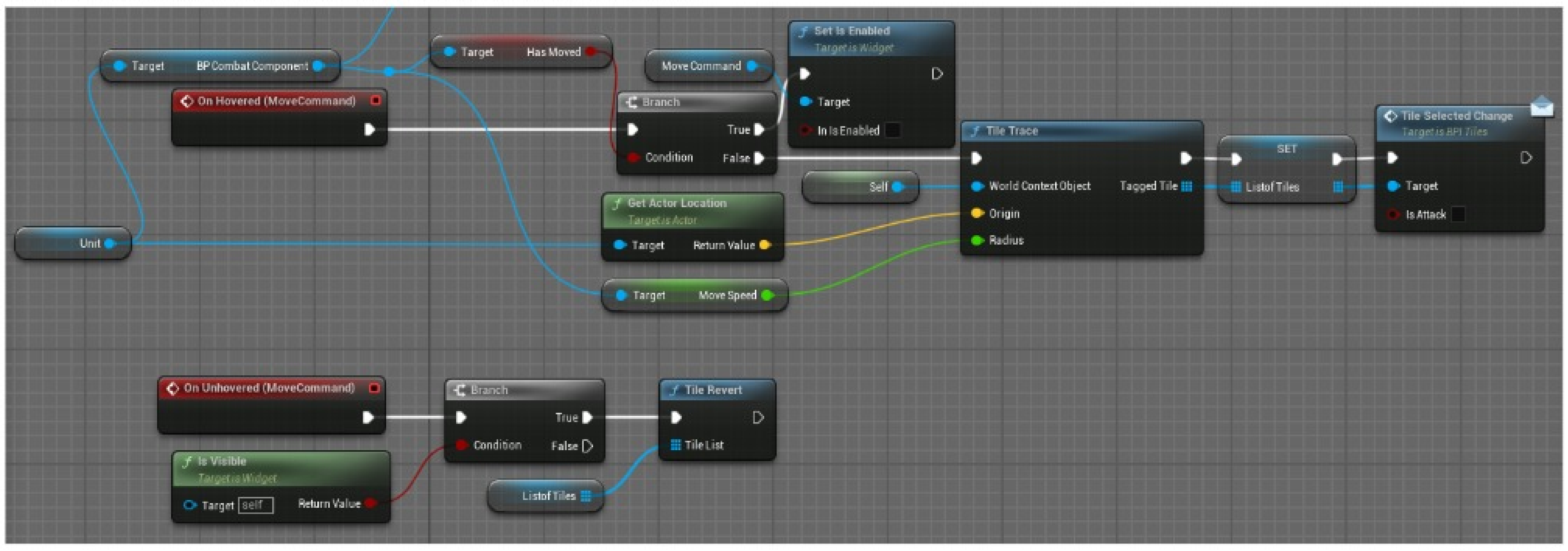
BPFL_Battle

Involved Variables
WorldContextObject - Object Reference
Origin - Vector
Radius - Float
IsPlayerUnit - Bool
Pawns - Array of EObject Type Query
UnitsInRange - Array of Actor Object References
TileList - Array of Actor Object References
EnemyUnits - Array of Actors from GameMode
PartyUnits - Array of Actors from GameMode

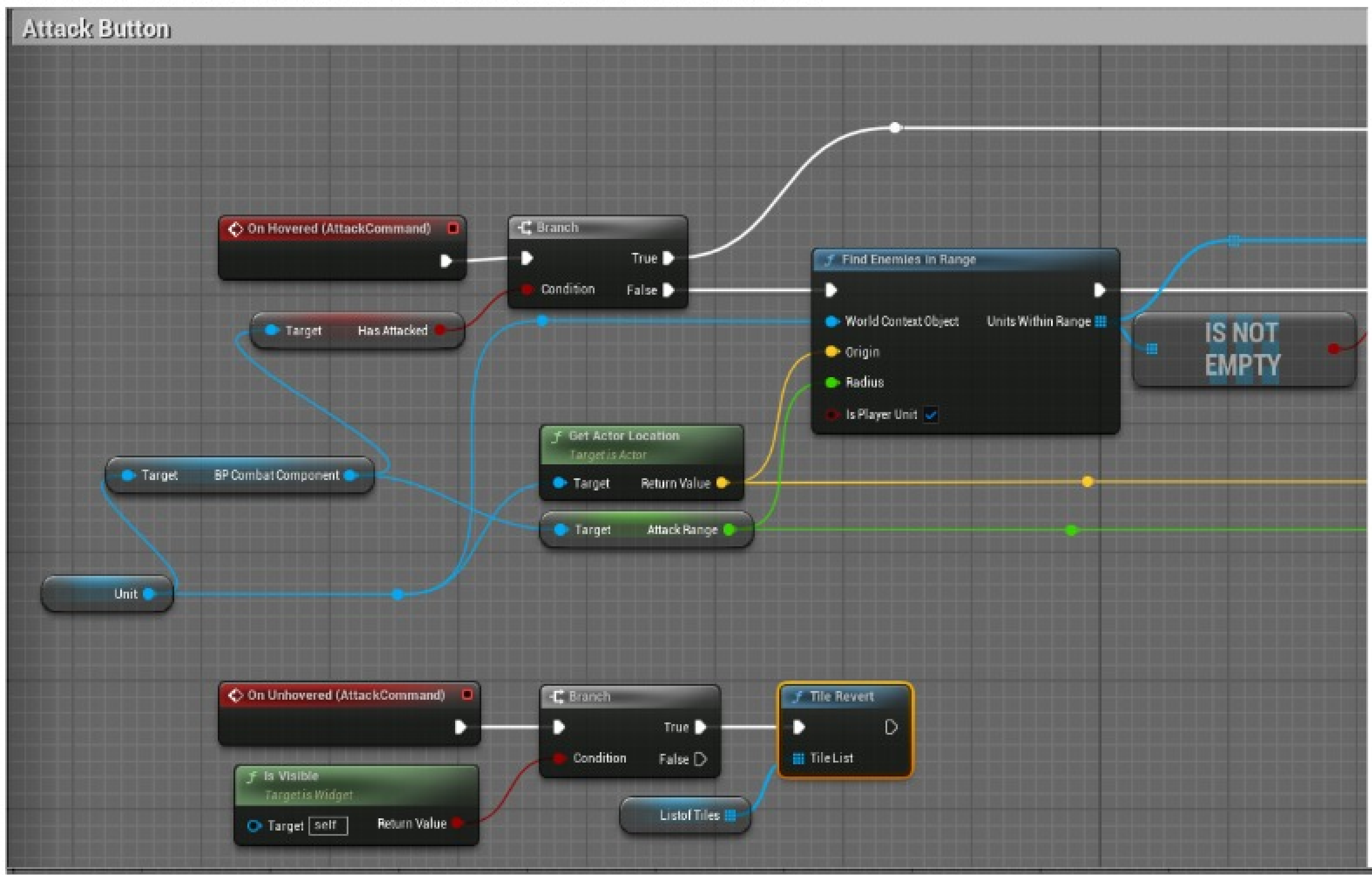
Involved Functions
FindEnemiesInRange - Function with input World Context Object, Origin, Radius, and IsPlayerUnit

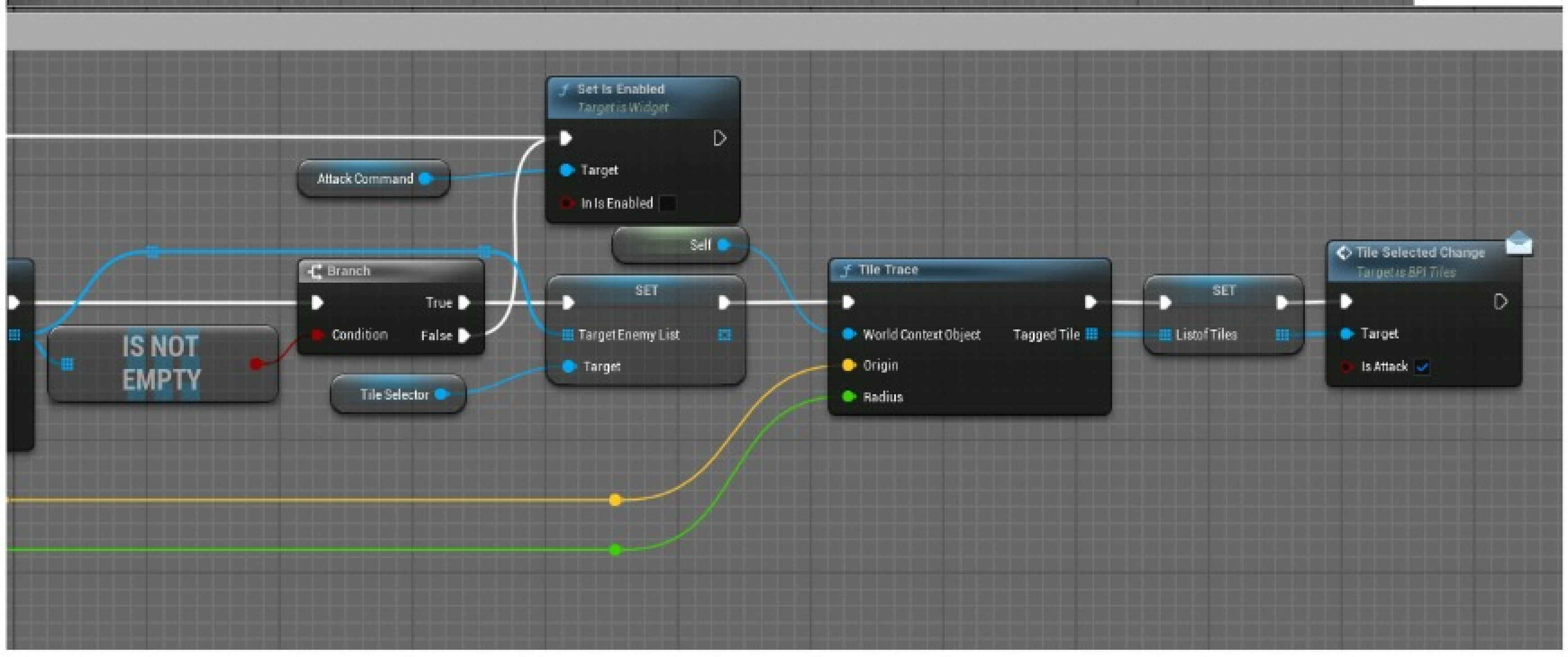
ActionMenu will call Battle Function Library if triggering an attack instead of TileTrace for moving

MoveCommand Functions - On Hovered and On Unhovered

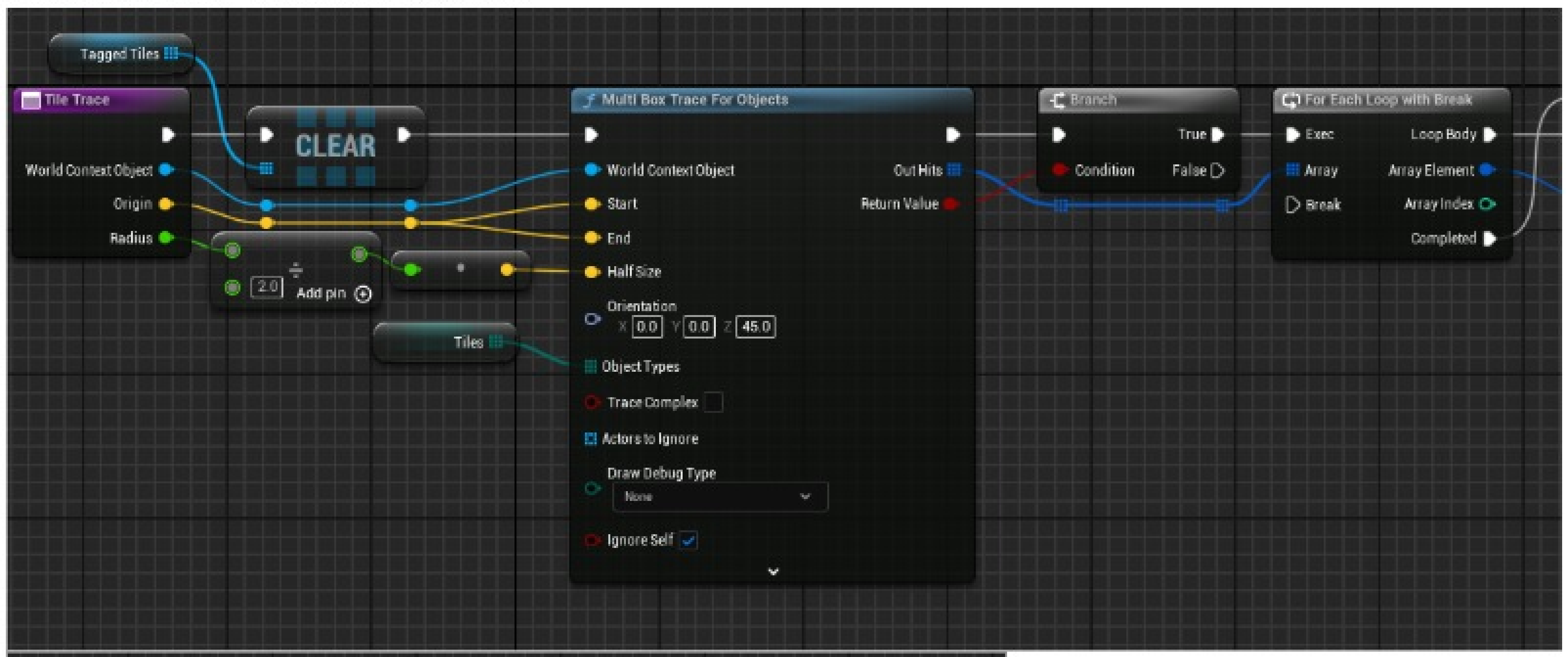


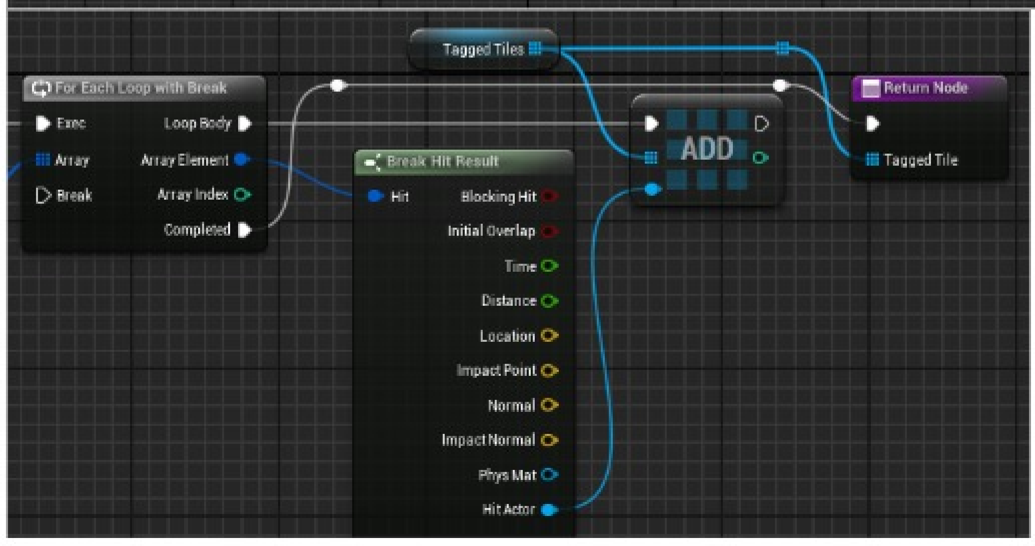
AttackCommand Functions - On Hovered and On Unhovered



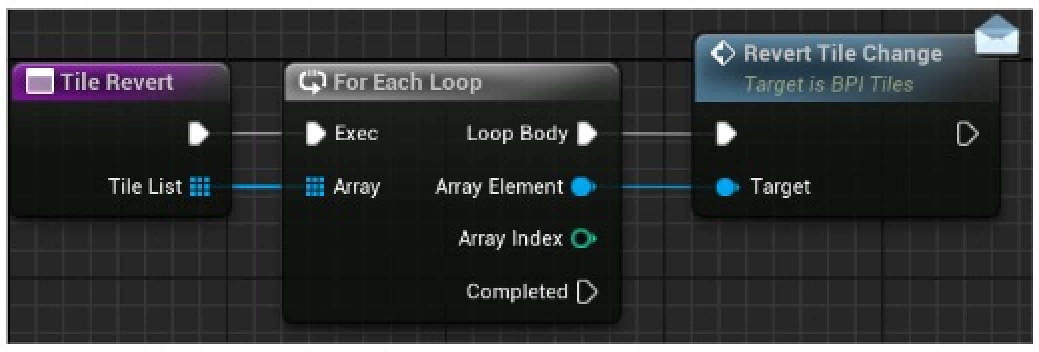


BPFL_TileTrace - TileTrace Function

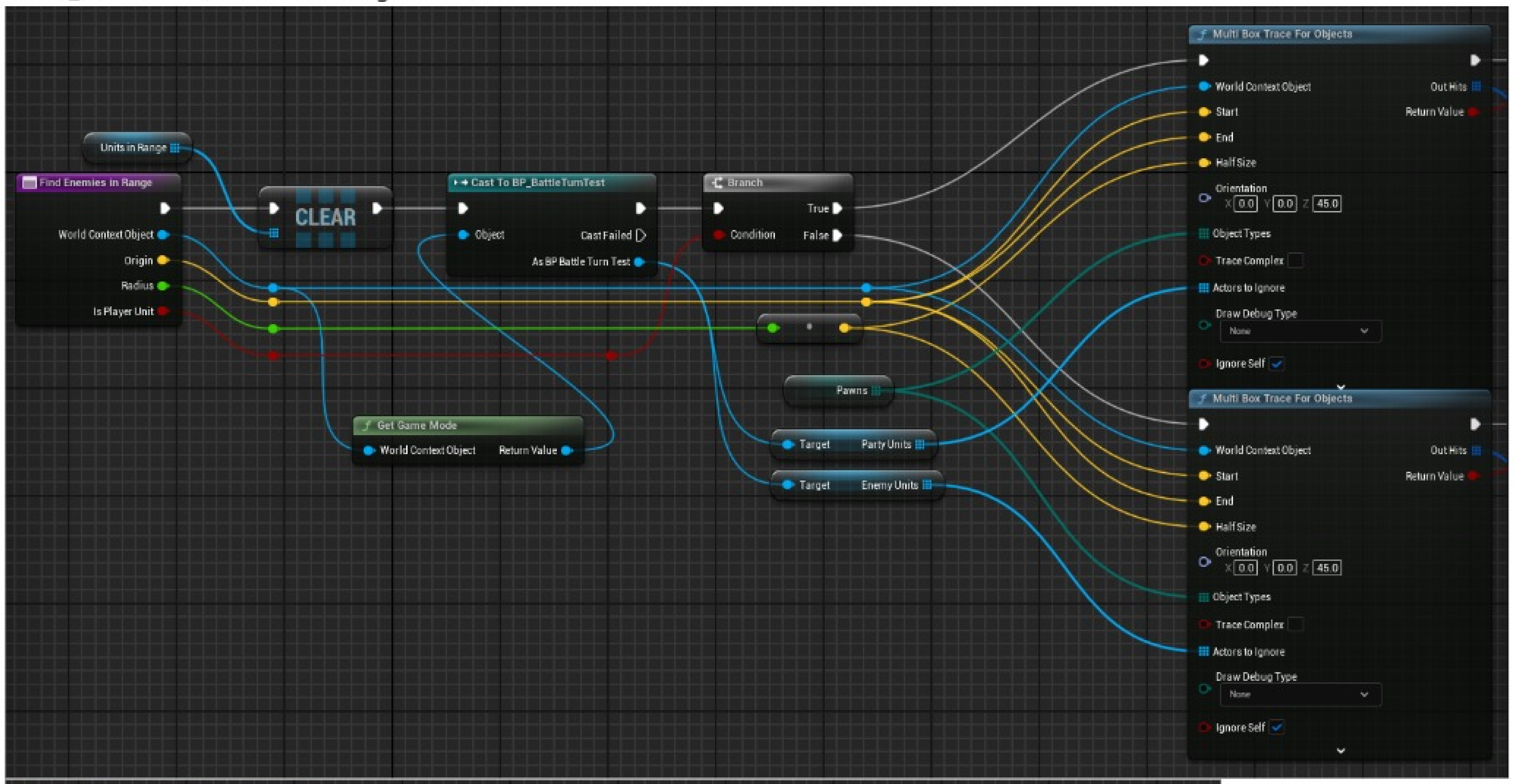


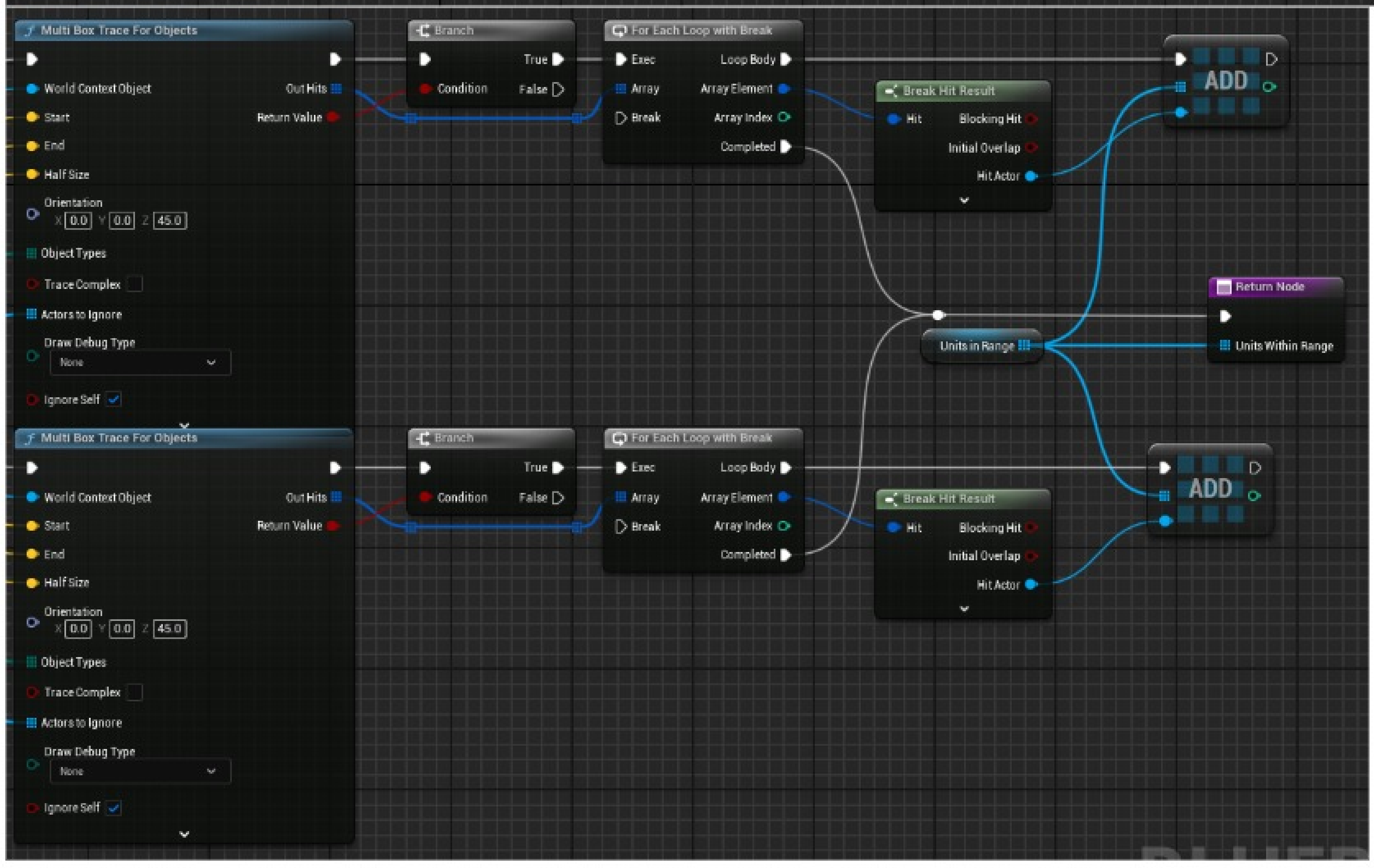


BPFL_TileTrace - TileRevert Function

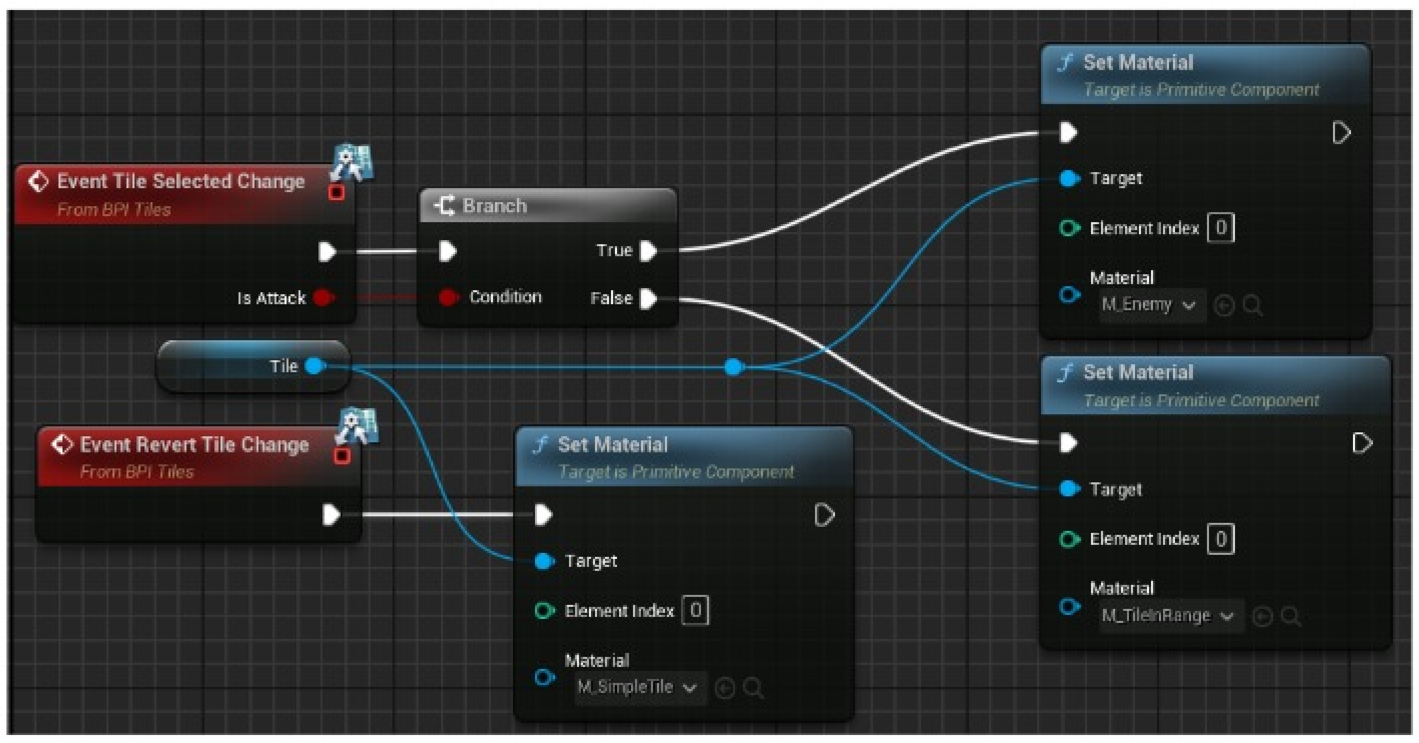


BPFL_Battle - FindEnemiesInRange Function





BP_TileGrass Functions - TileSelectedChange and RevertTileChange



Pseudo Code

```
BP_ActionMenu - On Hovered (MoveCommand)
When Button is hovered over
  If BP_CombatComponent Bool HasMoved is True
    Set is Enabled state of MoveCommand Button to False
  Else, call TileTrace function of BPFL_TileTrace with inputs of a reference to Self for WorldContextObject, Actor Location of Unit reference for Origin vector, and
.     MoveSpeed Float of BP_CombatComponent for Radius.
  Set Array ListofTiles equal to return value of TileTrace function.
  Call TileSelectedChange function of BPI_Tiles with input ListofTiles Array and IsAttack bool false.

BP_ActionMenu - On Unhovered (MoveCommand)
When the mouse moves from hovering over the Button
  If WBP_ActionMenu is currently visible on player's screen
    Call TileRevert function of BPFL_TileTrace with input ListofTiles Array.

BP_ActionMenu - On Hovered (AttackCommand)
When Button is hovered over
  If BP_CombatComponent Bool HasAttacked is True
    Set is Enabled state of AttackCommand Button to False
  Else call FindEnemiesInRange Function of BPFL_Battle with inputs of a reference to Self for WorldContextObject, Actor Location of Unit reference for Origin
.     vector, AttackRange Float of BP_CombatComponent for Radius, and Bool IsPlayerUnit is True.
  If the returned Array is empty;
    Set is Enabled state of AttackCommand Button to False
  Else set TargetEnemyList of TileSelector to the returned Array
  Call TileTrace Function of BPFL_TileTrace with inputs of a reference to Self for WorldContextObject, Actor Location of Unit reference for Origin vector, and
.     AttackRange Float of BP_CombatComponent for Radius.
  Set Array ListofTiles equal to return value of TileTrace function.
  Call TileSelectedChange function of BPI_Tiles with input ListofTiles Array and IsAttack bool true.

BP_ActionMenu - On Unhovered (AttackCommand)
When the mouse moves from hovering over the Button
  If WBP_ActionMenu is currently visible on player's screen
    Call TileRevert function of BPFL_TileTrace with input ListofTiles Array.

BPFL_TileTrace - TileTrace Function
When Function is called with Inputs WorldContextObject actor reference, Origin vector, and Radius float.
  Clear contents of Array TaggedTiles
  Trigger a Multi Box Trace for Objects with the Function Input WorldContextObject for World Context Object, the Input Origin Vector for the Start and End inputs,
  .     the Radius Input divided by 2 and converted to a Vector for the Half Size input, the Orientation shifted by 45 degrees on the Z axis, and the Tiles array in the
  .     Object Types input.
  If there is a Return Value
    Trigger a For Each Loop for each element returned by the Trace where the Hit Result is broken down to the Hit Actor and then added to the TaggedTiles Array.
  When completed, return the TaggedTiles Array.

BPFL_TileTrace - TileRevert Function
When Function is called with Input TileList Array
  Trigger a For Each Loop where each element of the Array calls BPI_Tiles RevertTileChange Function.

BPFL_Battle - FindEnemiesInRange Function
When Function is called with Inputs WorldContextObject actor reference, Origin vector, Radius float, and IsPlayerUnit Bool.
  Clear contents of UnitsInRange Array
  Cast to Game Mode obtained from the WorldContextObject actor reference.
  If IsPlayerUnit is True,
    Trigger a Multi Box Trace for Objects with the Function Input WorldContextObject for World Context Object, the Input Origin Vector for the Start and End
.       inputs, the Radius Input converted to a Vector for the Half Size input, the Orientation shifted by 45 degrees on the Z axis, the Pawns array in the Object
.       Types input, the PartyUnits Array obtained from the Game Mode in the Actors To Ignore slot, and Ignore Self set to True.
    If there is a Return Value
    Trigger a For Each Loop for each element returned by the Trace broken down to the Hit Actor and then added to the UnitsInRange Array.
    When completed, return the UnitsInRange Array.
  If IsPlayerUnit is False
    Trigger a Multi Box Trace for Objects with the Function Input WorldContextObject for World Context Object, the Input Origin Vector for the Start and End
.       inputs, the Radius Input converted to a Vector for the Half Size input, the Orientation shifted by 45 degrees on the Z axis, the Pawns array in the Object
.       Types input, the EnemyUnits Array obtained from the Game Mode in the Actors To Ignore slot, and Ignore Self set to True.
    If there is a Return Value
    Trigger a For Each Loop for each element returned by the Trace broken down to the Hit Actor and then added to the UnitsInRange Array.
    When completed, return the UnitsInRange Array.
```

BP_TileGrass - TileSelectedChange Function

```
When BPI_Tiles TileSelectedChange is triggered, this event begins
   If IsAttack is true
      Set the Material of Static Mesh Component Tile to M_Enemy
   Else set the Material of Static Mesh Component Tile to M_TileInRange
```

BP_TileGrass - RevertTileChange Function

```
When BPI_Tiles RevertTileChange is triggered, this event begins
   Set the Material of Static Mesh Component Tile to M_SimpleTile
```

Summary and Explanation

BP_ActionMenu - On Hovered (MoveCommand)

This function controls what happened when the mouse or cursor hovers over the Move button in the Action Menu. The Action Menu is the window set up to prompt a player's options during a turn, whether to move or attack in this context. The goal is to have the selected character's movement range displayed while the button is hovered over, unless the character has already moved, by changing the mesh material colour from a default green to either a highlighted blue or vibrant red to show either movement range or attack range. It checks this by first checking for the condition 'HasMoved'. If this is true, it will simply disable the button as the character has already moved in this turn.
If the character has not yet moved this turn, the function with call the TileTrace function of the function library BPFL_TileTrace, which will be explained in its own section. This function will return an array, ListofTiles, which contains the tiles the character can move to within their MoveSpeed range. The function then provides this array to the BPI_Tiles TileSelectedChange function, specifying that IsAttack is false as this determines the colour the tiles will change to. This is where this function ends, doing little besides receiving the initial input and tying the other functions together to complete the task. The actual movement logic would only take place once the button is clicked, which would have its own function. Since the tile colouring logic is handled within the OnHovered function, there is no need to repeat it when the button is clicked.

BP_ActionMenu - On Unhovered (MoveCommand)

This function does the exact opposite of the one above, but is much simpler as the necessary info is provided by the On Hovered function; it must have been hovered over in order to be moved off of. The goal is to reset the tiles' colour to their original green instead of leaving them showing move range or attack range. Once the function detects that the button has been moved from, ensuring that the Action Menu is indeed visible to ensure it does not reset when not intended, it will call the TileRevert function from the BPFL_TileTrace using the ListofTiles developed in the On Hovered function.

BP_ActionMenu - On Hovered (AttackCommand)

The AttackCommand button functions in almost the exact same way as the MoveCommand button, but instead of showing the character's movement range, it displays their attack range. For melee characters, this will simply be adjacent tiles as they will have an attack range of 1, but ranged characters could have much more. The other main difference here is that the button quickly calls the FindEnemiesInRange function of BPFL_Battle, which will be explained in its own section, but essentially returns an array of enemy units that could be attacked. If there are none within range, the button is disabled, though this can be reset by moving if the player has not yet moved. The function then moves through the same steps as the MoveCommand button, with the small difference of setting the IsAttack bool to true when calling the TileSelectedChange function so that the tiles will be coloured red instead of blue. Again, since the tile-changing logic is handled within the OnHovered function, there is no need to repeat it when the Button is clicked, leaving the OnClicked function free to handle the actual attacking turn logic.

BP_ActionMenu - On Unhovered (AttackCommand)

Again, this function simply resets the tile colours in exactly the same way as the MoveCommand. Using the array created in the On Hovered function, it resets the colours through the TileRevert function.

BPFL_TileTrace - TileTrace Function

This function within the TileTrace function library handles the magic of detecting what tiles are within the character's movement range. The BP_ActionMenu provides the input information of the Unit conducting the search, their position to determine the origin location of the trace, and their personal MoveSpeed value to determine the radius. First it clears any existing information held within the TaggedTiles array. Then, using the input information it was triggered with, it calls for a Multi Box Trace for Objects. This handy function triggers an invisible box being cast around the specified origin point with a specified radius. I shifted the orientation by 45 degrees on the Z axis so that it's corners are above, below, to the left and right of the character rather than a flat square. The Tiles array refers to all Tile nodes that a player can move to using physics Object Types like WorldDynamic, WorldStatic, Pawn, etc., though in this case a special one is made to house navigable Tiles.
This trace then returns an output which is broken down to return only hit actors of the type specified in the Tiles array, meaning every tile contained within this invisible box. The function then takes these Hit Actor Object References and adds them to the now cleared TaggedTiles array, which is then returned as the output of the function.

BPFL_TileTrace - TileRevert Function

This function is quite a bit simpler and just uses an array of tiles supplied by its input and for each item in the array, triggers the RevertTileChange function, which will then handle the actual changing of their materials back to the default green. This array of tiles will be exactly the same as the last one supplied by the TileTrace function above, so no new search needs to take place, and the last tiles changed will simply be restored.

BPFL_Battle - FindEnemiesInRange Function

This functions works in a similar manner to the TileTrace function, but instead of searching for an array of navigable tiles it searches for attackable enemies within range. The function needs to be fed the same input, the WorldContextObject, the Origin vector, and the Radius float which is provided by the AttackRange float instead of MoveSpeed, and this function also needs a bool to determine whether it is a player unit calling the function or an enemy unit, as this changes what the function searches for. It starts by clearing any values contained within the UnitsInRange Array so that it can be repopulated later. It then casts to the Game Mode where we have essential information stored, particularly the Arrays EnemyUnits and PartyUnits. Here it then branches depending on the value of the Bool IsPlayerUnit. If this is true, it enters a box trace similar to the one in the TileTrace function, with the WorldContextObject, Start, End, and Half Size all provided by the function inputs. The Orientation is again shifted by 45 degrees along the Z axis. Instead of a Tiles array, we now insert a Pawns array into the Object Type which will search for all characters, but we narrow this down by inputting the PartyUnits array from the Game Mode into the Objects To Ignore field, so that the trace will only detect enemy units. The function then wraps up in exactly the same way as the TileTrace function; checking that something was found, breaking the hit result to get the hit Actors, adding them to the UnitsInRange array, and then returning that Array at the end.
If the IsPlayerUnit bool at the start is false, the only key difference is in what the trace is told to ignore, replacing the PartyUnits Array with the EnemyUnits Array. The function then returns this Array to the ActionMenu in this case to inform whether there are any attackable entities within range. If there are not, the attack button is disabled and greyed out. This function could also be used in setting up an enemy AI, creating area of effect abilities that either heal or damage within an area, or creating aura effects emanating from certain characters.

BP_TileGrass - TileSelectedChange Function

These last two functions handle the actual colour changing to display the move and attack ranges with minimal logic as most of that is handled before these are even called. The only check that happens here is with an input Bool IsAttack, which will only change which colour the function uses. For this example, we have simple Materials, a default green to look like grass, a light blue to highlight, and a vibrant red. If the IsAttack Bool is true, the function will change the material of the Tile Mesh to be red, marking the character's attack range in red. Otherwise it will change the Mesh to blue, highlighting where the selected character can move to. Since this function is called within a For Loop, it will be called instantly for each tile contained within the TaggedTiles Array determined by the TileTrace function.

BP_TileGrass - RevertTileChange Function

Lastly, this very simple function serves only to restore the tile's default material when the tiles should no longer be displaying a different colour. It sets the Material for the Tile Mesh to the default green, and again, since it is called within a For Loop, every Tile that was changed by the TileSelectedChange function will be targeted.