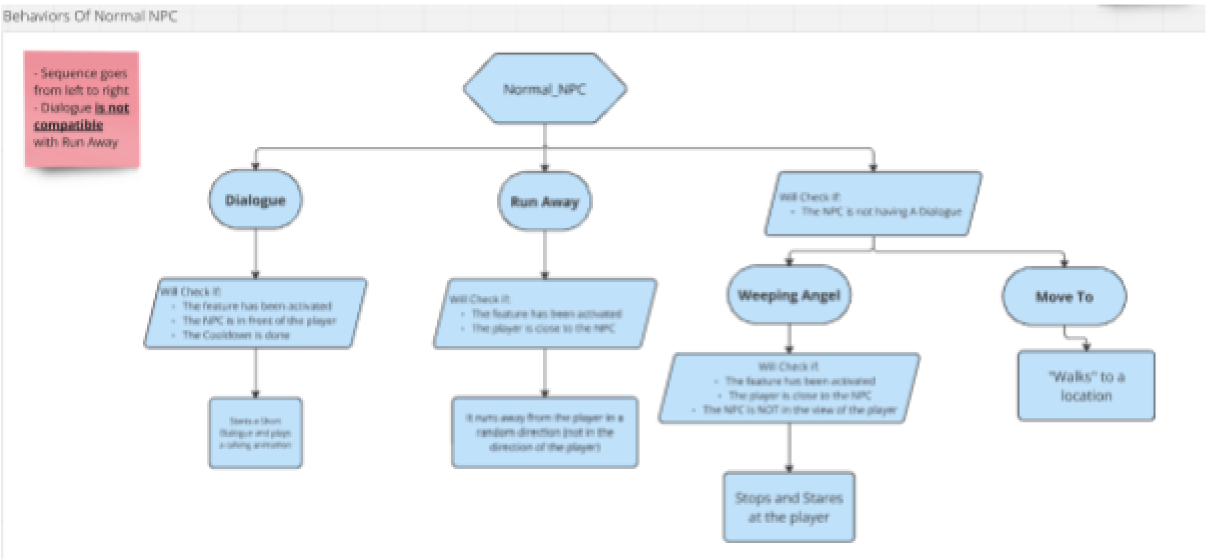
NORMAL NPC:

By Brandon Retana
Discord: brandon_lii

Description:

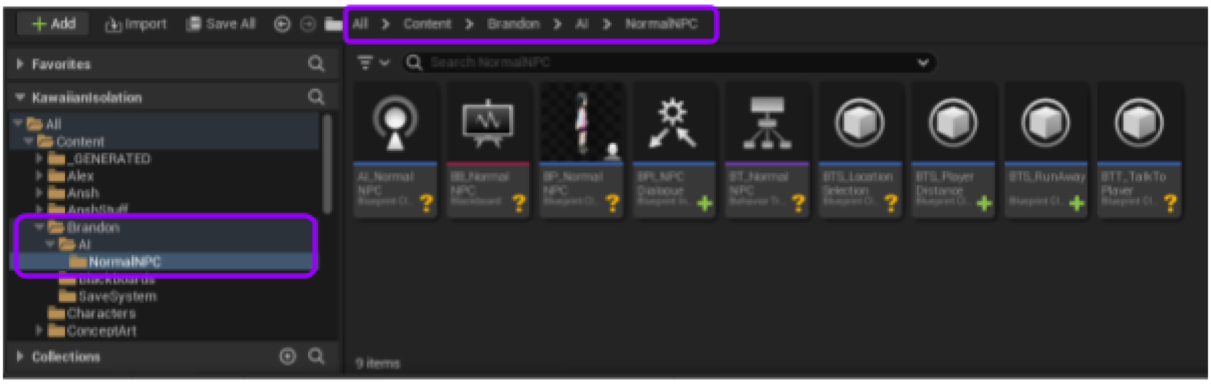
This AI has four different states it can be in; Move To, Run Away, Dialogue, and Weeping Angel. The base (will run no matter what) behavior is Move To which makes the NPC "walk" to a location. The other three behaviors can be selected from the details panel or directly from BluePrints. Run Away: makes the NPC run away from the player in a random way. Dialogue: starts a small random conversation with the player. Weeping Angel: makes the NPC stare at the player when the player is not looking, if the player is looking the NPC will resume its path.

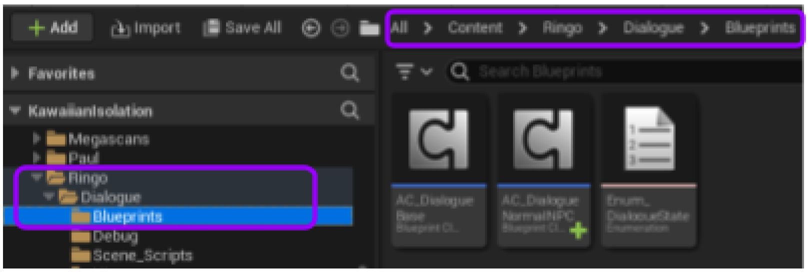

Behaviors Of Normal NPC
- Sequence goes from left to right
- Dialogue is not compatible with Run Away
Normal_NPC
Dialogue
Run Away
Will Check if:
- The NPC is not having A Dialogue
Will Check if:
- The feature has been activated
- The NPC is in front of the player
- The Cooldown is done
Will Check if:
- The feature has been activated
- The player is close to the NPC
Weeping Angel
Move To
Will Check if:
- The feature has been activated
- The player is close to the NPC
- The NPC is NOT in the view of the player
"Walks" to a location
Starts a Short Dialogue and plays a talking animation
It runs away from the player in a random direction (not in the direction of the player)
Stops and Stares at the player

Snapshot of what the behaviours do and how they get selected for the Normal NPC

Location Of Files:

Dialogue Component is in the Dialogue folder but can be moved later.



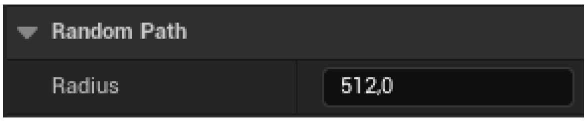


How to Use/Set up the Normal NPC:

The Character comes with many features that can be activated and deactivated from the details panel or during runtime using the helper methods shown above. To start the setup (as of now, might change once blackboards and save systems are implemented) click on BP_NormalNPC and scroll down to select the desired behaviors and set up their options.

Random Path:

By default the NPC will walk to random locations using the "Radius" option under the Random Path section to set the maximum distance these locations can be set up to.



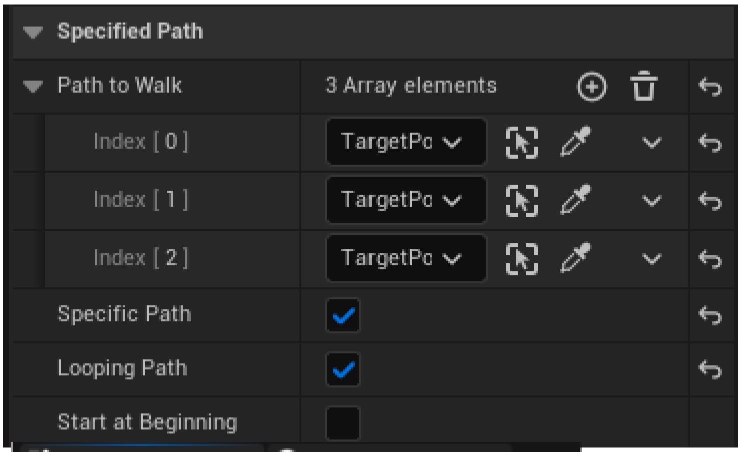
Random Path

Radius: 512,0

Specified Path:

If you want a specified path you have to add TargetPoints to the "Path to Walk" array, these are locations the NPC will travel to. Then click on the "Specific Path" checkbox, this tells the AI you want to enable the specific path behavior.

By default the AI will move to random locations once all the locations have been completed, you can check the "Looping Path" feature to make the AI go back and forth, meaning that once the AI goes to the last location in the array it will go backwards until it gets to the first location and again to the last location in an infinite loop. This is best if you want the NPC to go from point A to B to C then to B to A to B to C cyclically and avoid random paths. The other feature is "Start at Beginning" which makes the AI go to the first location in the array once it has completed the last location in the array, meaning it goes from point A to B to C then to A to B to C etc. This is best for making the NPC walk in some sort of circle.



Specified Path

Path to Walk: 3 Array elements

Index [0]: TargetPo

Index [1]: TargetPo

Index [2]: TargetPo

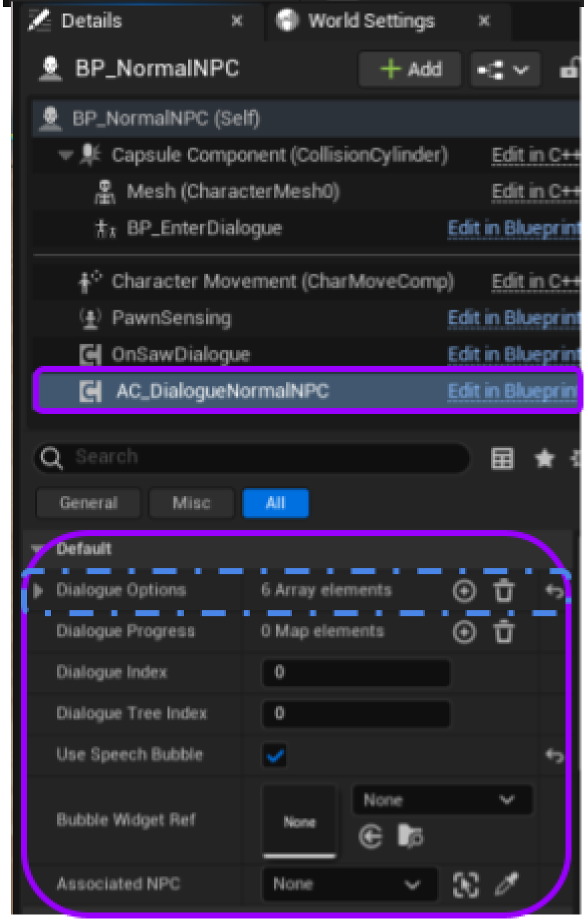
Specific Path: ☑

Looping Path: ☑

Start at Beginning: □

Dialogue:

The "Speech Bubble" feature is not ready yet! Check the "Can Talk" checkbox to enable the NPC to have conversations with the Player. The "End Dialogue Delay" is the delay time we want for a conversation to end, I recommend 3 as default. Then on the details panel scroll down to select the AC_DialogueNormalNPC then go to the Default section and under Dialogue Options, open the array and insert the dialogue lines that you want the NPC to use during a dialogue. Each element is a different dialogue event.Also, the Dialogue feature and the RunAway Feature ARE NOT COMPATIBLE DO NOT ENABLE BOTH BEHAVIORS AT THE SAME TIME.



BP_NormalNPC (Self)

Capsule Component (CollisionCylinder) Edit in C++

Mesh (CharacterMesh0) Edit in C++

BP_EnterDialogue Edit in Blueprint

Character Movement (CharMoveComp) Edit in C++

PawnSensing Edit in Blueprint

OnSawDialogue Edit in Blueprint

AC_DialogueNormalNPC Edit in Blueprint

Default

Dialogue Options: 6 Array elements

Dialogue Progress: 0 Map elements

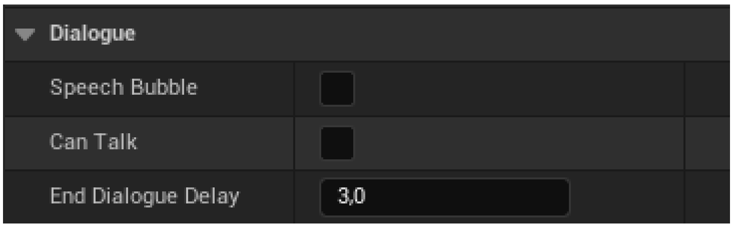
Dialogue Index: 0

Dialogue Tree Index: 0

Use Speech Bubble: ☑

Bubble Widget Ref: None

Associated NPC: None

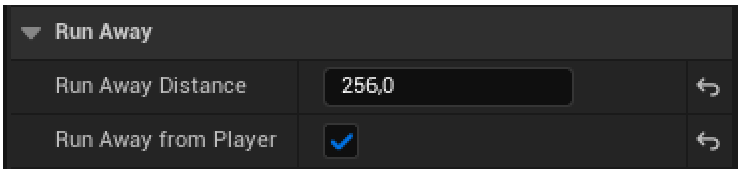


Dialogue

Speech Bubble: □

Can Talk: □

End Dialogue Delay: 3,0

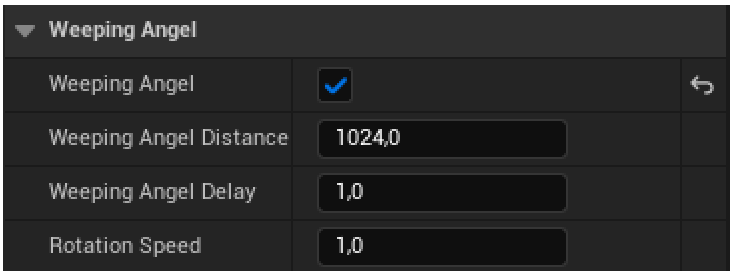


Run Away

Run Away Distance 256,0

Run Away from Player ☑

RunAway:
If you want the NPC to run away from the player click on the “Run Away From Player” checkbox, the “Run Away Distance” is the minimum distance between the NPC and Player for the NPC to start running away, I recommend 256. Also, the Dialogue feature and the RunAway Feature ARE NOT COMPATIBLE DO NOT ENABLE BOTH BEHAVIORS AT THE SAME TIME.



Weeping Angel

Weeping Angel ☑

Weeping Angel Distance 1024,0

Weeping Angel Delay 1,0

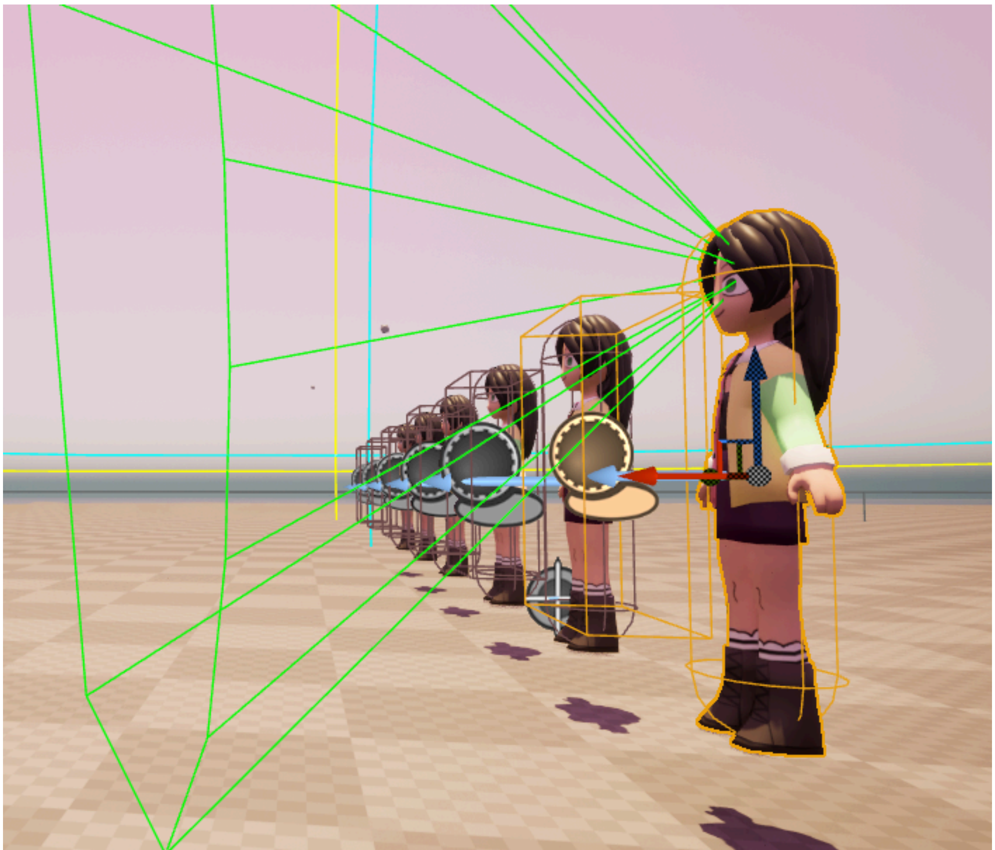
Rotation Speed 1,0

Weeping Angel:
The “Weeping Angel” checkbox is to enable the Weeping Angel feature. The “Weeping Angel Distance” is the maximum distance at which the Weeping Angel behavior will get activated. The “Weeping Angel Delay” is a small delay (in seconds) it takes an NPC to stop staring at the player. Finally, the “Rotation Speed” is how fast the NPC will turn to face the player.

Also, don’t forget to place a NavMesh Bounds object to set the bounds of your NPC!

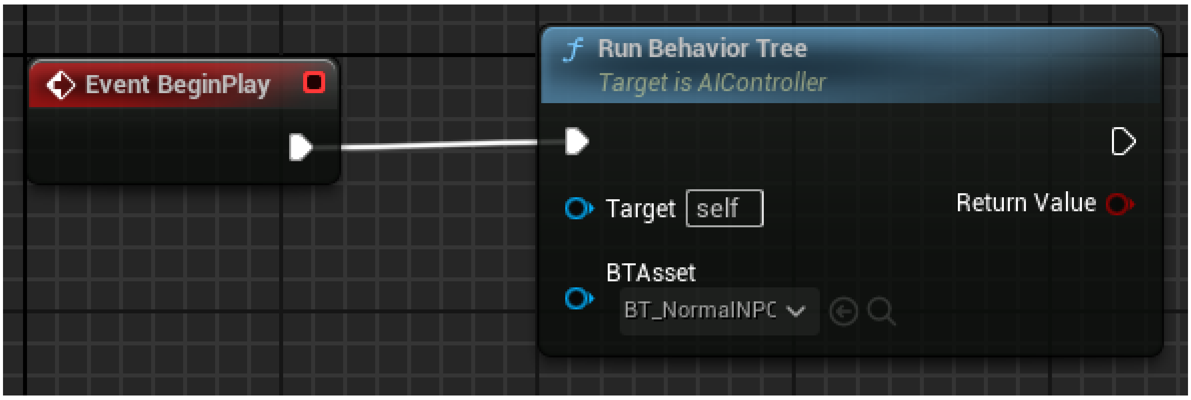
In the following section, there is an explanation of helper methods that need to be used to enable or disable the Dialogue, Weeping Angel, and RunAway behaviors from other blueprints. The helper methods are: Update Can Talk, Update Run Away, and Update Weeping Angel.

Now your NPC Army is Ready To Go!



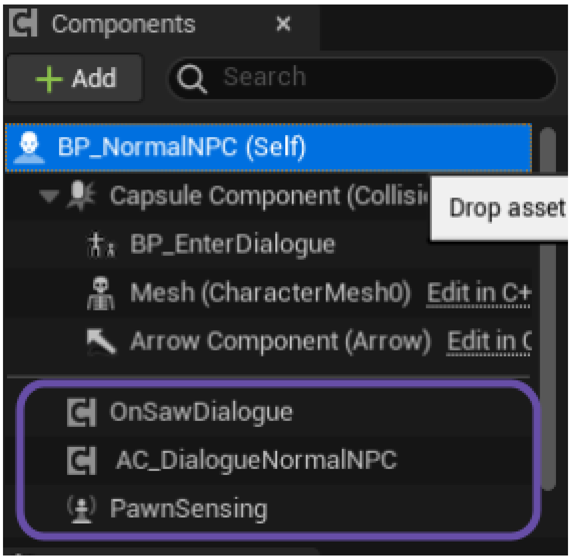
AI_Normal_NPC:

Is in charge of controlling the Pawn and running the behavior tree.



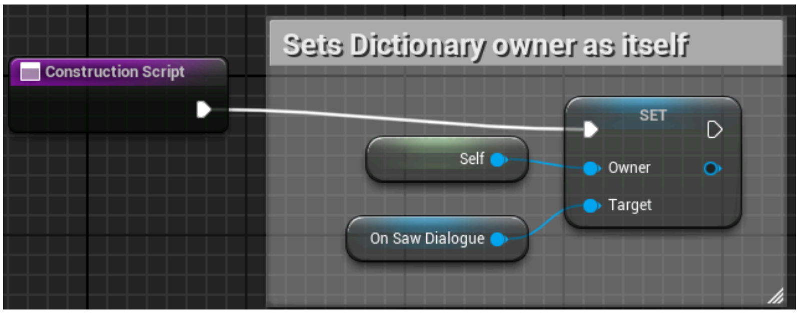
BP_NormalNPC:

Is a Pawn Blueprint that handles most of the logic and stuff for the NPC itself. It contains components to sense the player (PawnSensing), to use dialogue (AC_DialogueNormalNPC), and to save important data (OnSawDialogue) — stills needs to be implemented. It contains update functions for each behavior, and logic in the event graph.



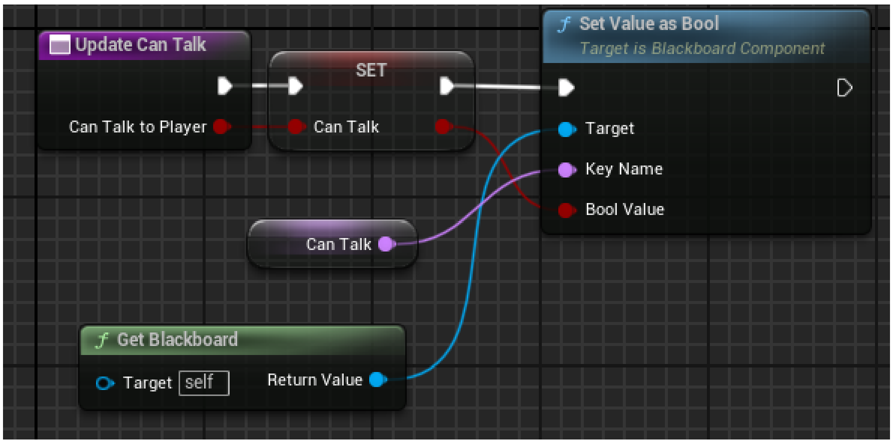
Constructor:

Sets necessary data, in the future loading data will be implemented.



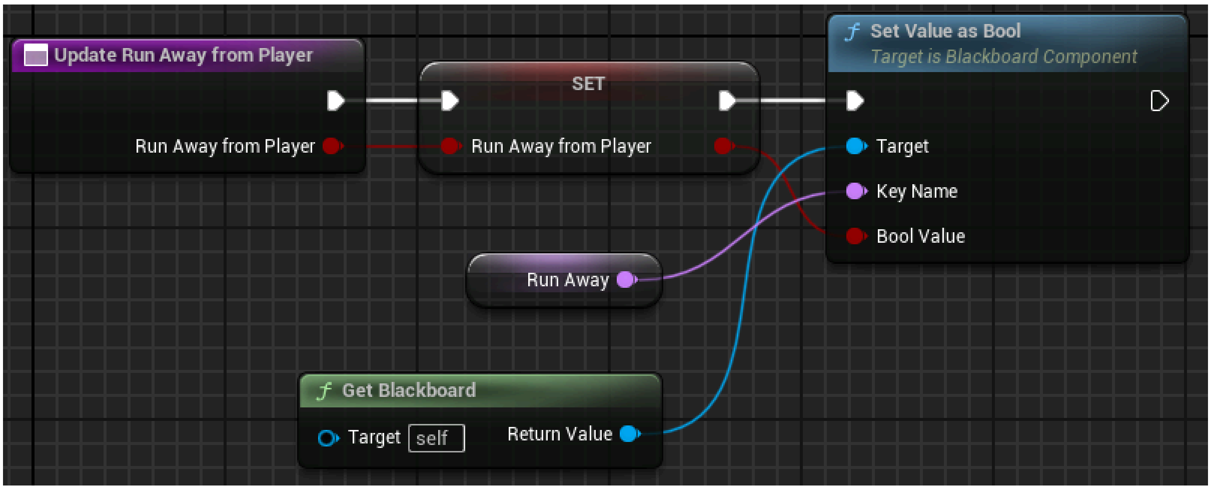
Update Can Talk:

Updates variables in the blueprint and the Blackboard for the tree behavior. It should be used to turn on or off the Dialogue feature.



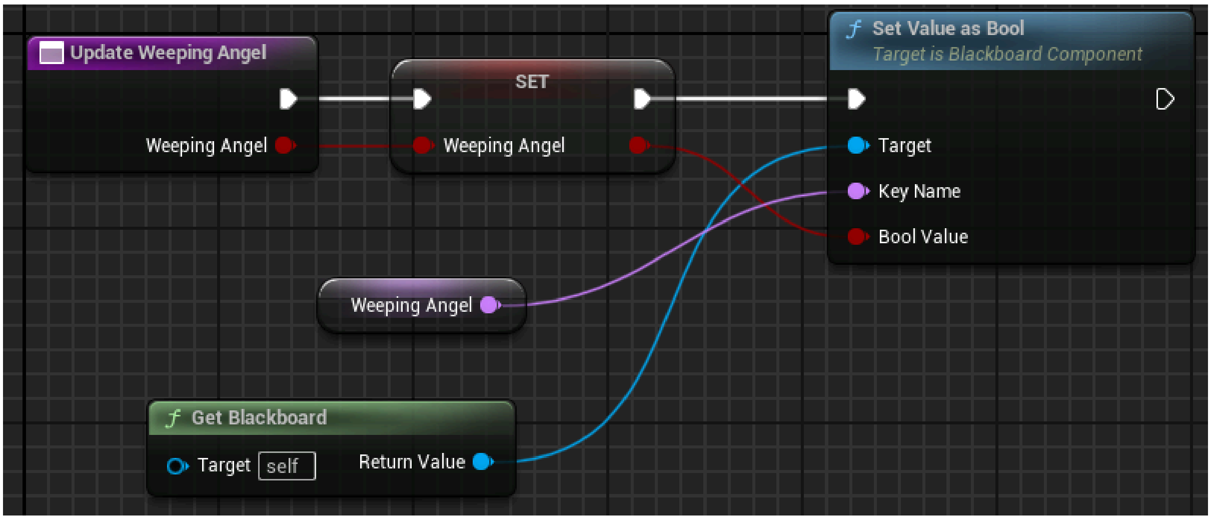
Update Run Away:

Updates variables in the blueprint and the Blackboard for the tree behavior. It should be used to turn on or off the Dialogue feature.

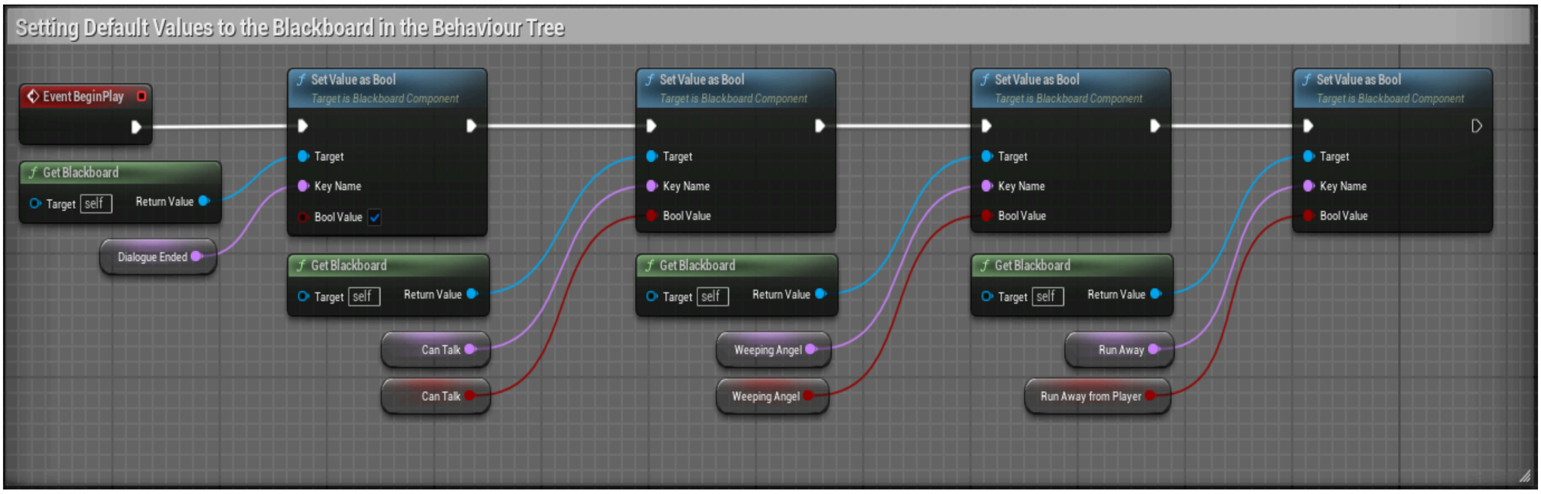


Update Weeping Angel:

Updates variables in the blueprint and the Blackboard for the tree behavior. It should be used to turn on or off the Dialogue feature.



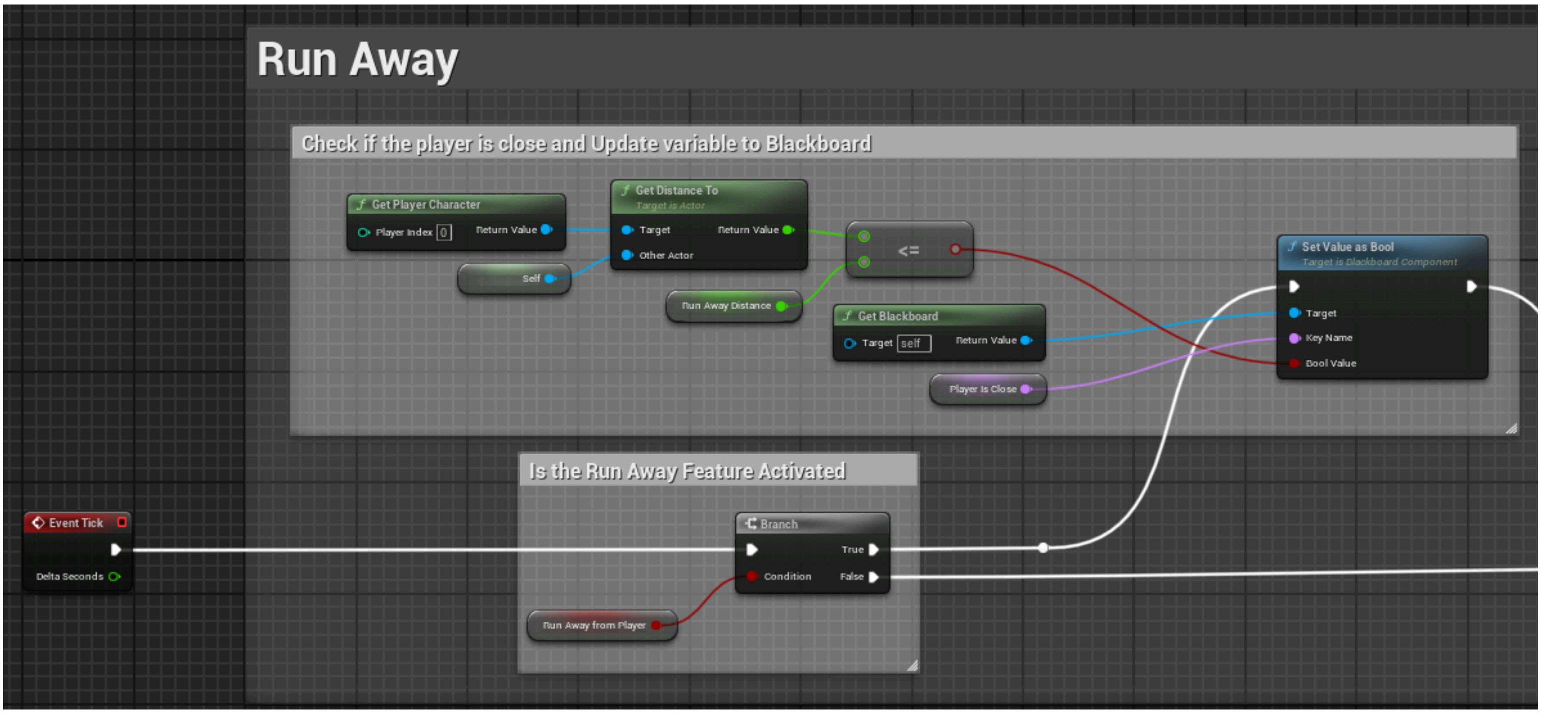
Event Graph: Contains the most important calculations for certain features. If you do not care how this works
SKIP EVERYTHING BELOW HERE:



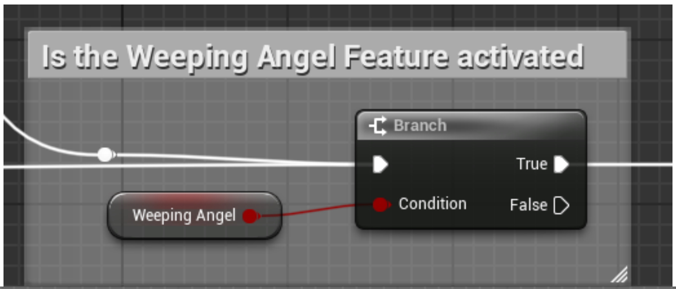
Event Begin Play: Sets the default values to the variables in the Blackboard of the Behavior Tree. This data is used to activate behaviors.

Event Tick: Contains the logic for Run Away and Weeping Angel.

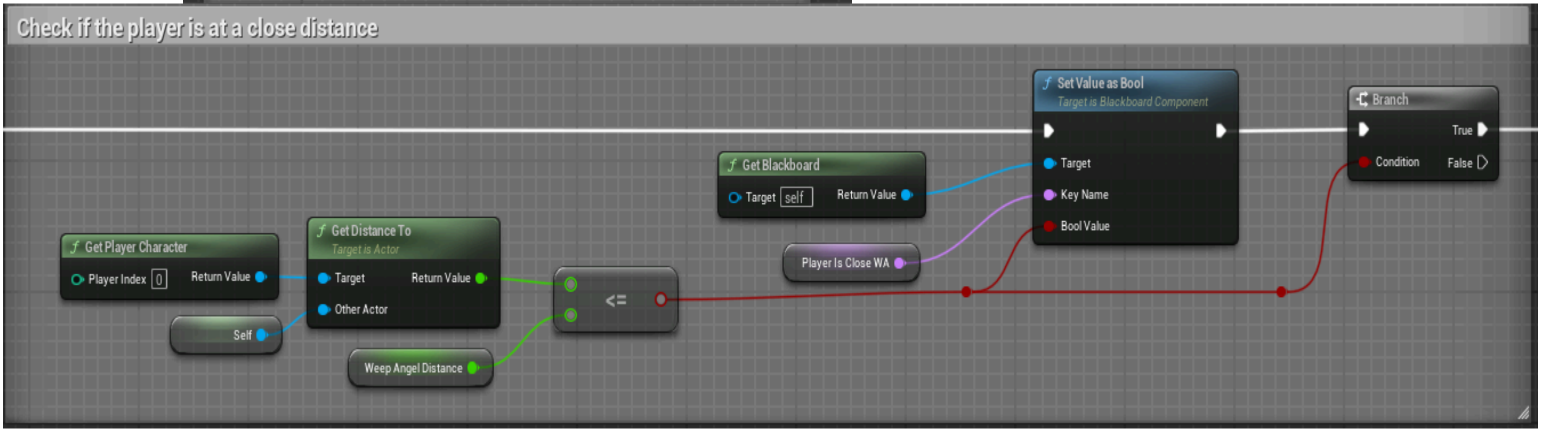
Run Away: Double checks if the feature has been activated to avoid unnecessary calculations, then checks if the distance of the NPC to the player is less than the value set, and updates the blackboard in the behavior tree.



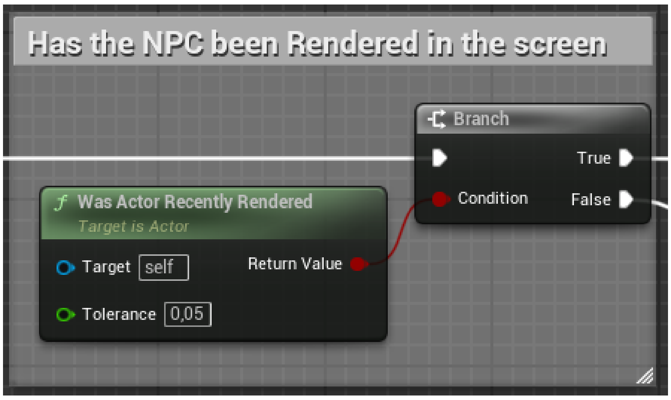
Weeping Angel: First it checks if the feature has been activated to avoid unnecessary calculations.



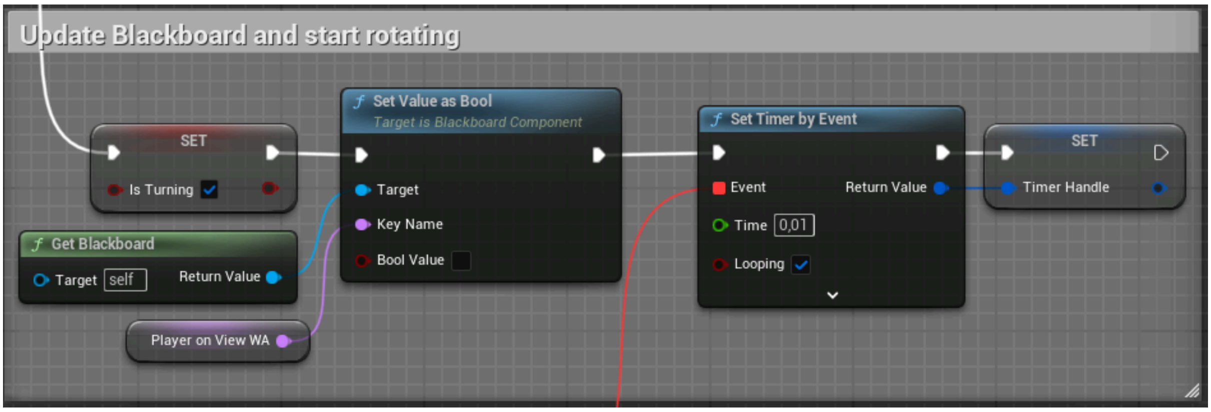
Next, it checks if the player and the NPC are at a lower distance and the one set, if the player is closer then it moves on, if not, calculations stop there.

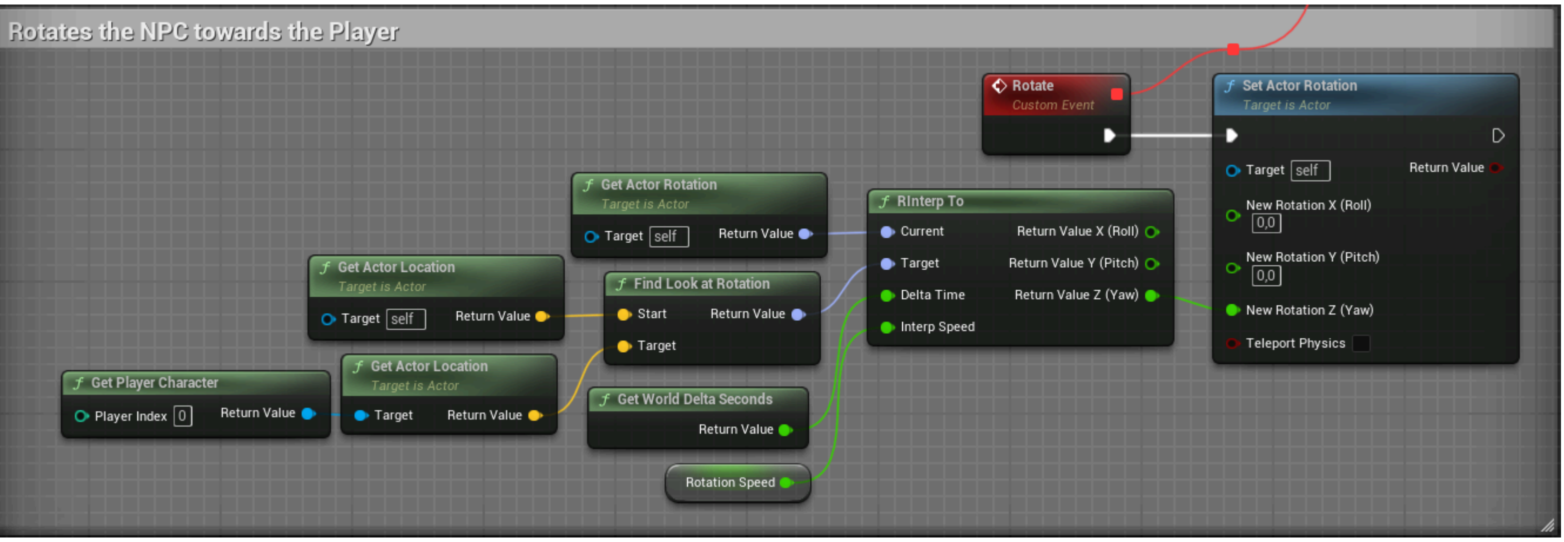


Then, it checks if the NPC has been rendered on the screen, this is to check if the player is seeing the NPC anywhere on the screen!

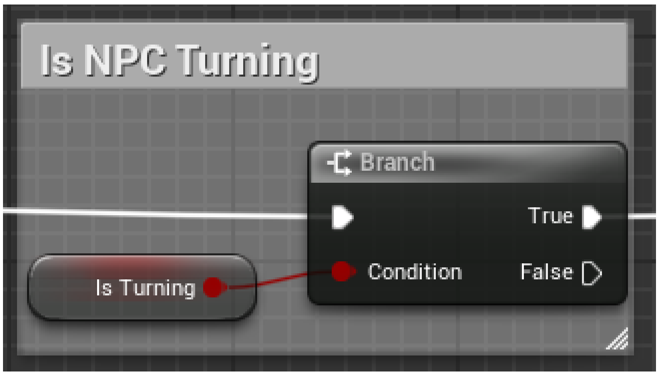


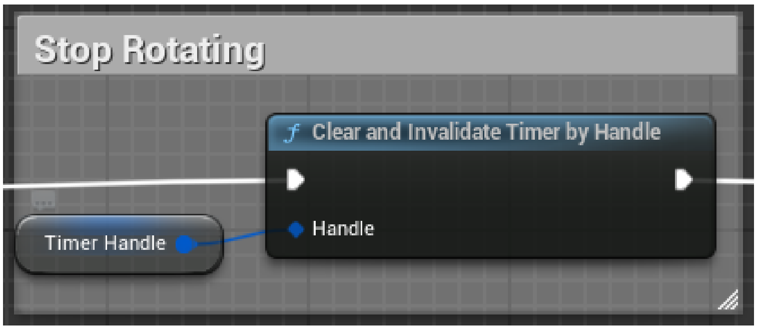
If the NPC has not been rendered then, it updates the blackboard in the behavior tree and starts an event by timer that handles all the rotating logic. The event that we just started is Rotate. This event gets the location of the player and NPC to determine the rotation the NPC should have to face the player, then it gradually changes the rotation.



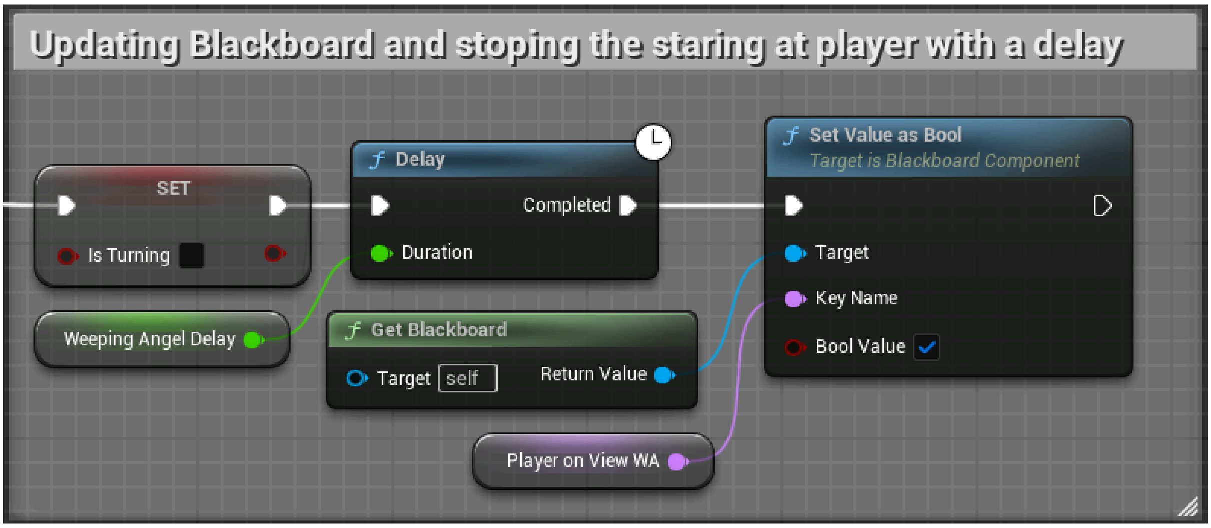


If the NPC has not been rendered (is not in the view of the player) then it checks if the NPC is currently turning. If it is, it stops the NPC by clearing the event by timer.



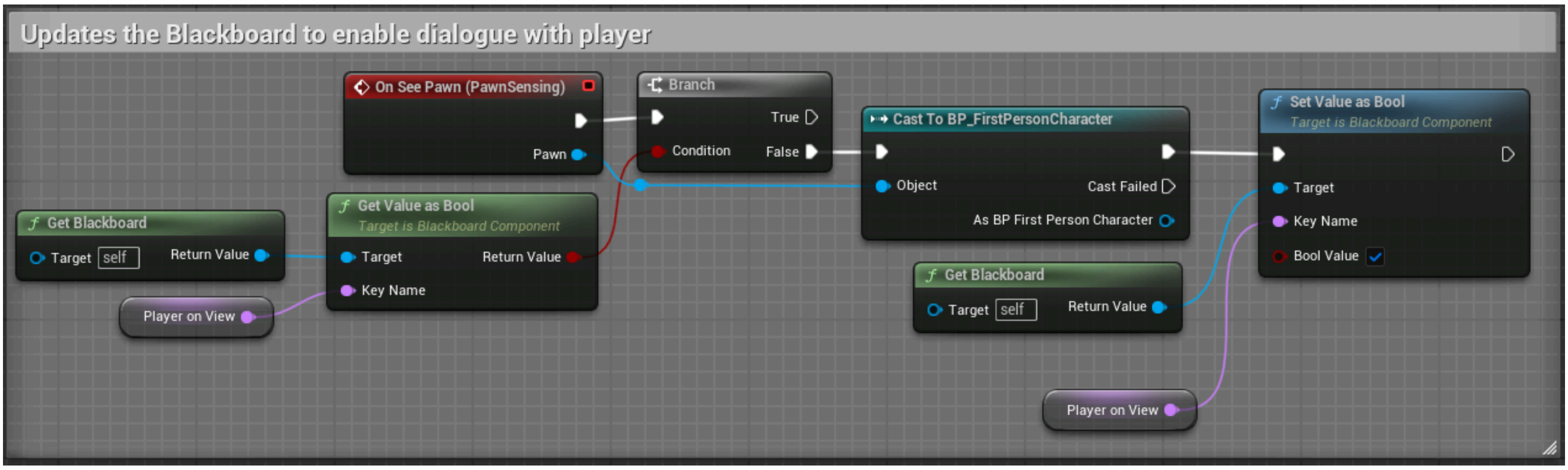


Once all these checks are completed, variables in the blueprint are updated and set to the blackboard in the behavior tree with a small delay. The delay allows for the player to have some time to find the NPC staring at it.

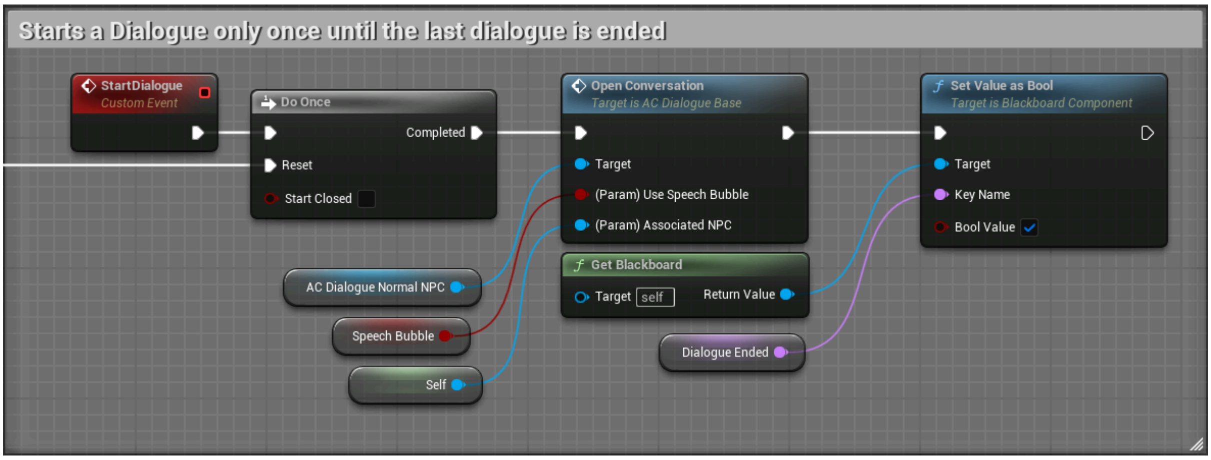


Dialogue: Handles all the logic to start and end a dialogue with the player.

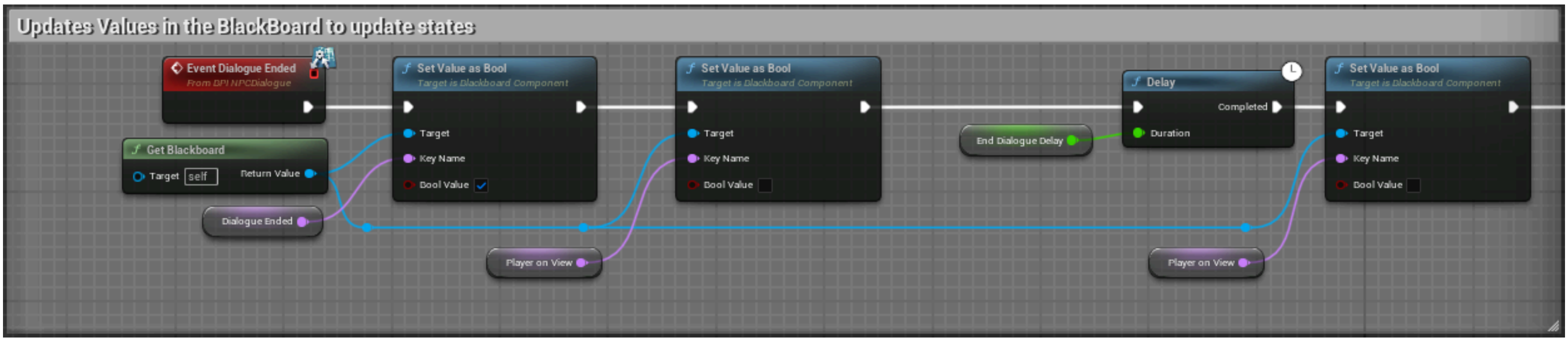
On See Pawn (Pawn Sensing): Gets activated when the player enters the Sight sense area of the NPC. This event checks if the pawn in front of the NPC is a player and if it is it updates variables in the blackboard in the behavior tree. This happens on every frame the player is in the area which means that the variable PlayerOnView is updated the first time and the following times it serves as a check to avoid unnecessary calculations. Also we only want to update from player in not on sight to player is on sight and no other times.



Start Dialogue: Only gets done once, in case the frame by frame updates fail for some reason, it would stop the NPC from having many dialogues at once. Then a conversation is opened through the AC_DialogueNormalNPC, then it updates the blackboard in the behavior tree.

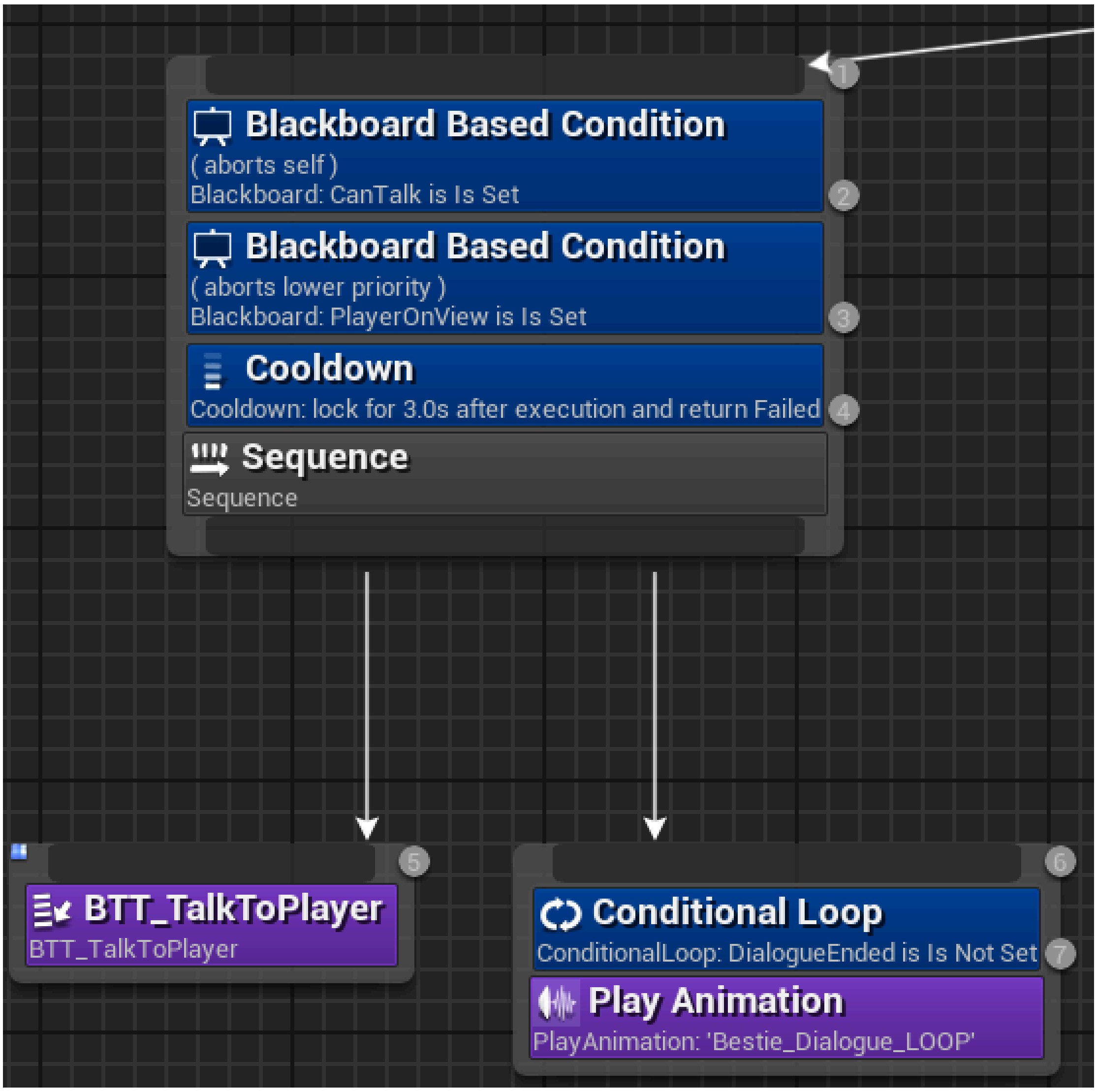


Event Ended Dialogue: It just updates the blackboard in the behavior tree to tell it that the conversation has ended and the player is not on view with a small delay to avoid starting a new conversation right after closing the actual one. After the delay it updates the PlayOnView variable again in case the OnSeePawn event was triggered and it accidentally updates the variable to true. Then it resets the Do Once on the StartDialogue event.

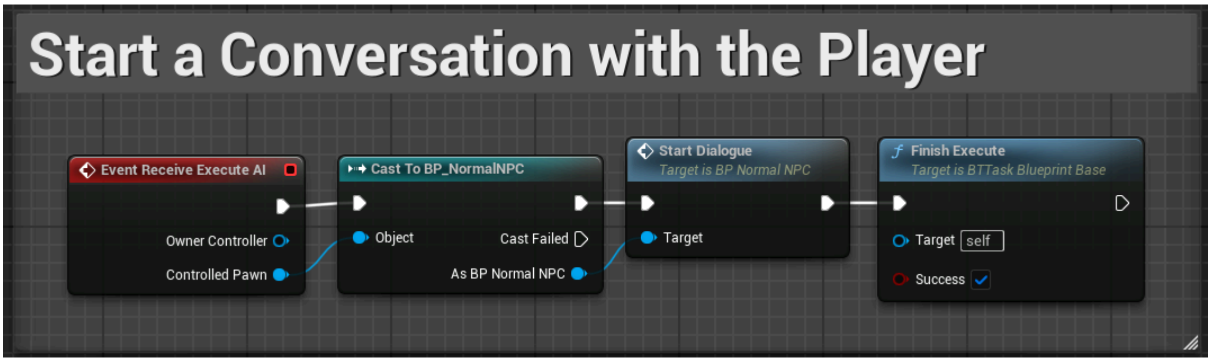


BT_NormalNPC:
Behavior Tree: This handles the activation logic of the behaviors. The activation starts from left to right. The first node chooses one sequence to go off starting at the leftmost branch.

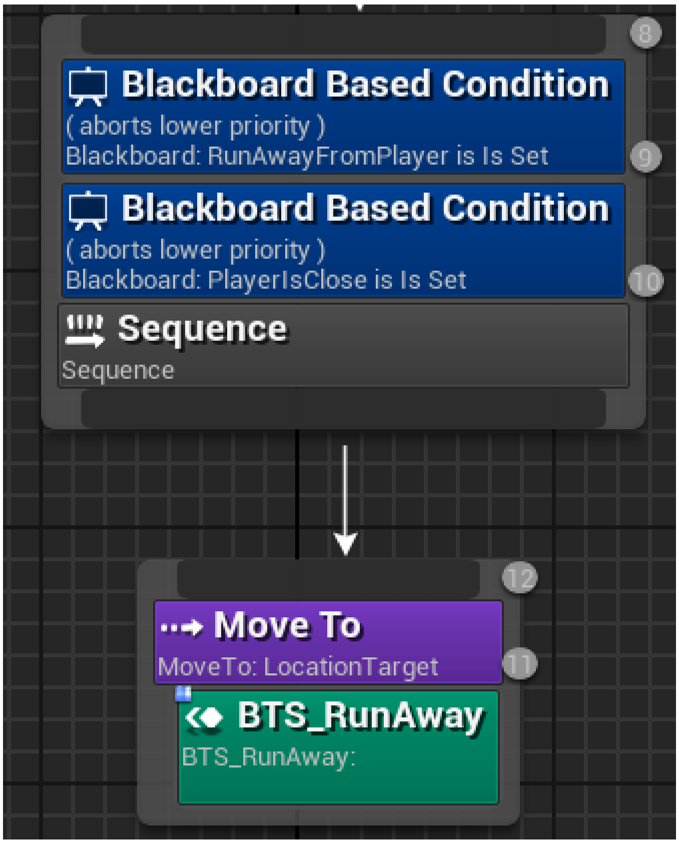
The leftmost branch has three checks. The first check is if the variable CanTalk is true, meaning can the NPC have a dialogue with the player. This check is to see if the user wants to use the dialogue feature at all. The second check is if the variable PlayerOnView is true, meaning, is the player in front of the NPC and within the sight area. At last, there is a cooldown that makes sure there is some time in between each dialogue. There are checks in the BP_NormalNPC events graph to avoid multiple calls of start dialogue while the feature is ongoing, this cooldown is to make sure it doesn't get called right after the task is done. Keep in mind that when a branch has been successfully completed the Selector will restart from the leftmost branch again, meaning that it would get the NPC in an infinite loop.


1
Blackboard Based Condition
(aborts self)
Blackboard: CanTalk is Is Set
2
Blackboard Based Condition
(aborts lower priority)
Blackboard: PlayerOnView is Is Set
3
Cooldown
Cooldown: lock for 3.0s after execution and return Failed
4
Sequence
Sequence
5
BTT_TalkToPlayer
BTT_TalkToPlayer
6
Conditional Loop
ConditionalLoop: DialogueEnded is Is Not Set
7
Play Animation
PlayAnimation: 'Bestie_Dialogue_LOOP'

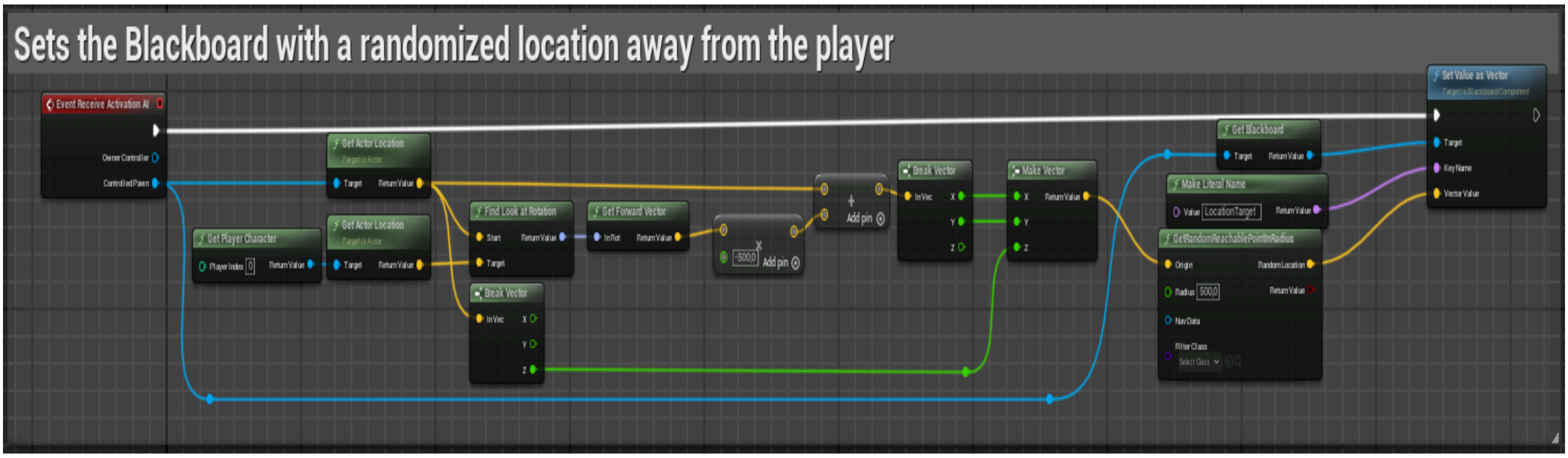
Once checks have been completed the task BTT_TalkToPlayer starts a dialogue with the player and then an animation gets played in a loop while the dialogue has not ended.



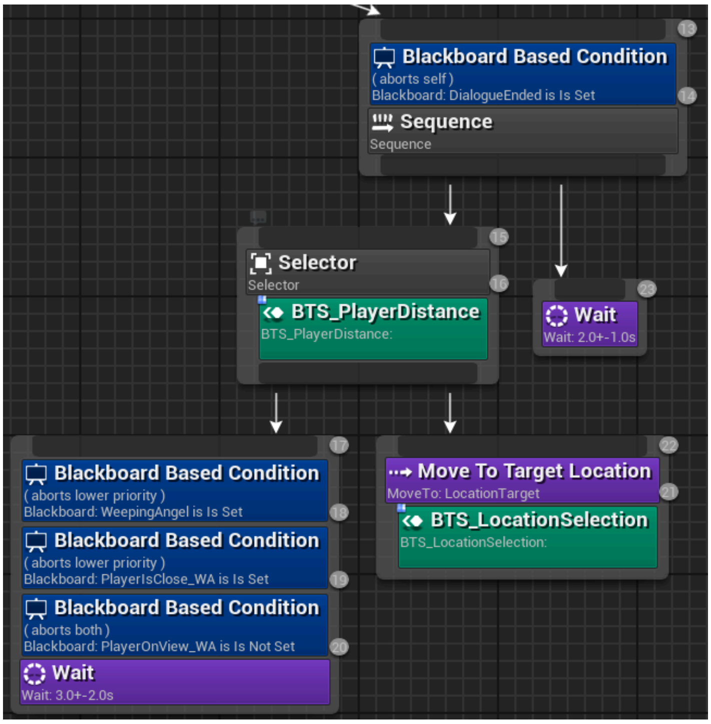
The middle branch handles the RunAway behavior with two checks. The first one is to check if the variable RunAwayFromPlayer is true, meaning the user wants to use this feature. The second check if PlayerIsClose is true, meaning the player is close to the NPC. Then the sequence leads to a Move To task that moves the NPC to a location that gets determined by the BTS_RunAway service.


8
Blackboard Based Condition
(aborts lower priority)
Blackboard: RunAwayFromPlayer is Is Set
9
Blackboard Based Condition
(aborts lower priority)
Blackboard: PlayerIsClose is Is Set
10
Sequence
Sequence
12
Move To
MoveTo: LocationTarget
11
BTS_RunAway
BTS_RunAway:

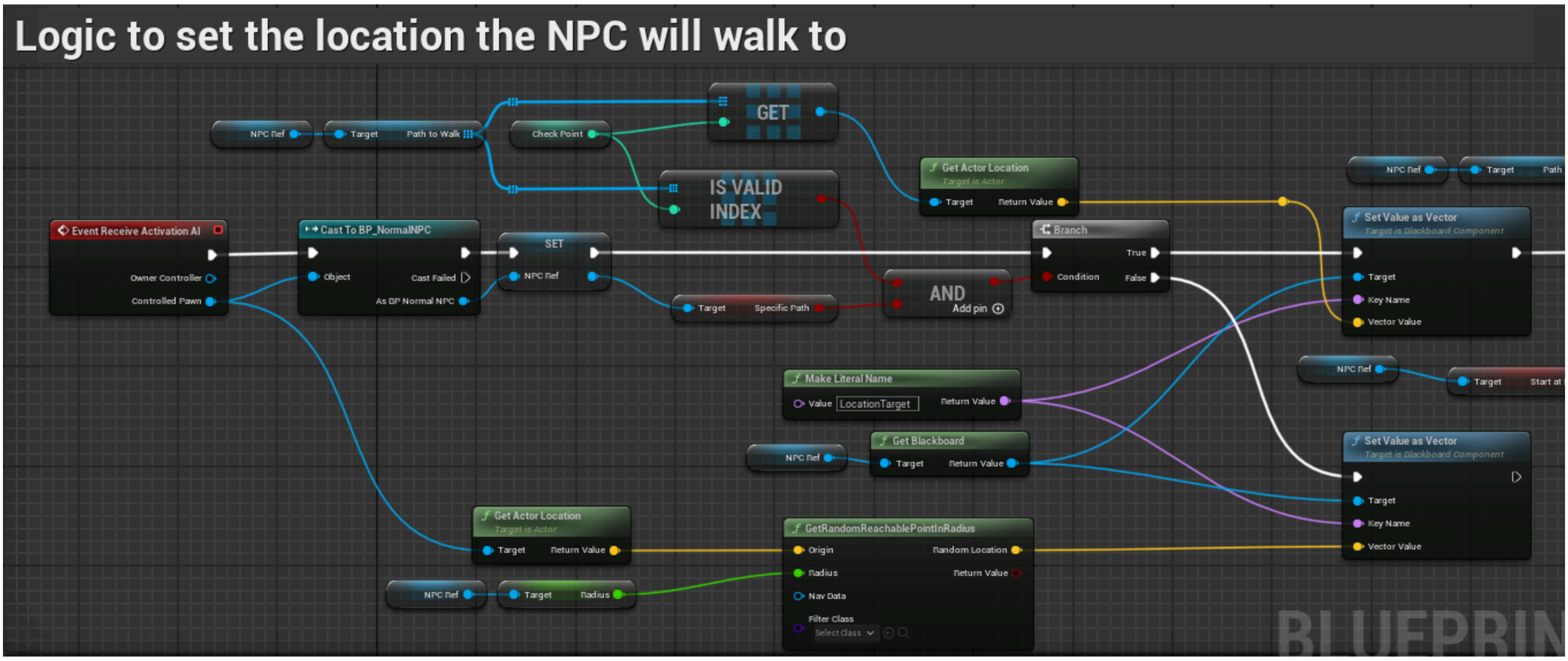
This service sets a location by first getting the direction of where the Player is at and extending it by -500 units to go the opposite direction then breaks that vector to make sure the NPC doesn't go in the Y axis (up). Then, a random point in a 500 units radius is generated and that gets set as the target location for the NPC to go to. Keep in mind the center of the radius is determined by the direction the player * -500 units. Once it gets this location the service updates the Location target in the Blackboard of the behavior tree.



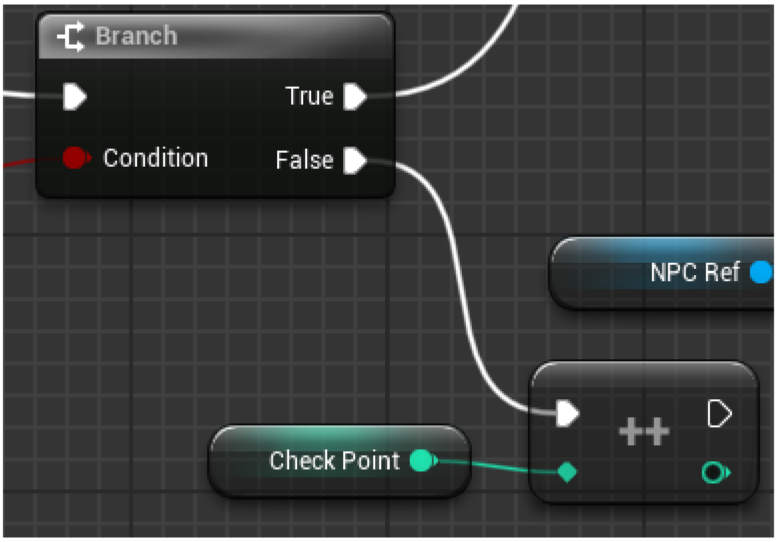
The Rightmost branch is in charge of two things, the Weeping Angels and the Move To behaviors. The main check to this branch is if there is a dialogue going on, meaning the variable DialogueEnded is true. Then the sequence goes down to the selector and updates the variable for the player distance, a backup in case scripted nodes fail to update. On this selector the Wait right most task has three checks. The first one is if the Weeping Angel feature has been activated, the WeepingAngel bool should be true. The second one checks if the player is close to the NPC, determined by PlayerIsClose_WA on true. The last check is if PlayerOnView_WA is not true. Then the NPC "waits" while the rotation happens in the Events graph.


Blackboard Based Condition
(aborts self)
Blackboard: DialogueEnded is Is Set
Sequence
Sequence
Selector
Selector
BTS_PlayerDistance
BTS_PlayerDistance:
Wait
Wait: 2.0+-1.0s
Blackboard Based Condition
(aborts lower priority)
Blackboard: WeepingAngel is Is Set
Blackboard Based Condition
(aborts lower priority)
Blackboard: PlayerIsClose_WA is Is Set
Blackboard Based Condition
(aborts both)
Blackboard: PlayerOnView_WA is Is Not Set
Wait
Wait: 3.0+-2.0s
Move To Target Location
MoveTo: LocationTarget
BTS_LocationSelection
BTS_LocationSelection:
13
14
15
16
17
18
19
20
21
22
23

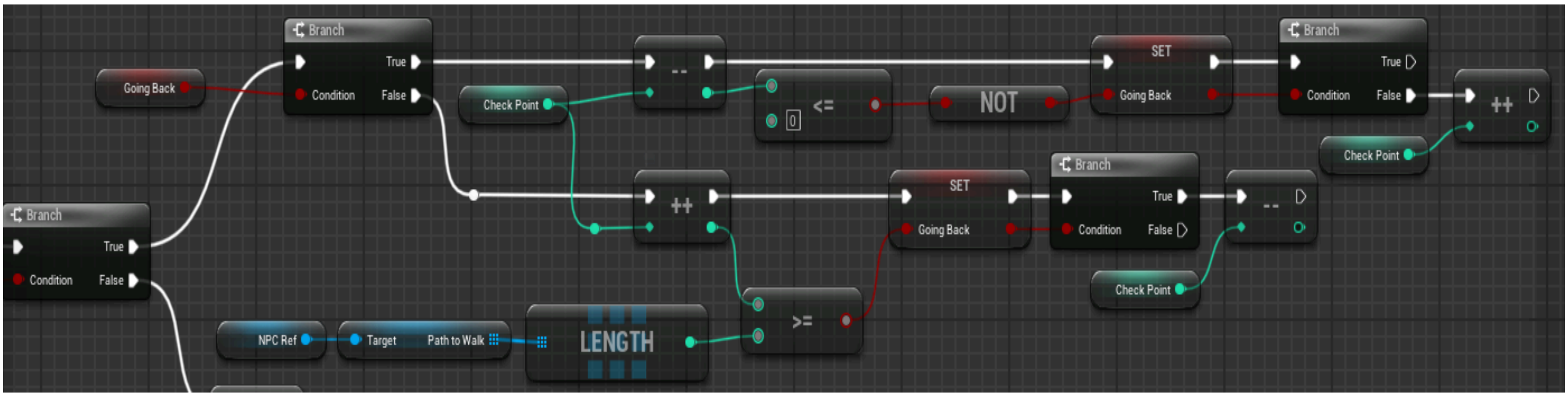
The last feature is Move To Target Location, this one runs when the WeepingAngel is not running. It uses a BTS_LocationSelection service to run the calculations. This service has two main features. The first one is to get a random location within a radius and set that as the target. The second one is to set locations based on an array of Target Points, the NPC will set each target point one by one. Additionally, this feature has three other features.



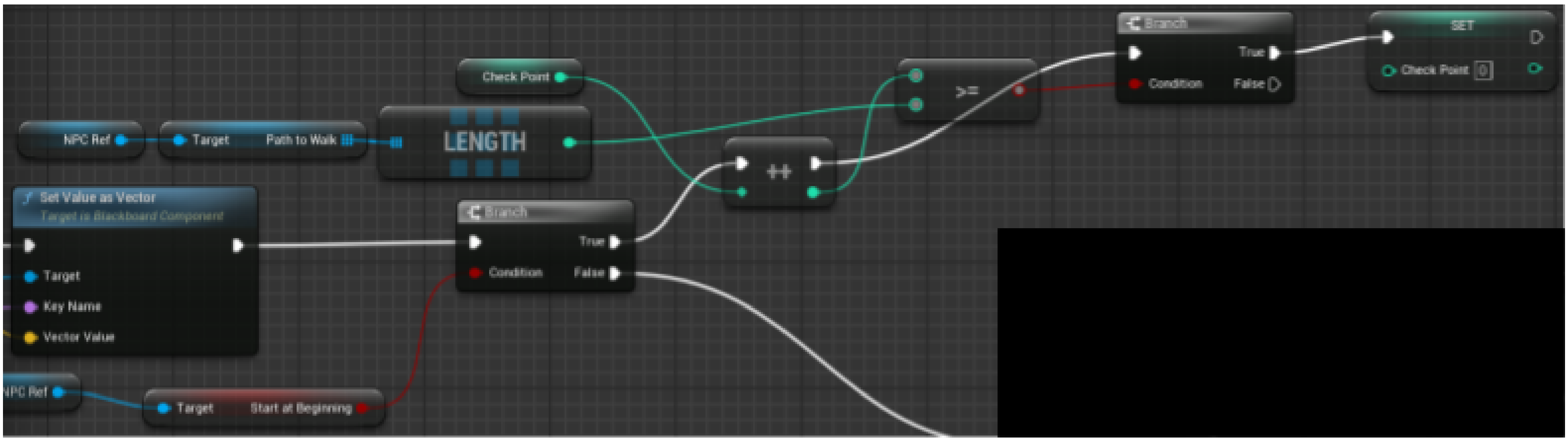
The first additional feature is to set random locations once all the locations in the array have been completed.



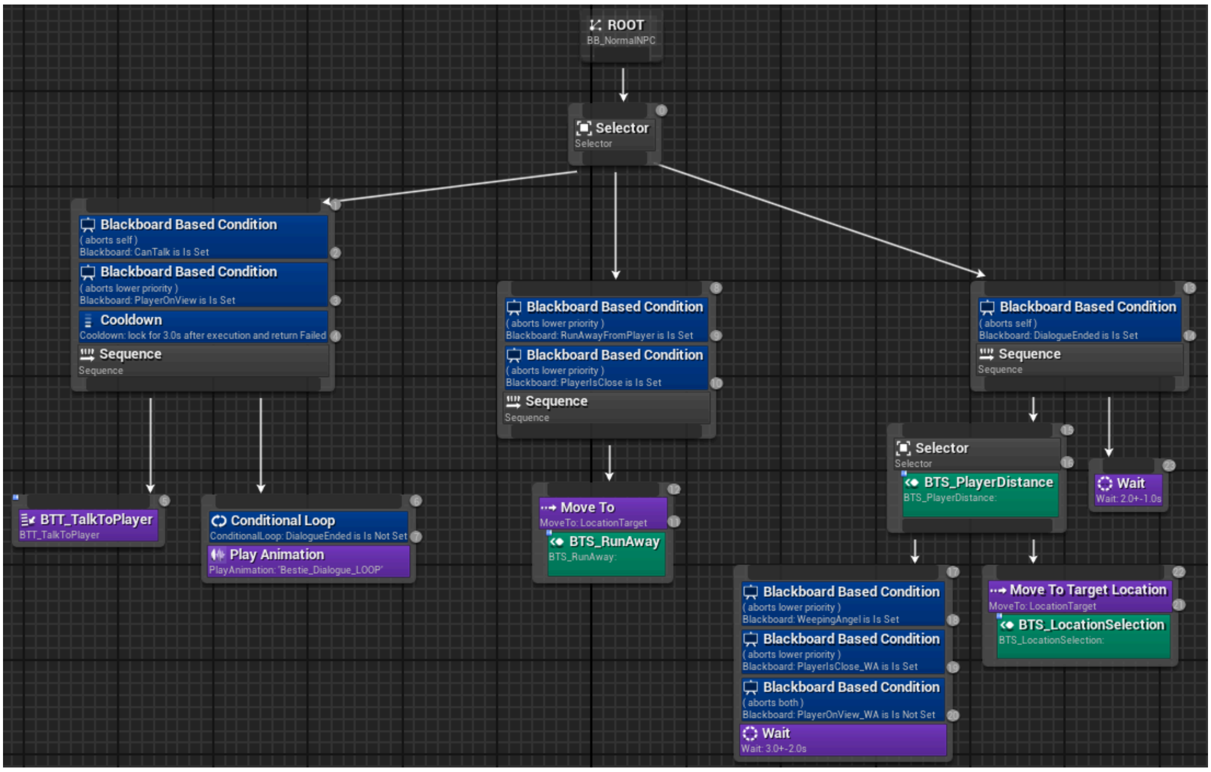
The second feature is to loop backwards through the array of locations, meaning that once the array has been completed the NPC will go through the array in reverse from last to first, once it gets to the first one, it will again go through the array normally.



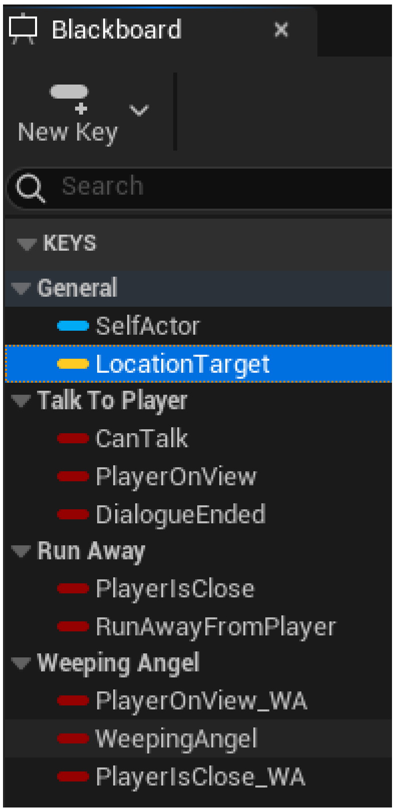
The last feature is to start at the beginning of the array, meaning that once the array has been completed, the NPC will go to the first location stored in the array, this is great for making the NPC walk in loops.



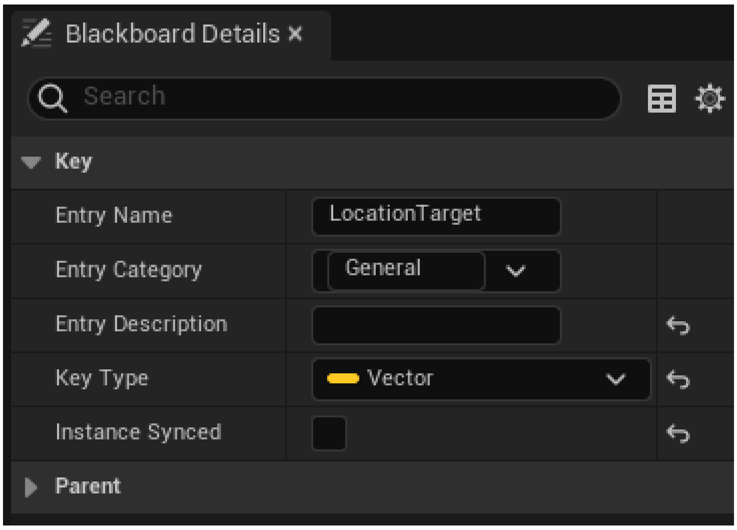
Picture of the full Behavior Tree!


ROOT
BB_NormalNPC
Selector
Selector
Blackboard Based Condition
(aborts self)
Blackboard: CanTalk is Is Set
Blackboard Based Condition
(aborts lower priority)
Blackboard: PlayerOnView is Is Set
Cooldown
Cooldown: lock for 3.0s after execution and return Failed
Sequence
Sequence
BTT_TalkToPlayer
BTT_TalkToPlayer
Conditional Loop
ConditionalLoop: DialogueEnded is Is Not Set
Play Animation
PlayAnimation: 'Bestie_Dialogue_LOOP'
Blackboard Based Condition
(aborts lower priority)
Blackboard: RunAwayFromPlayer is Is Set
Blackboard Based Condition
(aborts lower priority)
Blackboard: PlayerIsClose is Is Set
Sequence
Sequence
Move To
MoveTo: LocationTarget
BTS_RunAway
BTS_RunAway:
Blackboard Based Condition
(aborts self)
Blackboard: DialogueEnded is Is Set
Sequence
Sequence
Selector
Selector
BTS_PlayerDistance
BTS_PlayerDistance:
Wait
Wait: 2.0+-1.0s
Blackboard Based Condition
(aborts lower priority)
Blackboard: WeepingAngel is Is Set
Blackboard Based Condition
(aborts lower priority)
Blackboard: PlayerIsClose_WA is Is Set
Blackboard Based Condition
(aborts both)
Blackboard: PlayerOnView_WA is Is Not Set
Wait
Wait: 3.0+-2.0s
Move To Target Location
MoveTo: LocationTarget
BTS_LocationSelection
BTS_LocationSelection:

BB_NormalNPC:
Behavior Trees come with blackboards to store variables that are very useful when making AI behaviors. Every blackboard is just a bunch of variables that can be separated into categories. Each key/variable has settings, you can place a name, category, description, and type. It doesn't save data structures such as arrays or maps.



KEYS
- General
  - SelfActor
  - LocationTarget
- Talk To Player
  - CanTalk
  - PlayerOnView
  - DialogueEnded
- Run Away
  - PlayerIsClose
  - RunAwayFromPlayer
- Weeping Angel
  - PlayerOnView_WA
  - WeepingAngel
  - PlayerIsClose_WA



Key

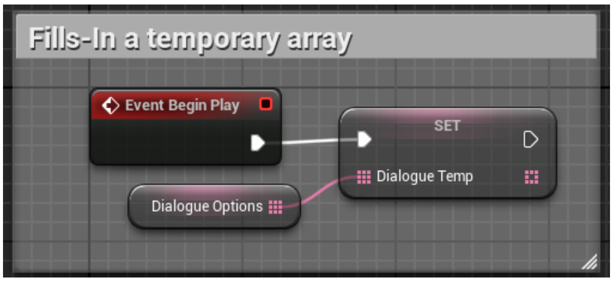
Entry Name	LocationTarget
Entry Category	General
Entry Description	
Key Type	Vector
Instance Synced	□

Parent

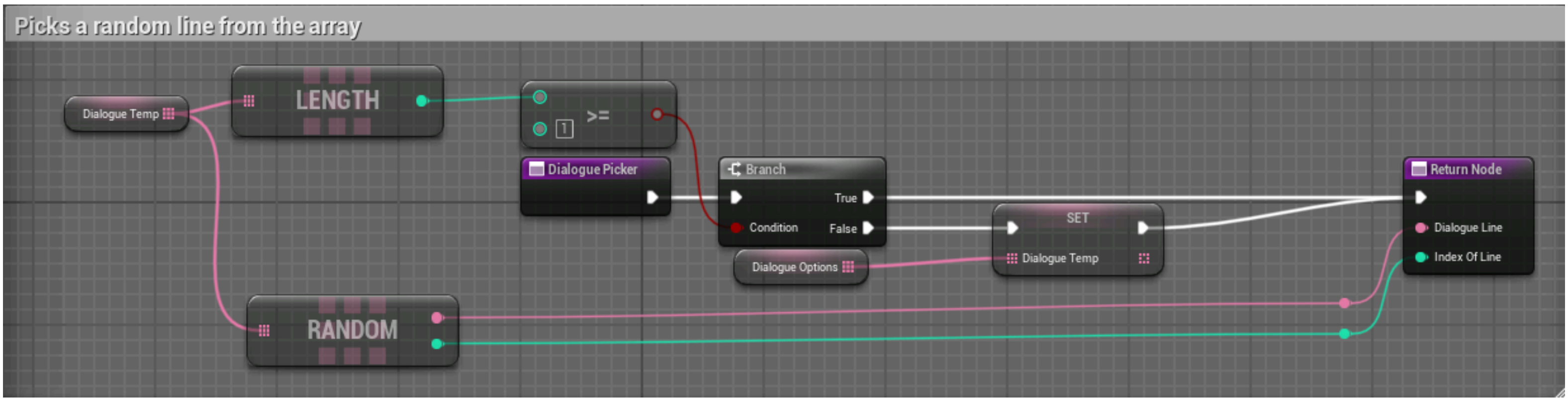
BPI_NormalNPC: It is a very simple interface to call the end dialogue event from the Dialogue component, I made it because I don't like casting…

AC_DialogueNormalNPC:

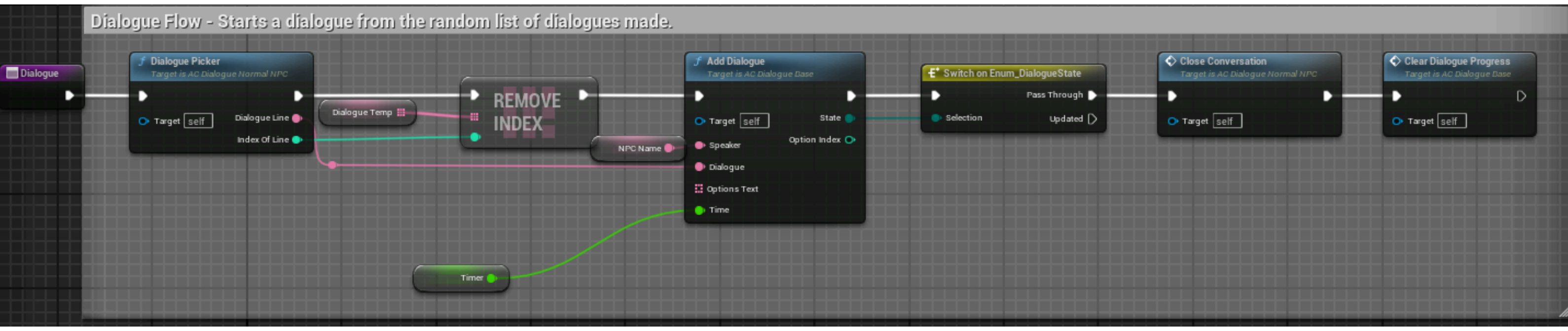
It is a child of the AC_DialogueBase blueprint that handles the dialogue flow. All that is done in this class is to create a temporary array with all the dialogue lines the NPC might say.



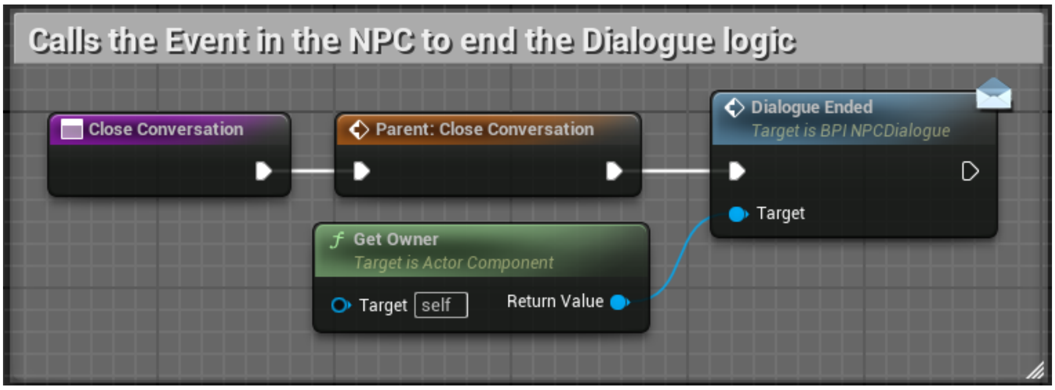
There are three other functions used. The first one is Dialogue Picker, which selects a random line from the temporary array and returns that.



The second function is Dialogue, which is overridden from the base class, all that is done is getting a line from Dialogue Picker, if the temporary array is empty, fill it up again. This is to avoid repeated lines of dialogue often. Then dialogue is added and the conversation is closed, at last we clear progress. Questions, refer to the dialogue Documentation.



The last function is Close Conversation which is an overridden function from the base class that calls the Dialogue Ended event in the BP_NormalNPC.


Calls the Event in the NPC to end the Dialogue logic
Close Conversation
Parent: Close Conversation
Dialogue Ended
Target is BPI NPCDialogue
Target
Get Owner
Target is Actor Component
Target self
Return Value

Resources Used To Create This System:

This system and documentation can be used for future/outside projects with proper accreditation and after KI is done.

- How To Make The AI Run Away From The Player - Unreal Engine Tutorial
- How To Create Weeping Angels in Unreal Engine 5
- How To Make Something Always Face The Player In Unreal Engine 5 (Tutorial)
- How to Make A Behavior Tree in Unreal Engine 5 Tutorial (Complete Guide)
- How to Make a Simple Behavior Tree in Unreal Engine 5 - Advanced AI
- https://dev.epicgames.com