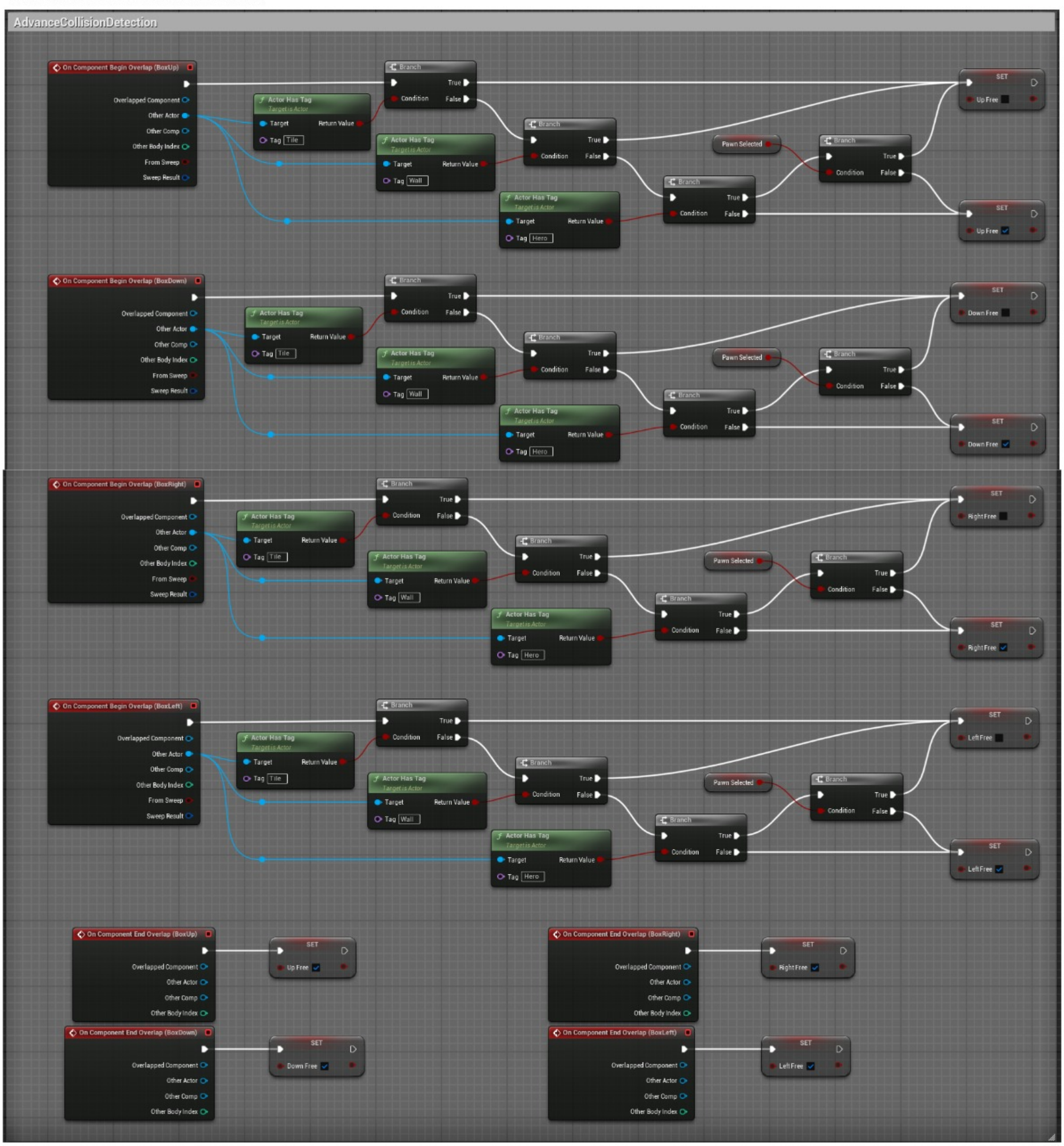
Tile Selector - Actor
Feature - Tile Movement (WASD and Mouse controls)

BP_TileSelector

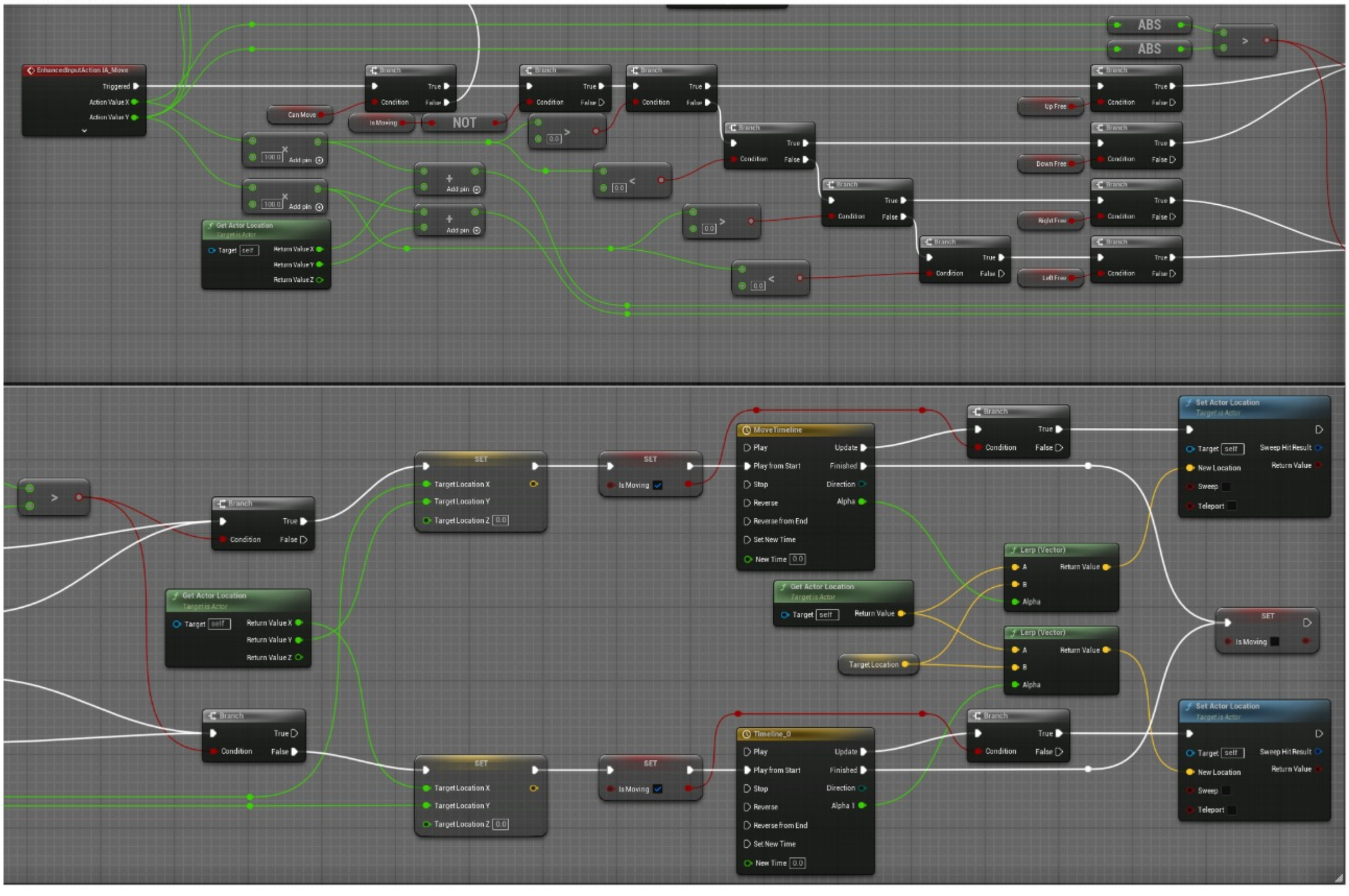
Involved Variables
CanMove - Bool
IsMoving - Bool
PawnSelected - Bool
UpFree/DownFree/LeftFree/RightFree - Bools
TargetLocation - Vector

Involved Functions
Advance Collision Detection - Event On Component Begin Overlap
Movement - Event EnhancedInputAction IA_Move
Get Cursor Location - Event Custom
Mouse Use - Event EnhancedInputAction IA_Select

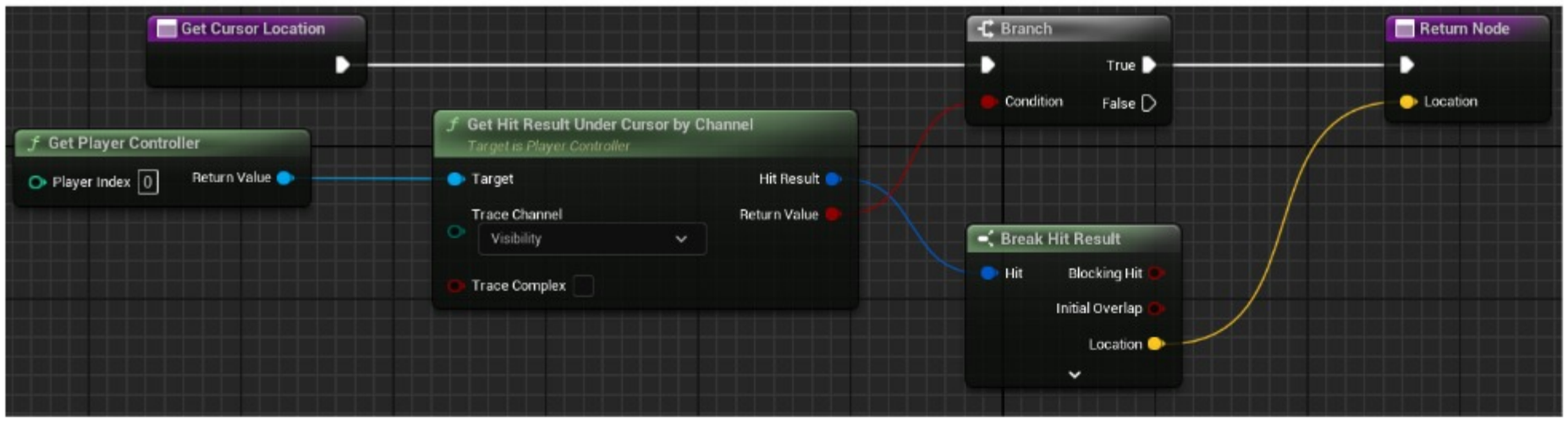
Advance Collision Detection Function



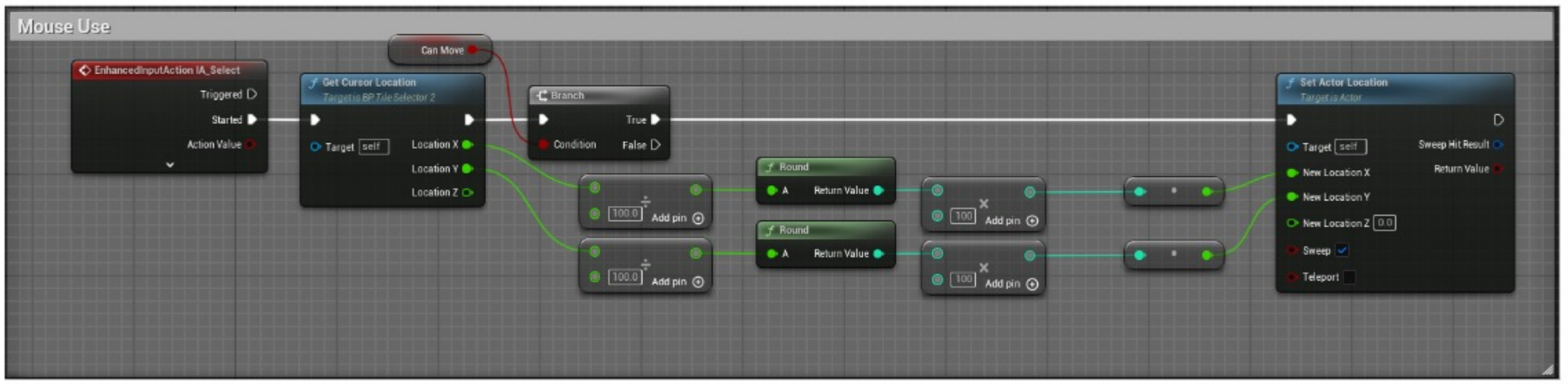
Movement Function



Get Cursor Location Function



Mouse Use Function



Pseudo Code

Advance Collision Detection

When Box Collider component BoxUp begins overlapping
  If other actor has tag 'Tile', set bool UpFree to False.
  Else if other actor has tag 'Wall', set bool UpFree to False.
  Else if other actor has tag 'Hero'
    if True, check if bool PawnSelected is True.
      if True, set bool UpFree to False.
      else set bool UpFree to True.
    Else set bool UpFree to True.

When Box Collider component BoxUp ends overlap
  Set bool UpFree to True.

Repeat both steps for Box Colliders BoxDown, BoxLeft, and BoxRight using bools DownFree, LeftFree, and RightFree respectively.

Movement

When Input Action IA_Move (WASD) is triggered
  If bool CanMove is True AND bool IsMoving is NOT True
    If Action Value X of IA_Move is greater than 0 AND bool UpFree is True, OR Action Value X is less than 0 AND bool DownFree is True
      If the Absolute value of Action Value X is greater than the Absolute value of Action Value Y
        Set Vector variable TargetLocation X value is equal to Action Value X * 100 + Actor Location X value
        Set Vector variable TargetLocation Y value is equal to Actor Location Y value
        Set bool IsMoving to True
        Enter Timeline with duration 0.2 seconds and float value Alpha of 0 to 1.
          On Update, if bool IsMoving is True.
            Set Actor Location to Vector Lerp between Actor Location and vector TargetLocation with Alpha value equal to Timeline Alpha float.
          On Finished, set bool IsMoving to False.
    If Action Value Y of IA_Move is greater than 0 AND bool RightFree is True, OR Action Value Y is less than 0 AND bool LeftFree is True
      If the Absolute value of Action Value X is NOT greater than the Absolute Value of Action Value Y
        Set Vector variable TargetLocation X value is equal to Actor Location X value
        Set Vector variable TargetLocation Y value is equal to Action Value Y * 100 + Actor Location Y value
        Set bool IsMoving to True
        Enter Timeline with duration 0.2 seconds and float value Alpha of 0 to 1.
          On Update, if bool IsMoving is True.
            Set Actor Location to Vector Lerp between Actor Location and vector TargetLocation with Alpha value equal to Timeline Alpha float.
          On Finished, set bool IsMoving to False.

Get Cursor Location

On custom Event called
  If get Hit Result under Cursor by Trace Channel Visibility with Target is Player Controller
    Break Hit Result and Return Location Vector value.

Mouse Use

When Input Action IA_Select is Started
  Call Get Cursor Location Function
  If bool CanMove is True
    Set Actor Location with values:
      New Location X value equal to Get Cursor Location returned Location X value / 100 rounded to nearest integer and then multiplied by 100 and converted to float value.
      New Location Y value equal to Get Cursor Location returned Location Y value / 100 rounded to nearest integer and then multiplied by 100 and converted to float value.

Summary and Explanation

Advance Collision Detection

This function relies on the Box Collider components extruding from the Actor, named BoxUp, BoxDown, BoxLeft, and BoxRight. These colliders act as feelers reaching into adjacent tiles to detect other colliders. This function checks for any tags the other actor might have and then returns whether that direction is free to move into. Actors with tags 'Wall' or 'Tile' will always been an obstruction, whereas 'Hero' will only be an obstruction if PawnSelected is True, meaning the controlled Actor is carrying another actor. If there is no obstruction of this kind, the adjacent tile must be free to move into.

Movement

This function handles the actual movement mechanic. First it checks that the Tile Selector Actor can move with the CanMove bool. It then checks that it is not already moving with the IsMoving bool. This is to prevent the player from moving midway through an ongoing movement, disrupting the movement distance and possible setting it out of sync with the Tiles.

Here the function splits into two halves, one handling X-axis movement and the other Y. It goes through this by checking whether the values for either axis are greater than or less than 0. If the player presses the W key for example, it would provide a value of 1, whereas S would return -1. It then does checks for whether the adjacent spot is free, using the bools returned from the Advance Collision Detection function; UpFree, DownFree, RightFree and LeftFree. If the bool is positive, movement in that direction can happen.

After the bool checks, the function checks whether the absolute value of X is greater than or less than Y. This is to prevent the player from pressing both at the same time and moving diagonally. As long as one is greater than the other, the function will then set the TargetLocation vector variable using the Tile Selector Actor's current location for either the X or Y value depending on which axis we are moving along, and the pressed key's Action Value (-1 or 1) multiplied by 100 (the length of a tile), added to the actor's current location to add movement. Once this is set, the function sets IsMoving to True to prevent any further inputs until movement is completed.

Movement (Cont.)
It then enters a Timeline 0.2 seconds long, which is then the duration of a movement action, and with a float value named Alpha that is from 0 to 1. On Update, which happens every tick, the function confirms that IsMoving is still True and updates the actor's location, using a vector Lerp to transition from the current location to the set TargetLocation variable. With the Alpha float value from the Timeline that increases proportionately to the elapsed time, it transitions to reach the TargetLocation when the Timeline is finished and Alpha is 1. This gives the movement a quick but noticeable animation so that the Tile Selector actor slides to its new location rather than jumping or teleporting. At this point the actor has moved to the intended position and the function sets the IsMoving bool to False and allow the player to move again.

Get Cursor Location
The purpose of this function is solely to translate the mouse position into a useable vector value. It is a custom event called only at the beginning of the Mouse Use function. When called, it gets a hit result from the mouse location and translates it into a world position. The function then ends by returning this value to be used in the Mouse Use function.

Mouse Use
This function handles using the mouse to move the Tile Selector actor around, so that the player can click on a tile to instantly move there rather than moving tile by tile using WASD. It achieves this by first calling the custom Get Cursor Location function which returns a vector value. It then confirms whether the CanMove bool is True and it is allowed to move. If this is True, it directly sets the actor's new location using the X and Y values returned by the Get Cursor Location. These values are divided by 100 and then rounded to the nearest integer, before being multiplied again. This is so that the value places it neatly inside a tile, it must end in '00'. For example, if the Get Cursor Location returned an X value of 234 and a Y of 568, the outcome would be rounded to 200 and 600, snapping it to a tile.