

Algorithms and screenshots

1 **Cache**:

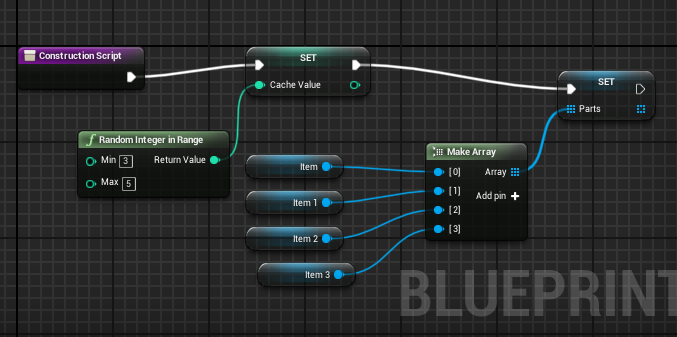
Name and type: Cache item – Actor blueprint

Variables:

Miner – Ai Agent 1 “This is a saved copy of the previous ai agent that mined this cache. Used for communication with the agent and simplification of code”. Has no default value

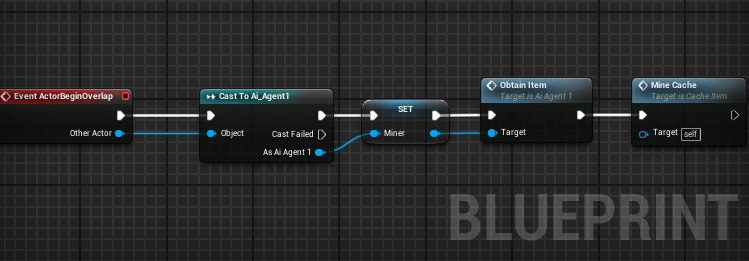
Parts [] – Array of Static mesh components “Used for simplification of code and hold all of the pieces that make up the object visually”. Has no default value

Cache Value – Integer = 0



**Construction script:**

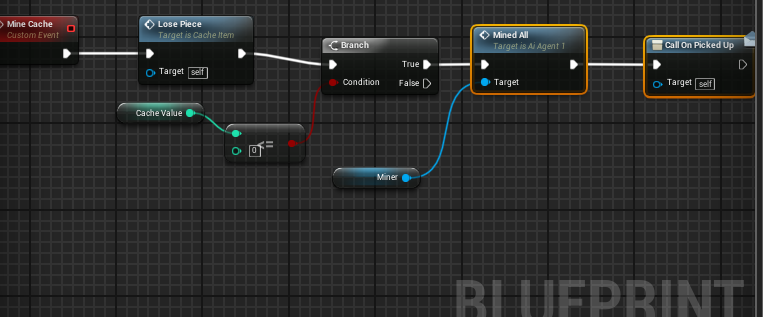
Sets the actual value for the variables Cache value and Parts [].



**Main / On overlap function:**

When an Ai Agent 1 overlaps with the cache then the miner variable is set, and the functions

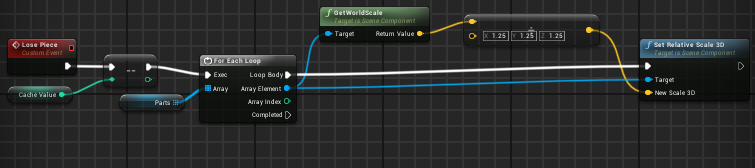
Obtain Item (See Ai Agent 1 :: Obtain Item) and Mine Cache is Called.



**Mine Cache:**

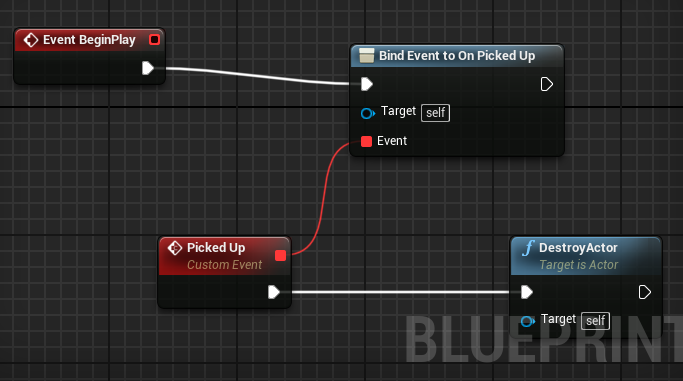
Calls the function Lose piece and if the cache value is less than or equals 0

then call Mined All (See Ai Agent 1 :: Mined All) then Call the event dispatcher on picked up



**Lose Piece:**

Decrement Cache value and for all the parts divide the scale by 1.25 so it will shrink by .25.



**On Picked Up Event Dispatcher / Picked Up Event:**

On Picked up is called only once the entire cache has been mined so the destroy actor event is bound to this event. So, when the cache has been fully mined it will be destroyed.

**Communication Lines:**

The Cache Item uses the saved miner to call functions in **Mine Cache** and

in the **Main / on Overlap Function.** The way it communicates to the Cache Spawner through the event dispatcher **On Picked Up**.

2 **Cache Spawner**:

Name and type: Cache Spawner – Actor blueprint

Variables:

Index – Integer “A temp variable used to find which location in the array should be used” = 0

Probability – Float “A random integer which will decide which probability has been selected”

Wait Time – Float “An amount of time that the spawner waits before spawning new caches when told to spawn caches.” = 5.0

CachesMined – Integer “A running total of the caches mined. Set to -1 by default for simplification in code” = -1

NumCaches – Integer “The total number of caches in the world at one time. Is compared with CachesMined to know when to respawn caches into the world” = 0

Spawn Double – Bool “Lets the spawner know if it is spawning 1 or 2 caches” = False

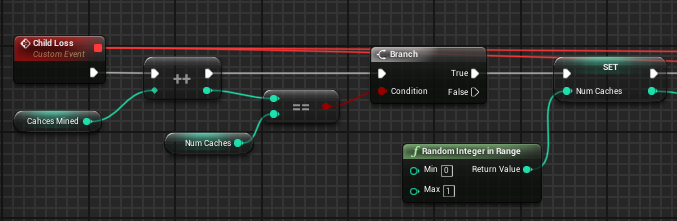
Location Prob – Struct {Vector Location, Float Probability} “A data type that holds a location and the probability of using the location”

Locations 1 and Locations 2 – Location Prob “Two tables with locations that caches can spawn. Locations 1 is used for the first cache to be spawned. Locations 2 is used for the second cache” = All the predefined locations the caches can spawn

Table – Location Prob “A temp array that holds which location table is being used in the function **Spawn Cache**”

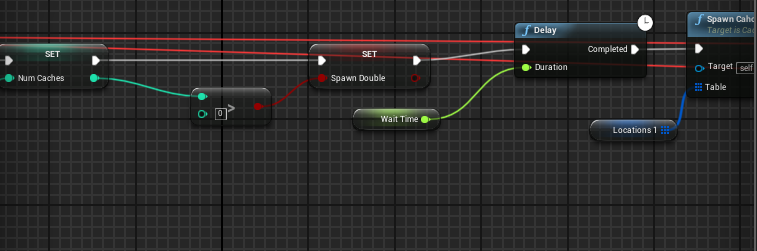


**Begin Play:**  This calls the main function that spawns the children that the spawner keep track of.



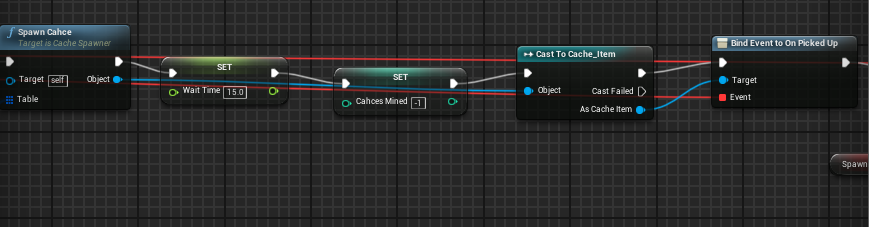
**Main / Child Loss Pt 1:**

Caches Mined is incremented then compared to Num Caches. The reason Caches Mined is set to -1 is so when this is first executed it will equal Num Caches which by default is equal to 0. If they are equal then a new total of caches to be spawned is generated and stored in Num Caches.



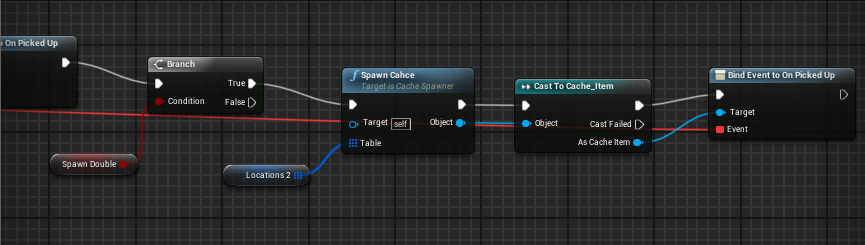
**Main / Child Loss Pt 2:**

If Num Caches is greater than 0 than we store the fact that the spawner needs to spawn 2 caches instead of 1. A delay is put in place to give the Ai time to go back to their home and wander before a new cache is spawned. Then **Spawn Cache** is called and given the first location table.



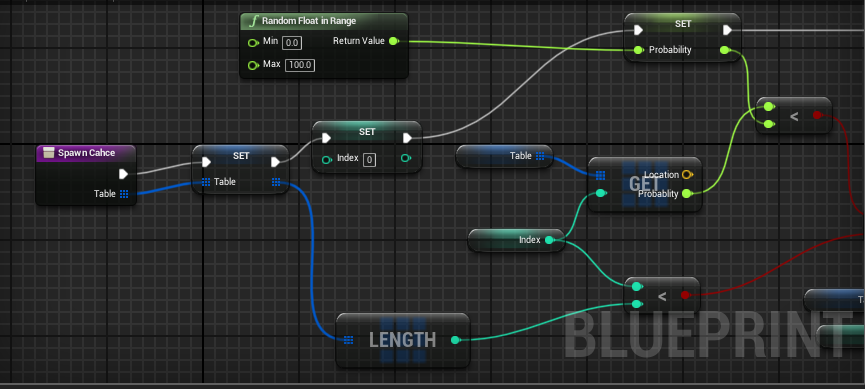
**Main / Child Loss Pt 3:**

After **Spawn Cache** is called it returns the reference to the Cache\_Item that is spawned and binds **Child Loss** to **On Picked Up** Event dispatcher. Wait Time and Caches Mined are set to their new default values.



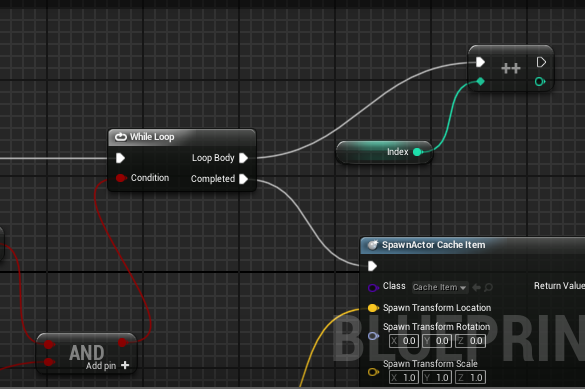
**Main / Child Loss Final:**

Lastly if Spawn Double is true do the same thing as **Pt 3** , but use Location Table 2.

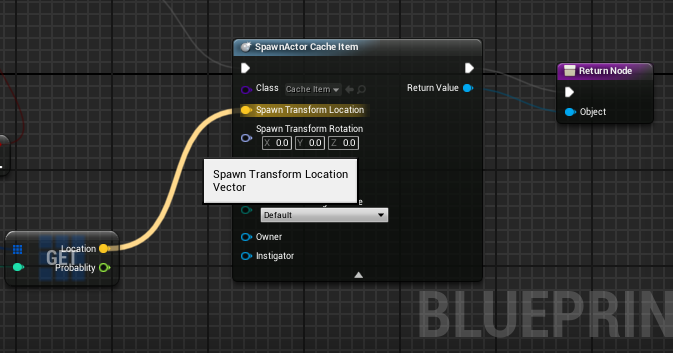
****

**Spawn Cache Pt 1:**

First Set Table = to the in parameter and reset the index to 0. Generate a random number between 0 and 100 and that is the probability value. As long as Index < Table.Length() and the locations probability is less than the generated probability Increment Index (See Pt 2).



**Spawn Cache Pt 2:**

****

**Spawn Cache Final:**

Once the generated probability is less than the locations probability then spawn the **Cache\_Item** at the location generated and return the reference to the actor spawned.

**Communication Lines:**

The only thing the Cache Spawner needs to communicate with is the **Cache\_Item** and it does this by using the Event Dispatcher **On Picked Up** found in **Cache\_Item.**

3 **AI Agent**:

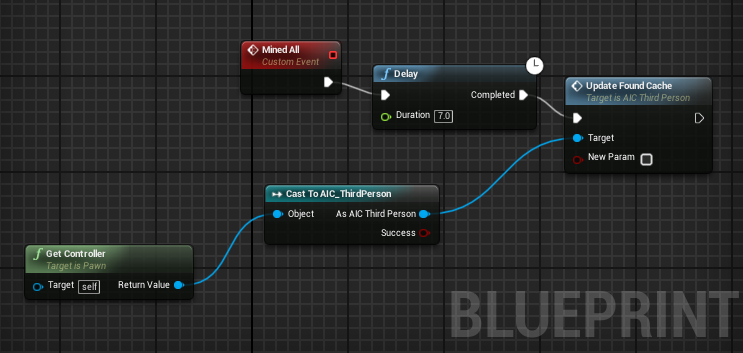
3a **Ai Agent 1**:

Name and type: Ai Agent 1 – Actor blueprint

Variables:

Target – Actor Reference “A reference to the target perceived by the agent’s sense”

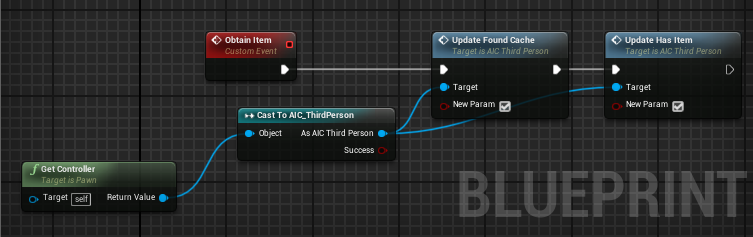
Target Lost Timer – Timer Handle “Used for timer-based functions predefined by unreal”



**Mined All:**

Tells the agent that it has mined all of a cache is updates Found Cache to false

(See AI controller:: Update Found Cache for implementation). The 7 second delay is put in place to give the ai time to return home before the variable Found Cache is updated.

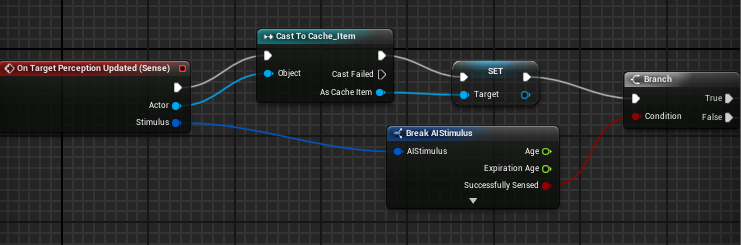


**Obtain Item:**

Tells the agent it has obtained an item from a cache.

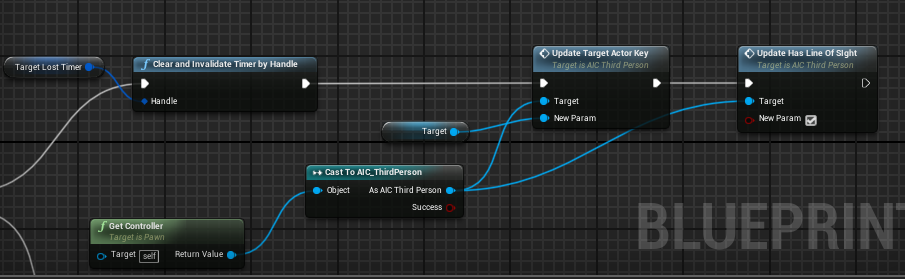
(See AI controller:: Update Found Cache for implementation).

(See AI controller:: Update Has Item for implementation).



**Main / Perception Updated Pt 1:**

If the agent sees a **Cache\_Item** then it sets the target reference to the perceived cache.

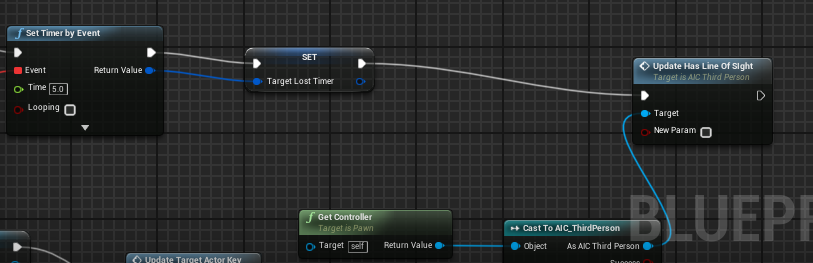


**Main / Perception Updated Pt 2 (True Path):**

If the Agent has perceived a **Cache\_Item** then clear the Target Lost timer then Update Target actor with the Target actor reference and update has line of sight to true.

(See AI controller:: Update Target Actor Key for implementation).

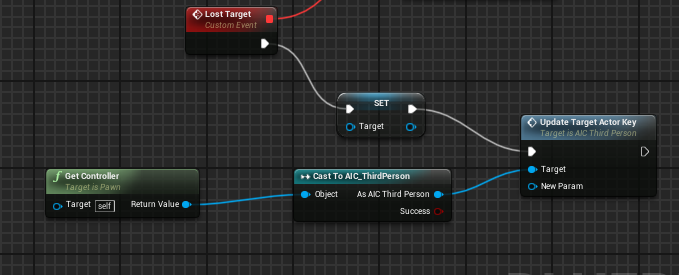
(See AI controller:: Update Has Line Of Sight implementation).



**Main / Perception Updated Pt 2 (False Path):**

Sets The target Lost Timer and updates Has line of sight to false.

(See AI controller:: Update Has Line Of Sight implementation).



**Main / Perception Updated Pt 2 (False Path) Lost Target:**

Lost target is bound to the set timer event above. Set the target actor reference to null and update the target actor to null as well to simulate losing the target.

(See AI controller:: Update Target Actor Key for implementation).

**Communication Lines:**

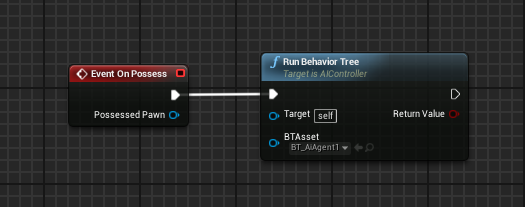
The agent uses the **AIC\_ThirdPerson Ai controller** to communicate to the behavior tree and blackboard through Update Events like Update Target Actor.

3b **Ai controller**:

Name and type: AIC\_ThirdPerson– Ai Controller

Variables:

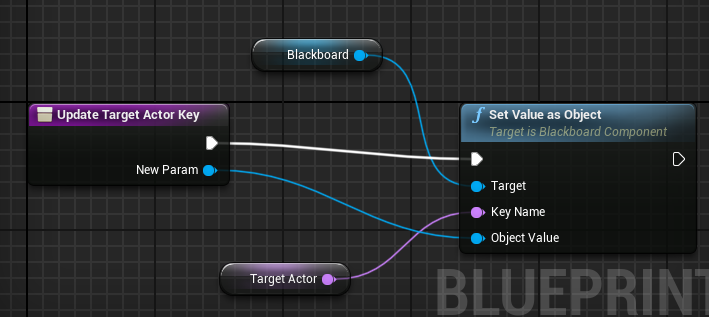
Target Actor, Line of sight, Found Cache, Has Item – Name “All Are Names with blackboard keys as default values” (See Blackboard keys)



**On Posses:**

When the controller possesses the previously discussed Ai Agent it then runs the Behavior tree

(See Behavior tree for implementation)



**Update \_\_\_\_\_\_\_ :**

For Update Has line of sight, Update Target Actor Key, Update Has Item, and Update Found Cache all have basically the same implementation just change The key and the type of value being set.

**Communication Lines:**

The sole purpose of an Ai controller is communication between the actor / agent and the behavior tree. The agent calls the Update functions that the controller handles and sends to the behavior tree to update values in the blackboard.

4 **Behavior tree and Blackboard**:

Blackboard Keys:

Target\_Location – Vector “The location the agent has chosen to wander to”

Home – Vector “The location where the agent is spawned”

Most\_Recent, Second\_Recent, and Almost\_Forgotten - “Keys used to simulate memory

of past cache locations”

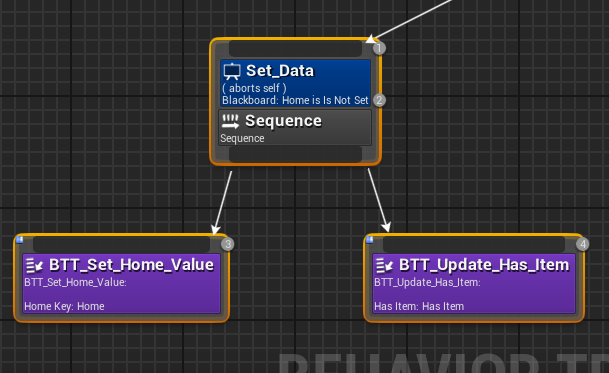
Target\_Actor – Vector “The location of a cache that the agent has found”

Has\_Line\_of\_sight – bool “Is true if the agent sees a Cache”

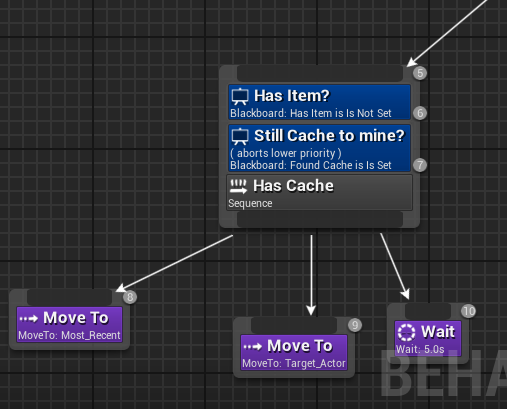
Has Item - bool “Is true if the ai has overlapped with a cache and has not returned home”

Found Cache – bool “Is true if the ai has found a cache and has not fully mined it”

**Behaviors and Tasks:**

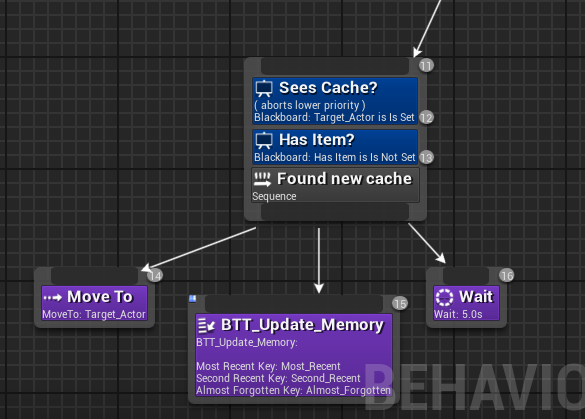
****

**Set Data:** The First behavior called. Checks if Home has been set and if it hasn’t then set the home value to the current location of the Agent. Update Has item is also called to set Has Item to false, because Obtain Item (See Ai Agent 1 :: Obtain Item) Sets Has item to true.

****

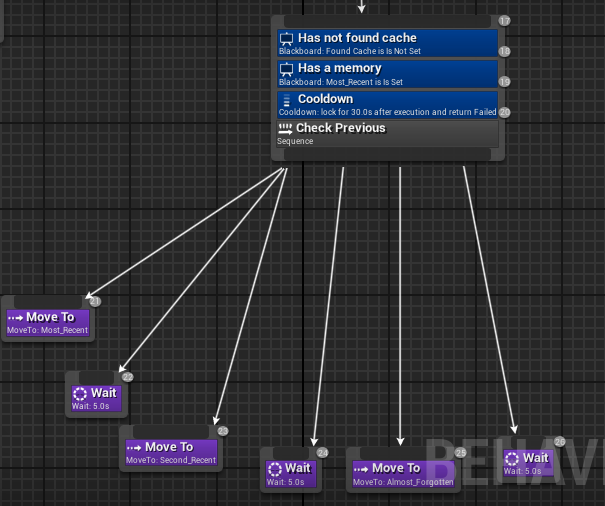
**Has Cache:**

Can only run if the Agent has found a Cache and there is still more cache to mine. “Has Item?” Checks to see if the agent doesn’t have an item, because if it has an item it cant mine more of the same cache. The Agent moves to the most recent memory of a cache then moves to the target actor to actually mine the cache.

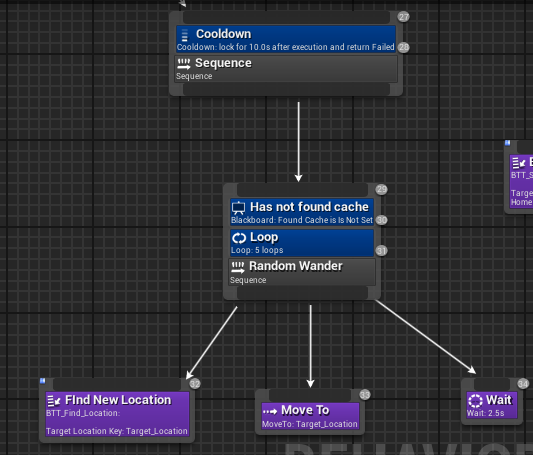


**Found New Cache:**

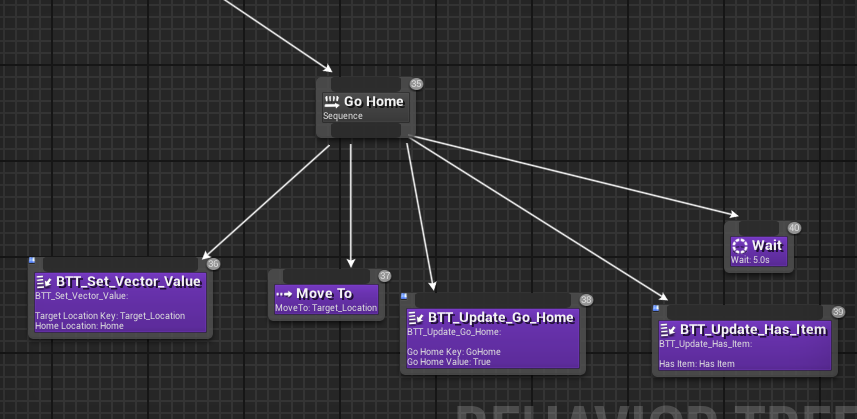
If Target actor is set than all other events lower in priority are abandoned to go to that cache. As long as the agent does not have an item then it can mine the newly found cache. The agent moves toward the cache then updates its memory (See BTT\_Tasks) and then waits.

**  
Check Previous:**

Only executes if a cache has not been found and has a memory of at least one cache. Then checks all previously visited cache sites. After executing there is a 30 second cool down timer so the ai Agent has time to wander.

  
**Random Wander:**

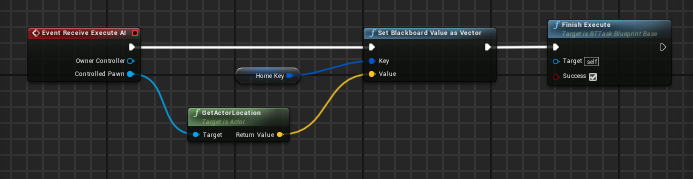
As long as the agent has not found a cache have it find a location it can move to, move to it and wait repeat 5 time. After completed there is a small 10 second cool down for the agent to spend time at home.



**Go Home:**

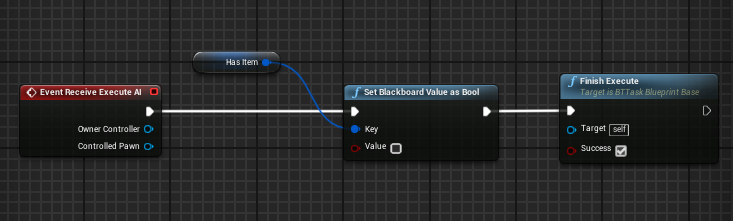
Set the new target location to home. Moves to home. Updates that the agent is at home and that the agent has no item then waits.

**BTT Tasks:**

****

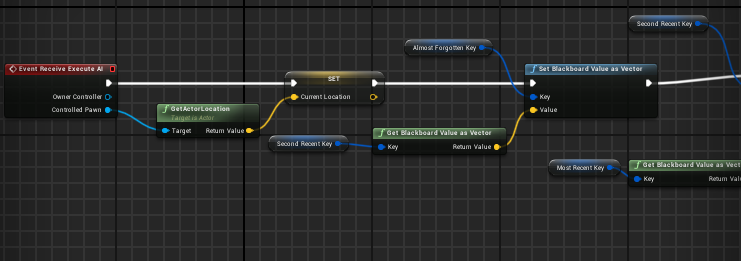
**BTT\_Set\_Home\_Value:**

Get the agents current location and sets the Home key to that value.



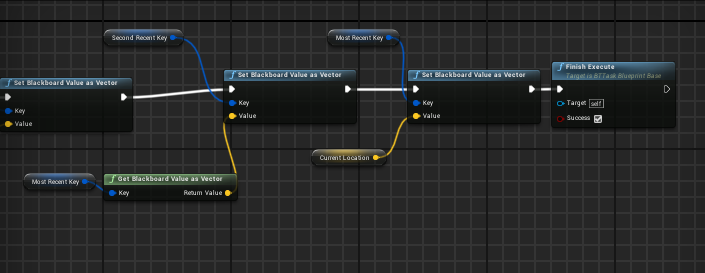
**BTT\_Update\_Has\_Item:**

Updates Has item to false. This represents the agent dropping off the item at home.

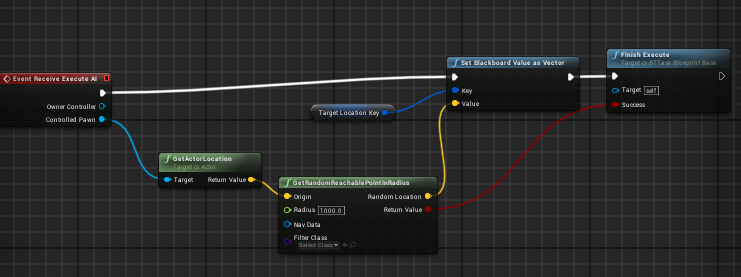


**BTT\_Update\_Memory Pt1:**

Sets current location to a temp variable for later. Then take Second Recent and move it to almost forgotten. Then take Most recent and move it Second recent. Finally take the temp variable for current location and move it to Most recent.

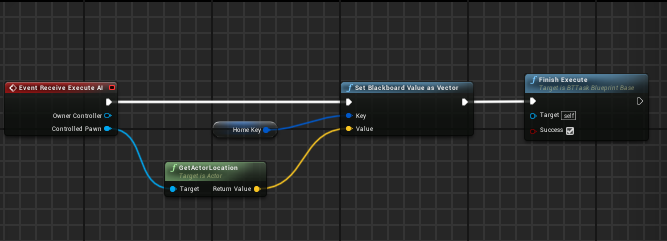
****

**BTT\_Update\_Memory Pt2:**

****

**BTT\_Find\_Location:**

Looks for a navigable location and sets target location to the new location found.



**BTT\_Set\_Vector\_Value:**

Sets target location to home.