视角

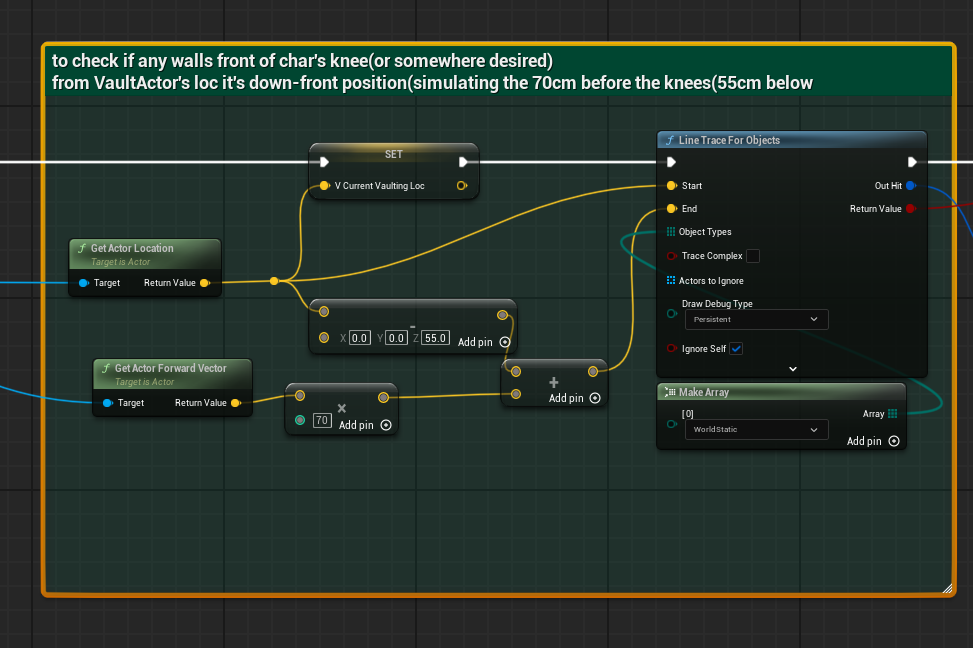
首先是尼玛fpp的话，引擎自带的rotation只开了yaw, 但是如果在摄像机那里开了rotation,一切就顺畅了

通信

motions

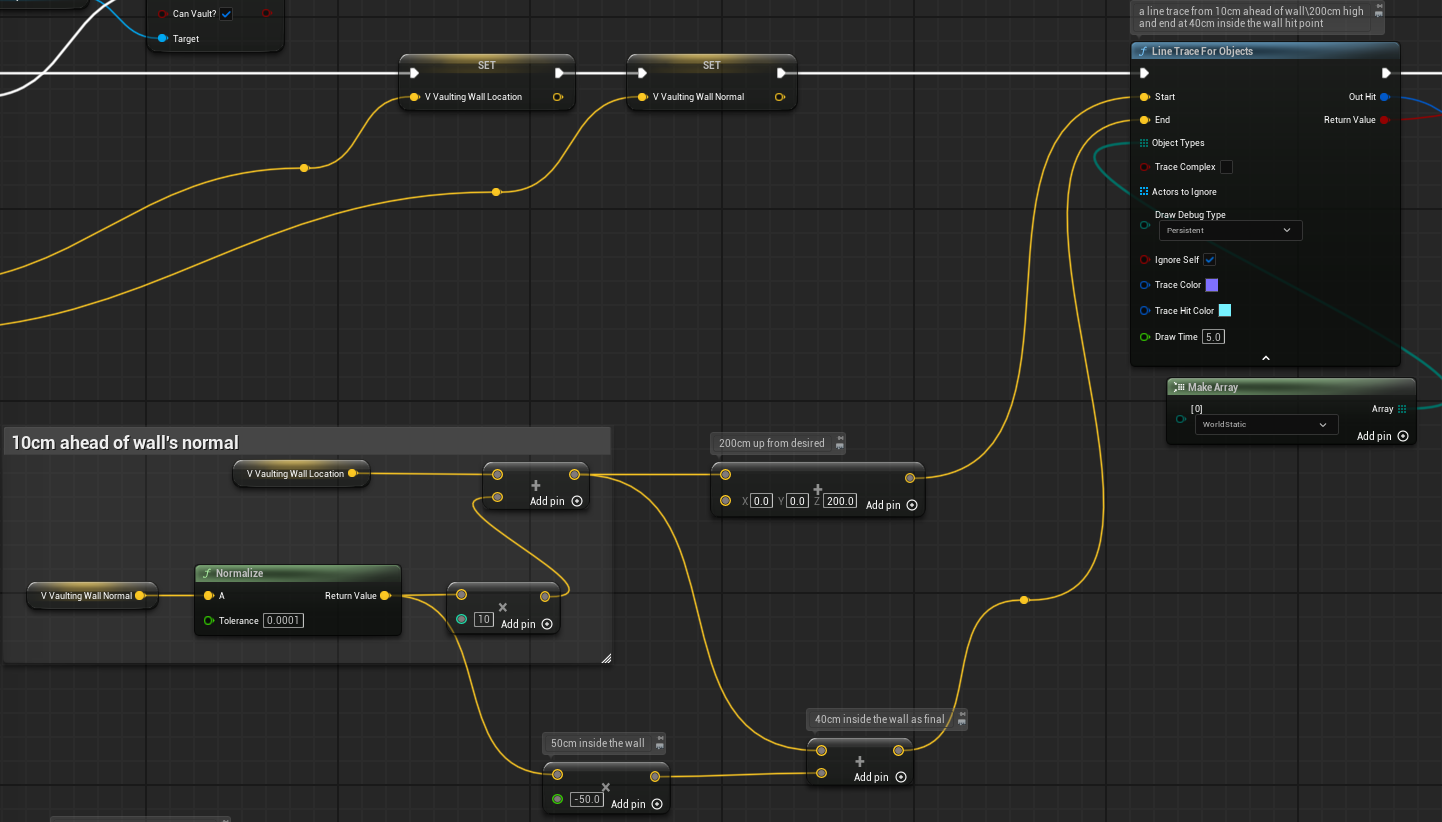
# # vaulting system

vaultSys



at first, line trace by objects to detect if any walls in front

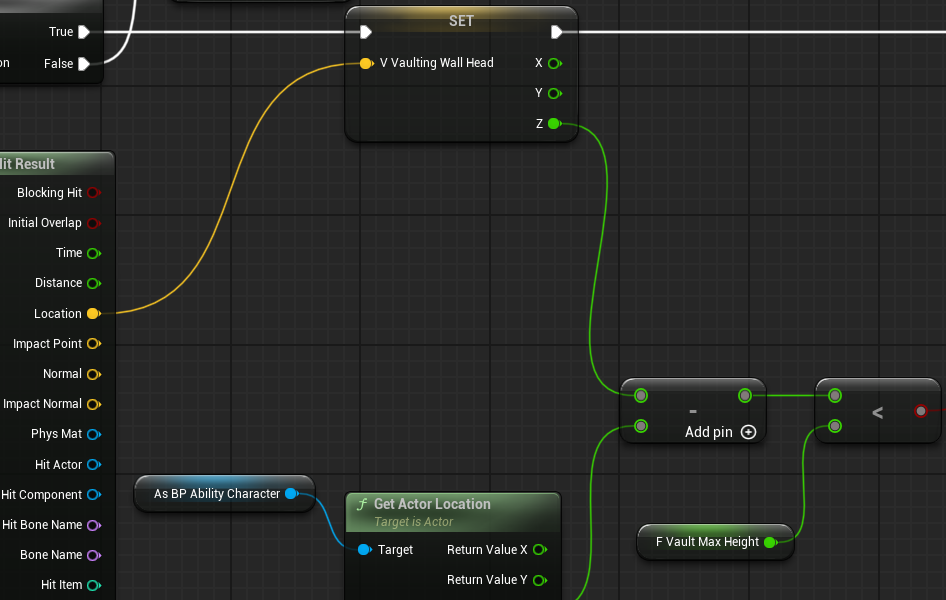
start from char’s position, ending in 70cm ahead&55cm below sight



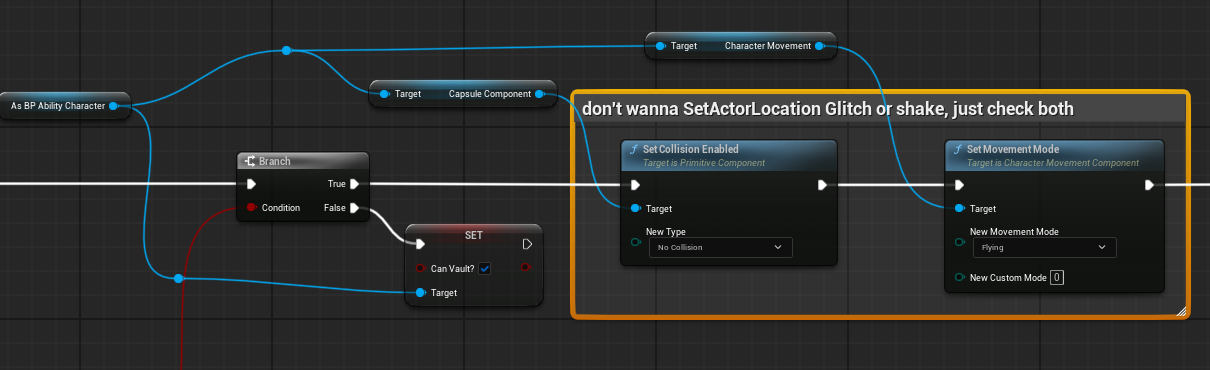
second trace: start from 200cm up,wall’s facing’ 10cm ahead

end in 40cm INSIDE of the wall

#use the hitLocation as climbing point(or wallHead)

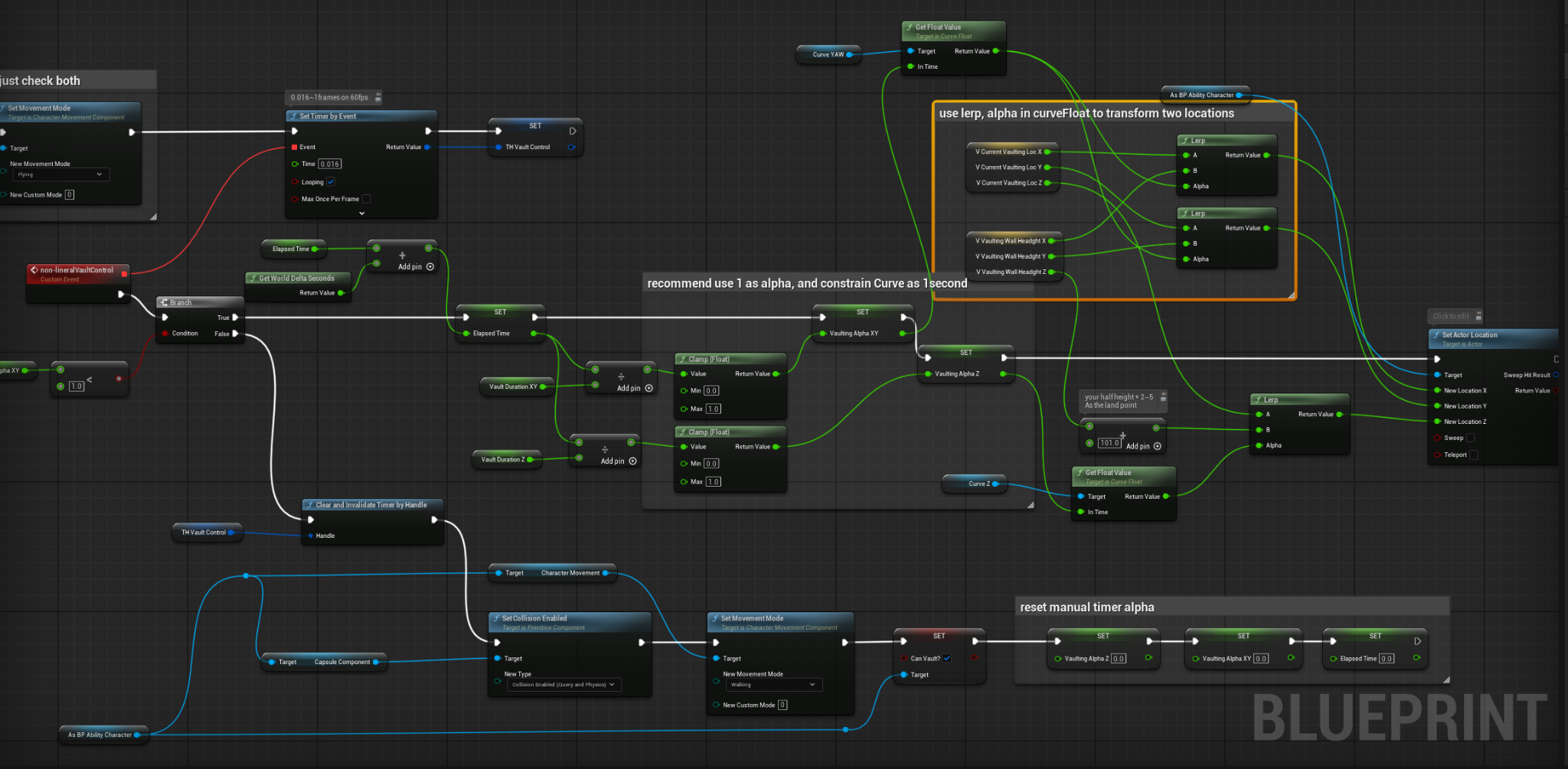


don’t forget compare the WallHeight, to control if “CANCLIMB”



Component Actor have no TIMELINE features.

16ms as oneFrame time in 60fps condition, but very useful for smooth Timer tick



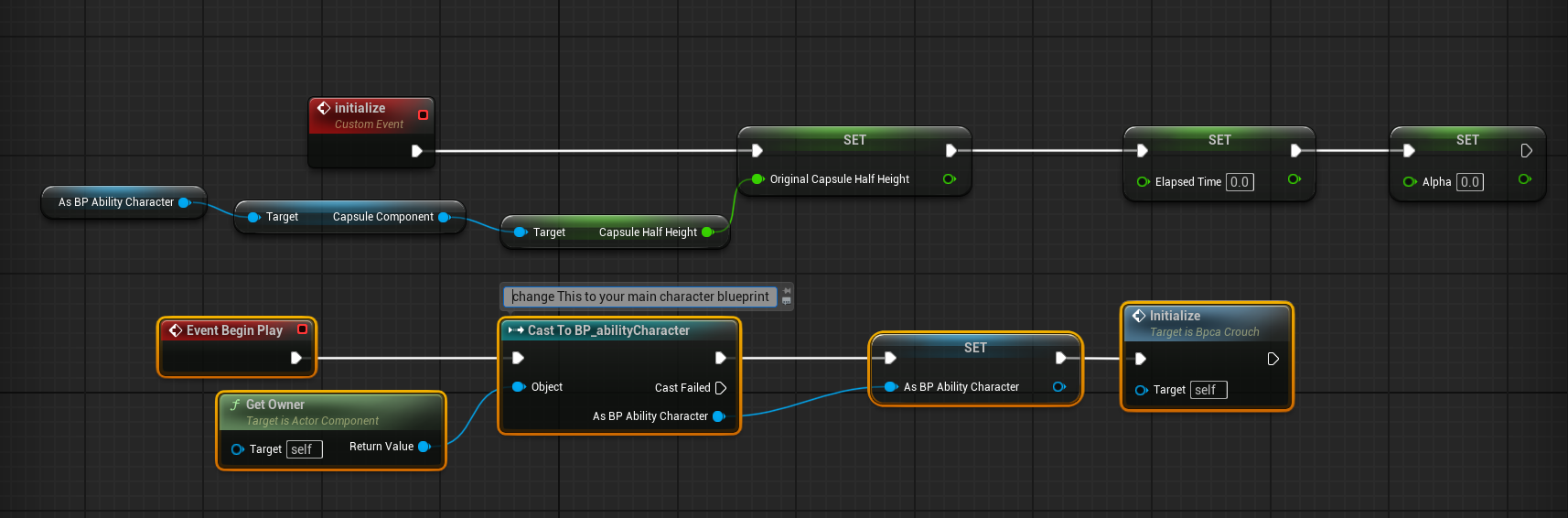
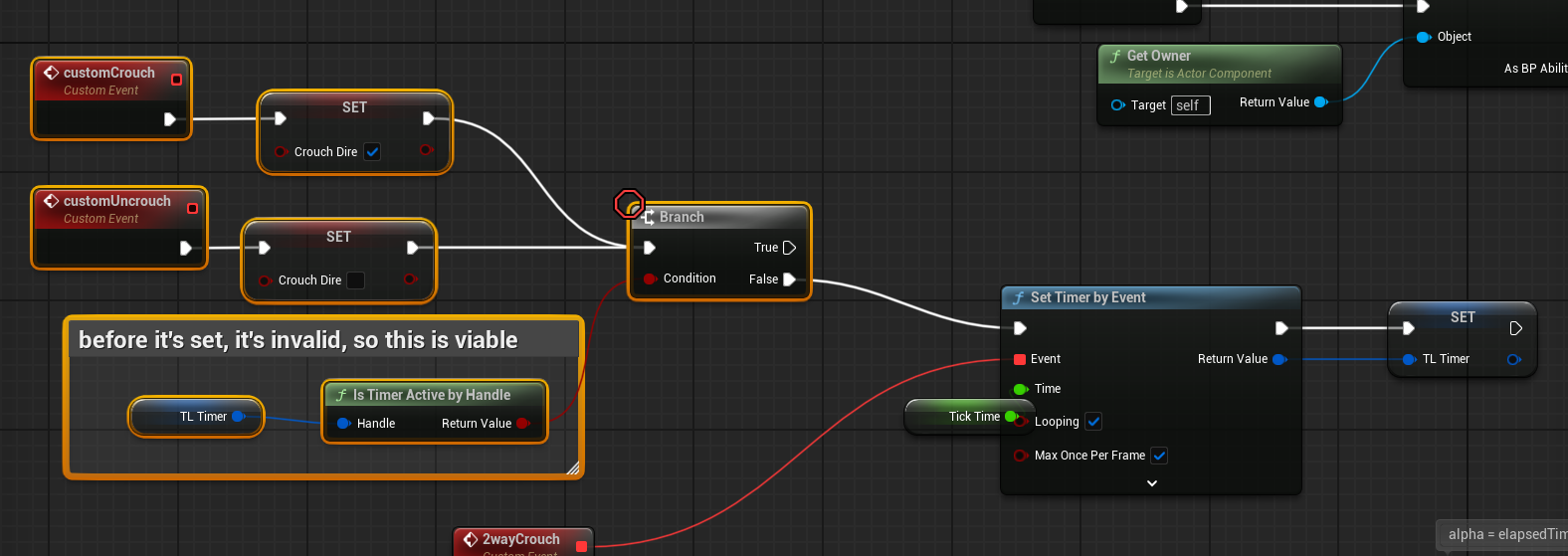
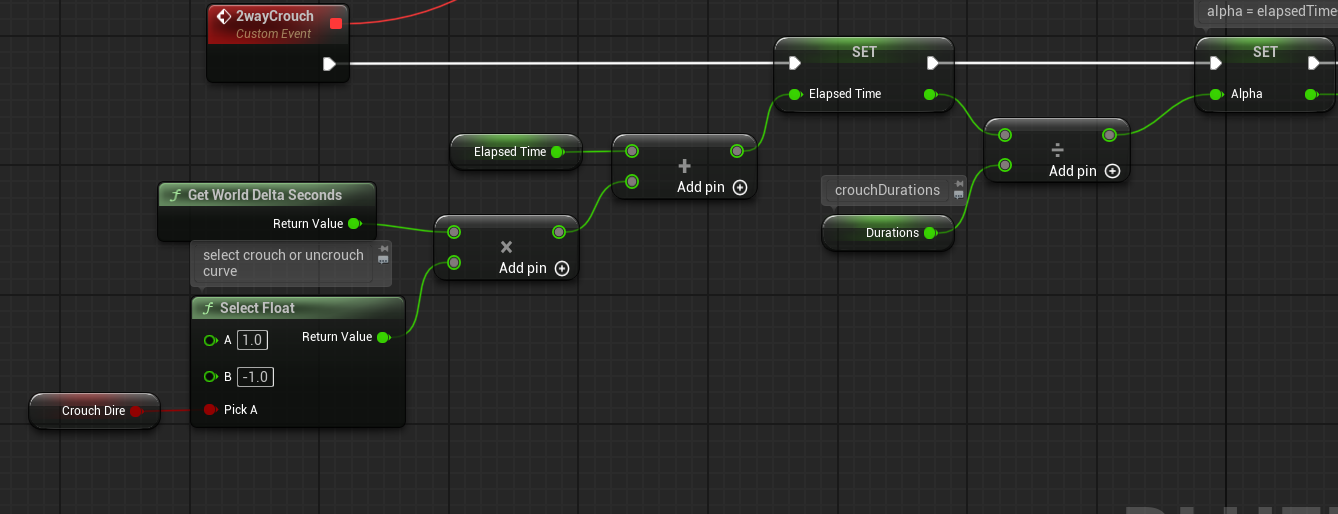
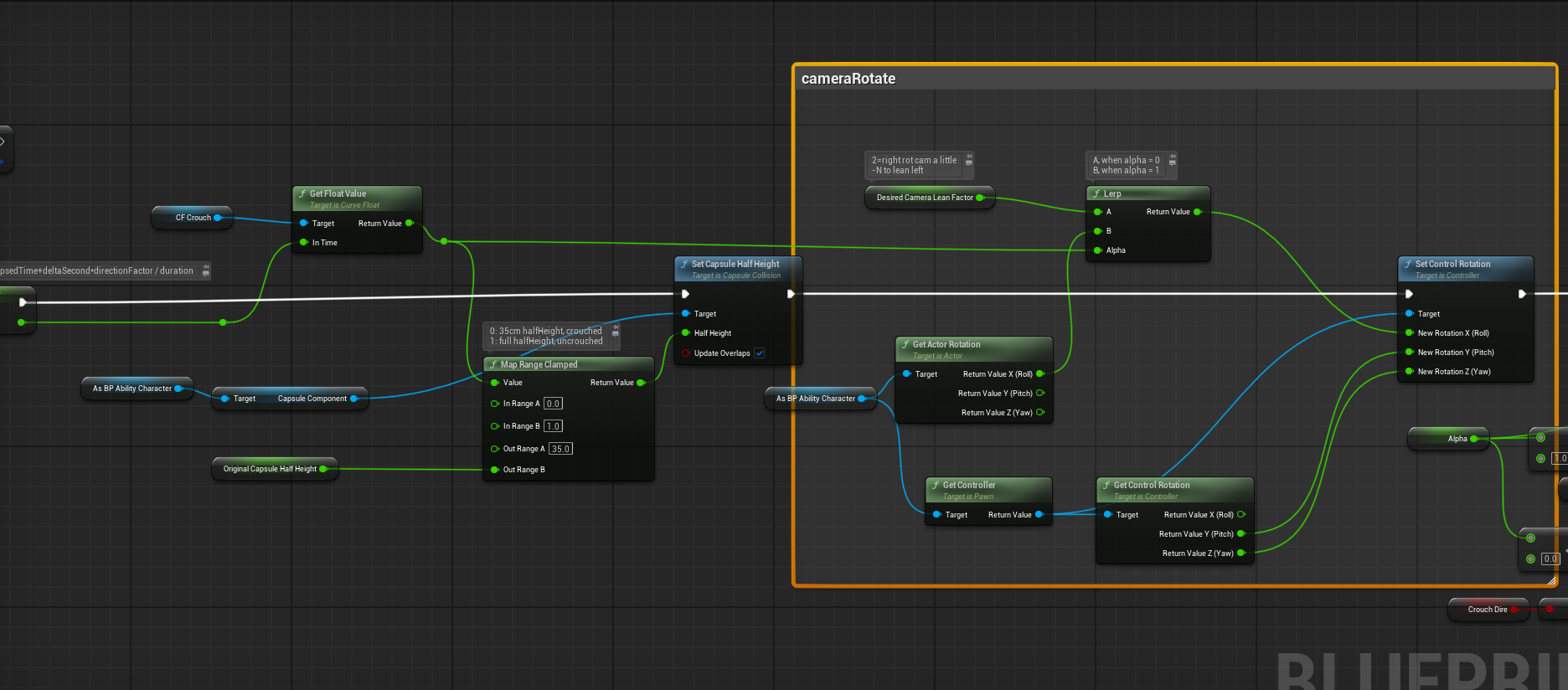
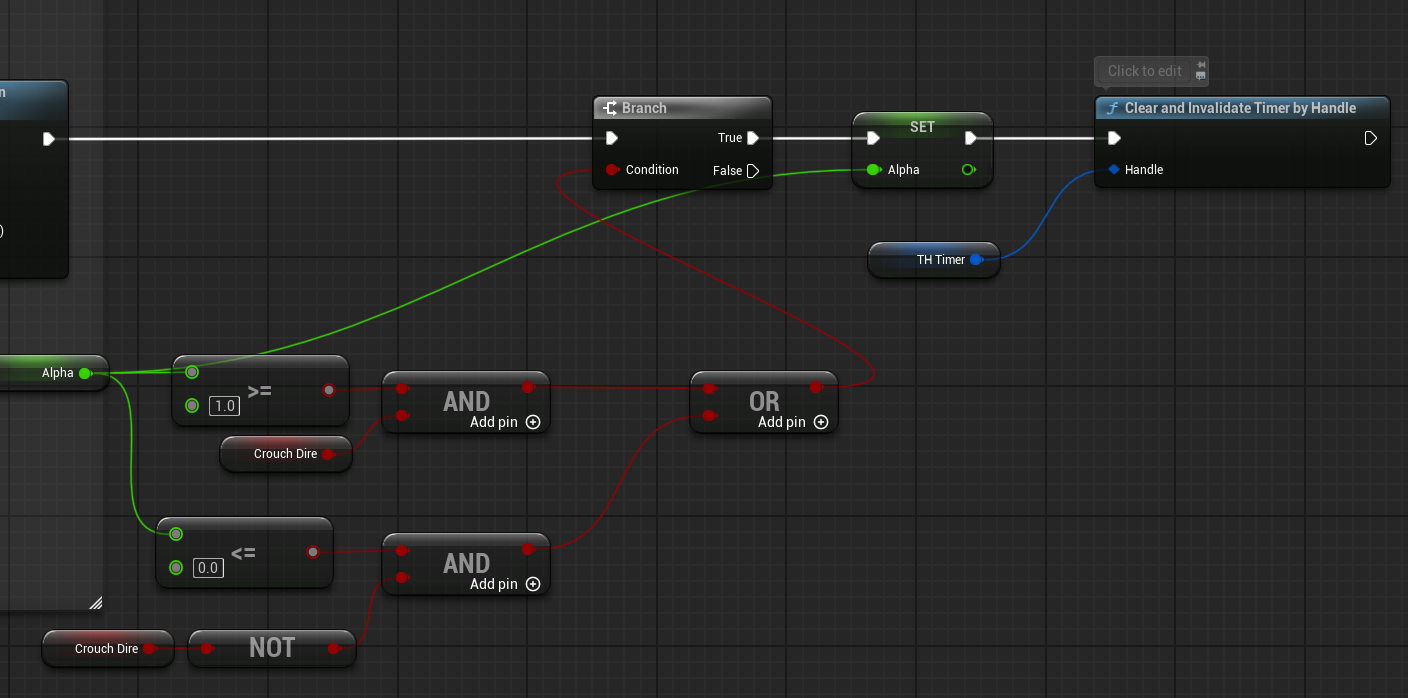
TL;DR:

use 2 FloatCurve, one for Z axis, should be quicker, another for XY, faster

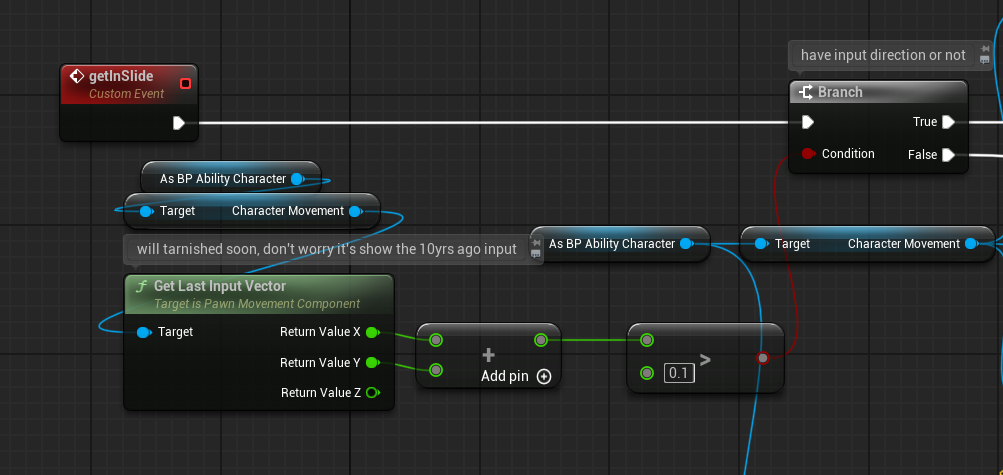
and using DeltaTimes/DurationTime=alpha, clamp into 0-1, use as 0-100% for a 1second Curve

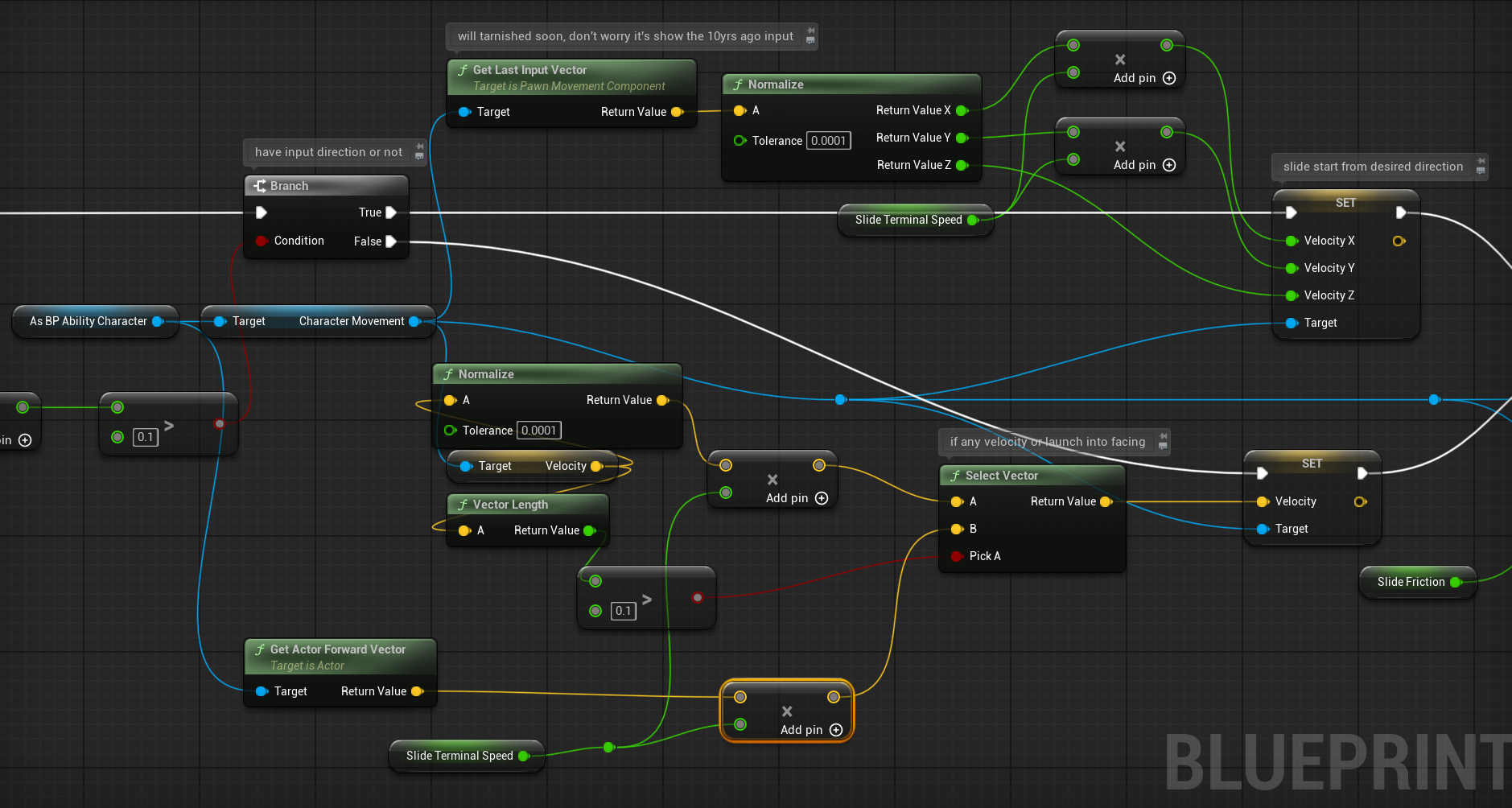
Non-lineralCrouch

crouch

1. initialize the ElapsedTime, alpha, halfHeight
2. as we using manually timeline
3. use bool and Select float as direction
4. **alpha = elapsedTime+deltaSecond\*directionFactor / duration**
5. **directionFactor either be 1or-1,**
6. 
7. rotate camera and set height
8. exiting condition : direction true & alpha ==1
9. or direction false && alpha==0

slide

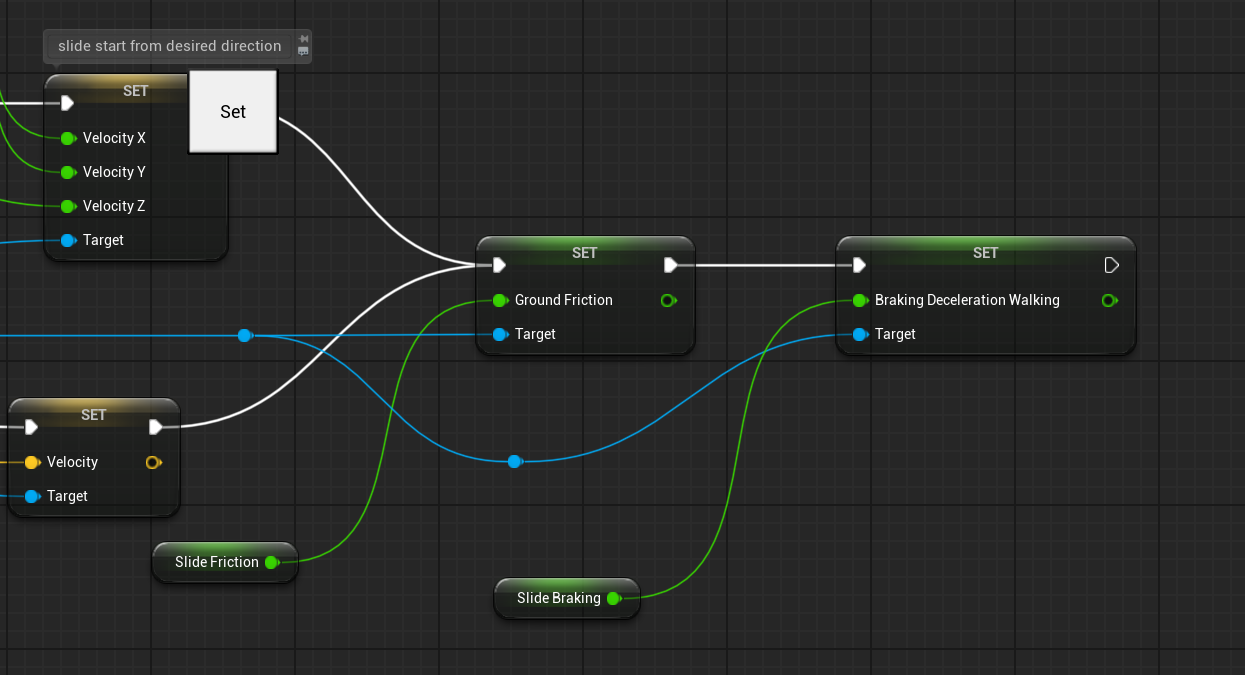
1.check if anyDirection input

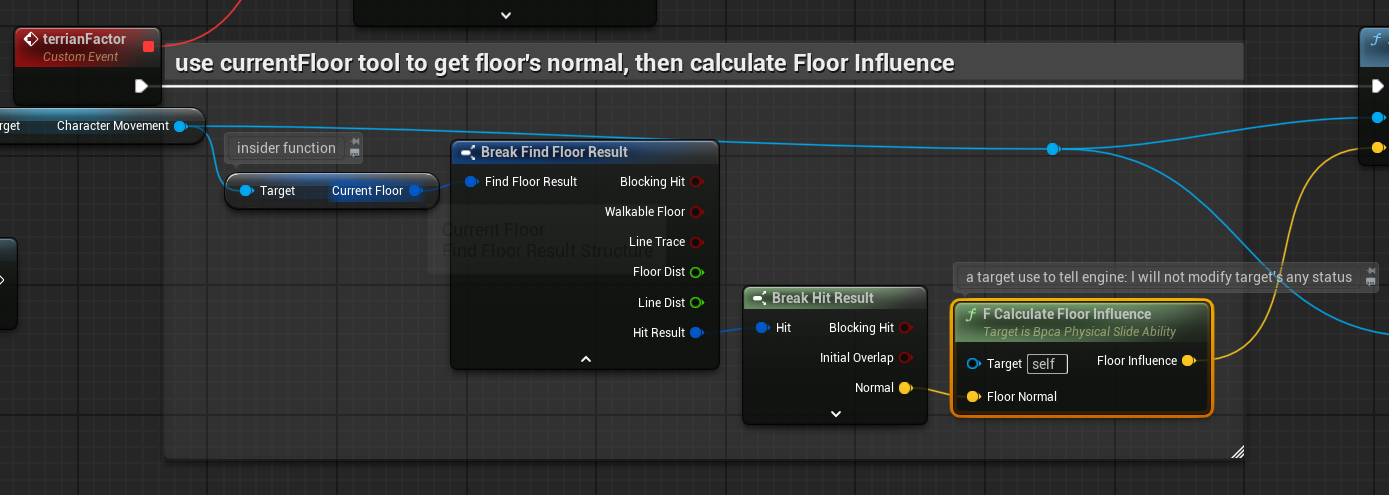
2.

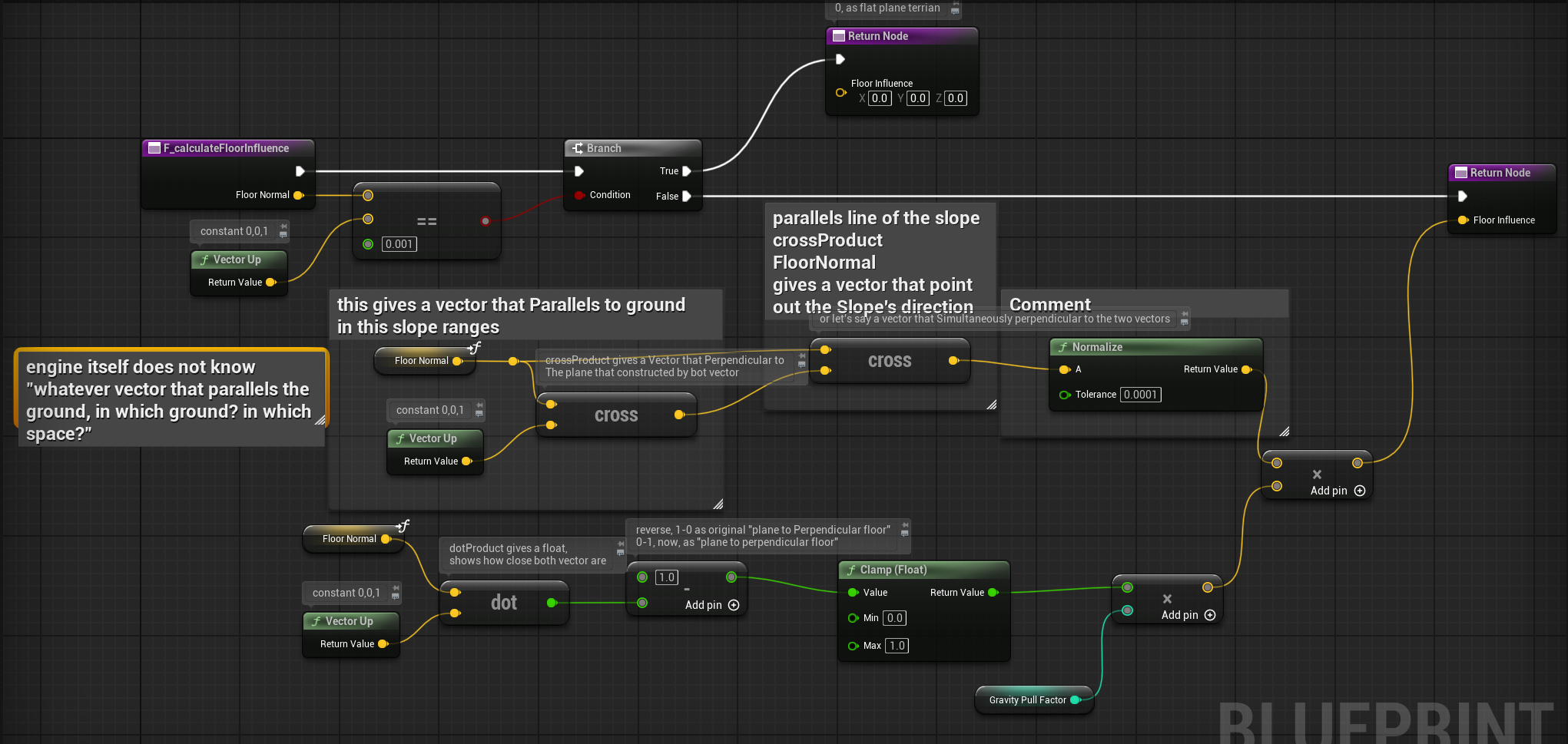
if any velocity or launch into facing

3. launch into it

4.secret of sliding: low friction\braking deceleration factor



5.

6.

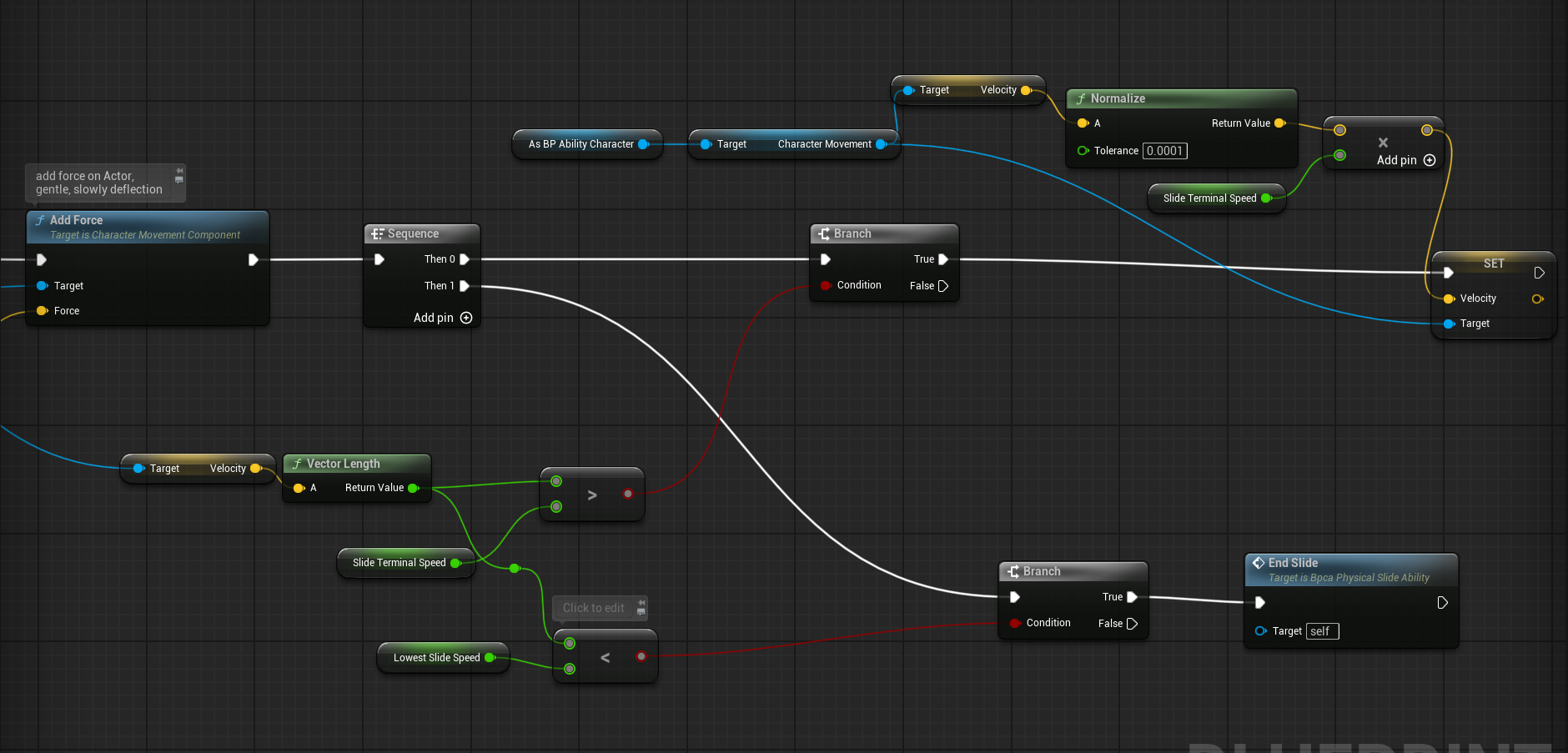
get a vector that parallels the ground of the slope zone

use crossProduct, floorNormal x up vector as v\_ground

than floorNormal x v\_ground to get vector that parallels the slope.

get influence factor of floorNormal to vectorUp via dotProduct,

1-factor to normalize as 0-1, shows how floorNormal close to the Perpendicular ground vector



7.add force, on Actor,gentle, slowly deflection

combine with timer, very smooth,

8. exit condition: velocity too low

collisions

now the collisions are based on Lowest echelon

Ignore<overlap<block

when two collision Interact, one are Ignore, other are block, Ignore will be run.

manually Timeline

#no timeline in component actor, so we using timer

no matter how many curve float are using,

try to standardized **Alpha,**

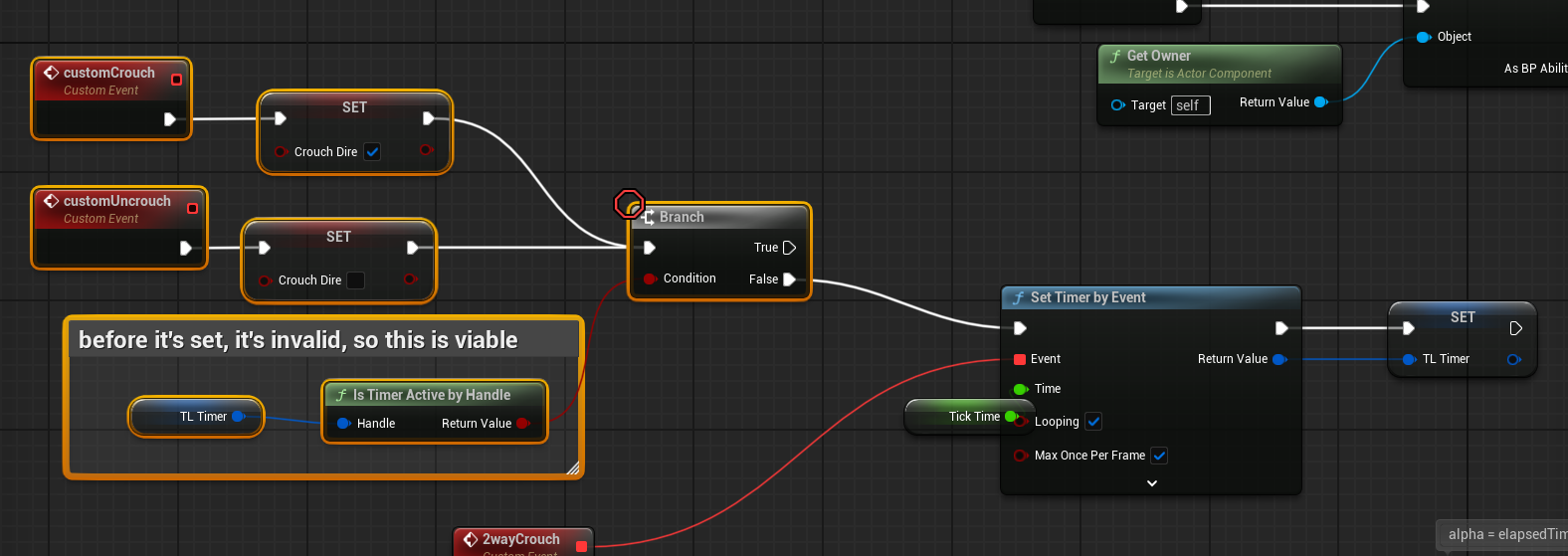
**alpha = elapsedTime+deltaSecond\*directionFactor / duration**

**directionFactor either be 1or-1,**

**Then in CurveFloat, try to nomalized in 1 second Curve (as 100%)**

**time pace Controlled by duration**

#note that timerHandle-invalidate only invalidate the handle, timer keep going



note that TimerHandle are invalidate before it’s set, so no reliability issue

**Make sure that alpha saved correctly, then should be fine**

movement control

#add force:

on actor, a gentle, not precise, slowly deflection

#set velocity:

very straight forward, combo with set **ground friction, braking decelerations**

#launch character:

quite same as set velocity

mysteriousReferences

object reference: point to some instant that already inside of the world

–spawn already, or inside the level

–can call the functions, lifeTime attached to actor

classReference: point to class, not instant

–**use to** spawn actor

–**defined** what i could spawn

—cannot call functions, or access the variables, **use to Spawn** only

soft reference: in Large projects,

– use to resolve lag\memory stress

– a nameOrDirections

– must be LoadAssets, or Async load

gearSystem

ID: like change name from AR into AR\_Mod 0

saves needs to know, …?

names - id - class

hierarchys

**Basic framework:**

bpca\_equipmentComponent

–to attach into \*\*character base blueprint\*\*

–manage “what to equip” by a **Enumeration variable**

**\*\*En\_propName\*\***

–forwarding “what to do with this Prop”

–doesn’t bind with How props’ work, how the animation plays

**An ability-like system, like others,**

**not operator of animations\input\meshes**

Hierarchys.

1.bp\_propBase

actor, that All Equipable props can Child by

weapons\props\usale item\whatever

**-overridable built-in functions**

-F\_equipProp

-F\_unequipProp

-F\_startFireProp

-F\_endFireProp

1-1.bp\_propC\_weaponBase

**child actor of bp\_propBase**

-fireRate variable

-bulletsType variable

-shoot logic

-specific weapons are also child actor of this actor

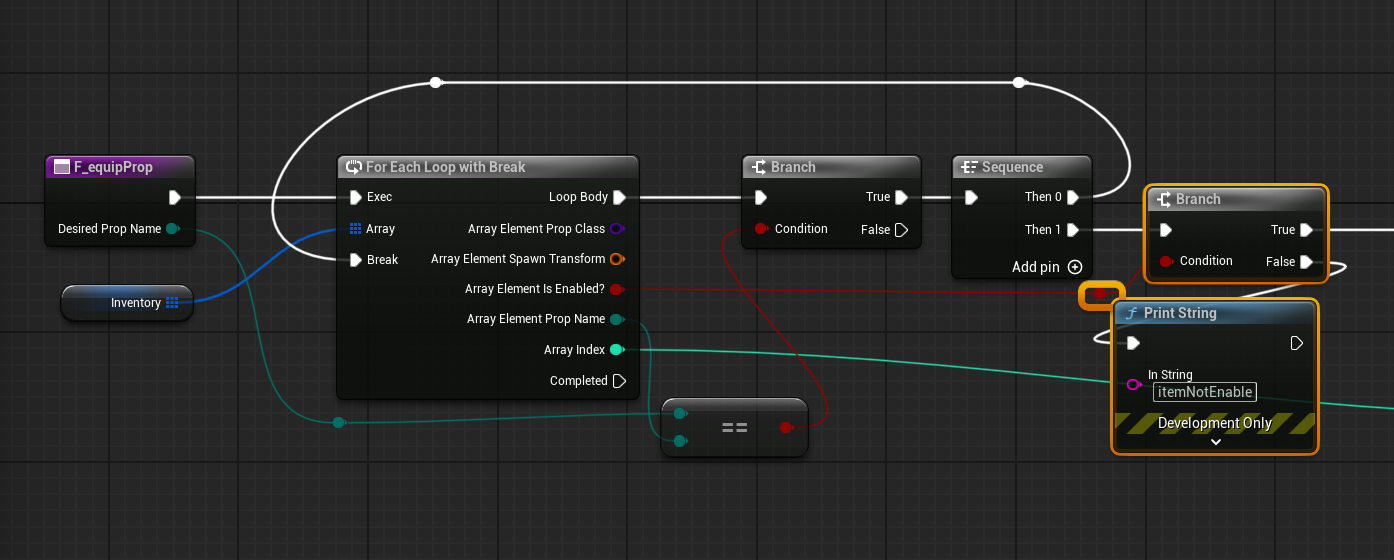
inventorySystems

An array variable inside of the bpca\_equipmentComponent

St\_propData as infos of props

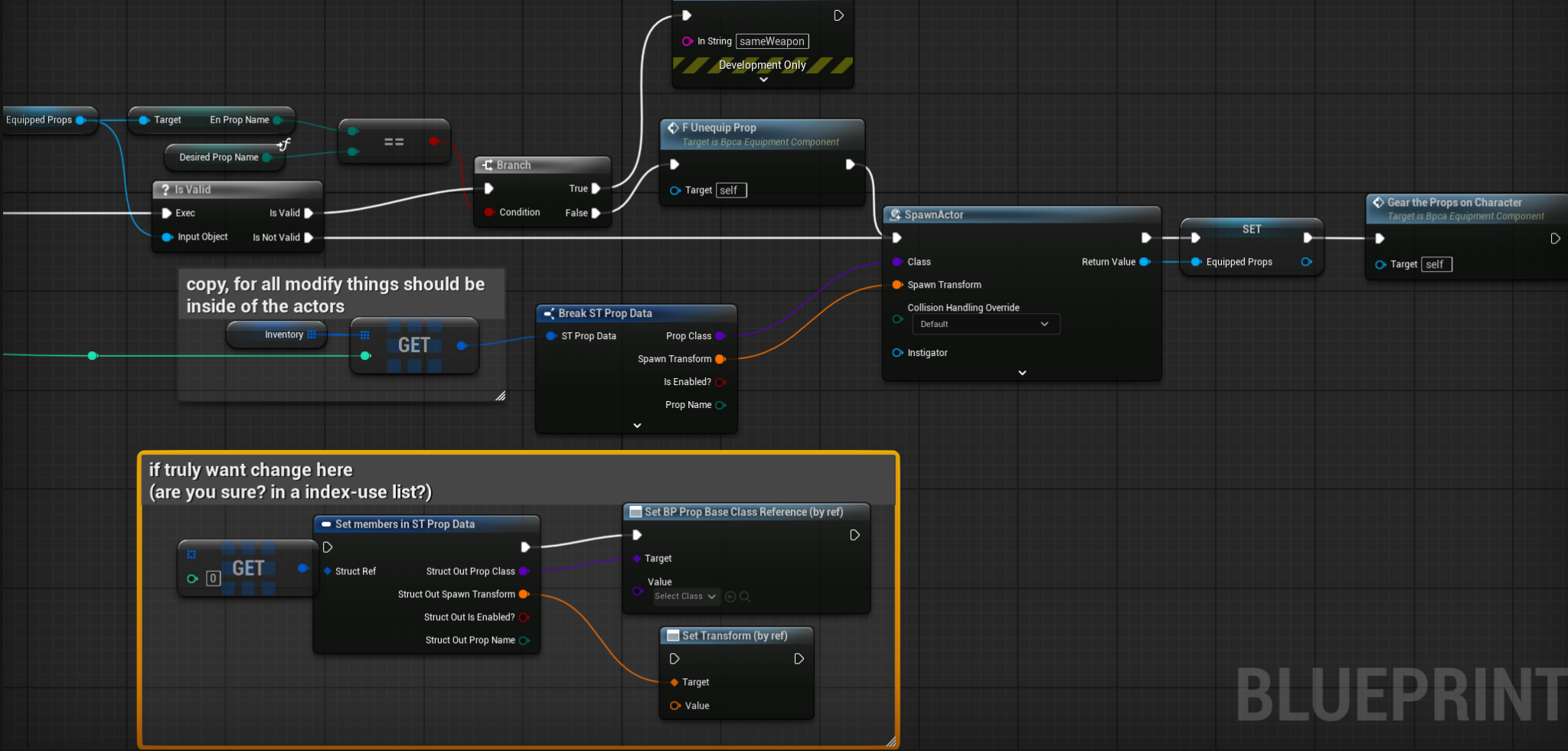
F\_equipProp

input: En\_propName

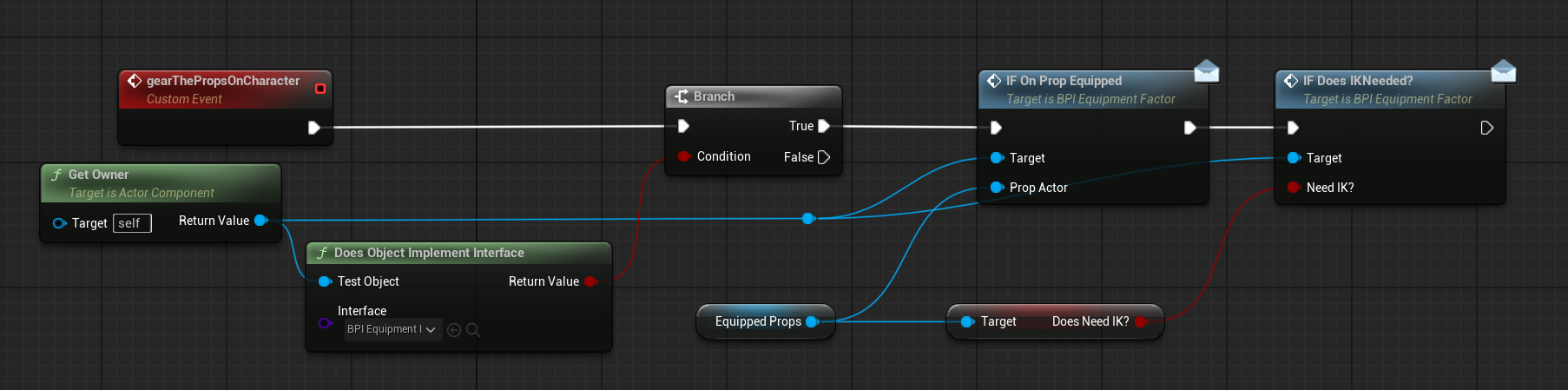
1.

traversal the array to find index by input EnumName

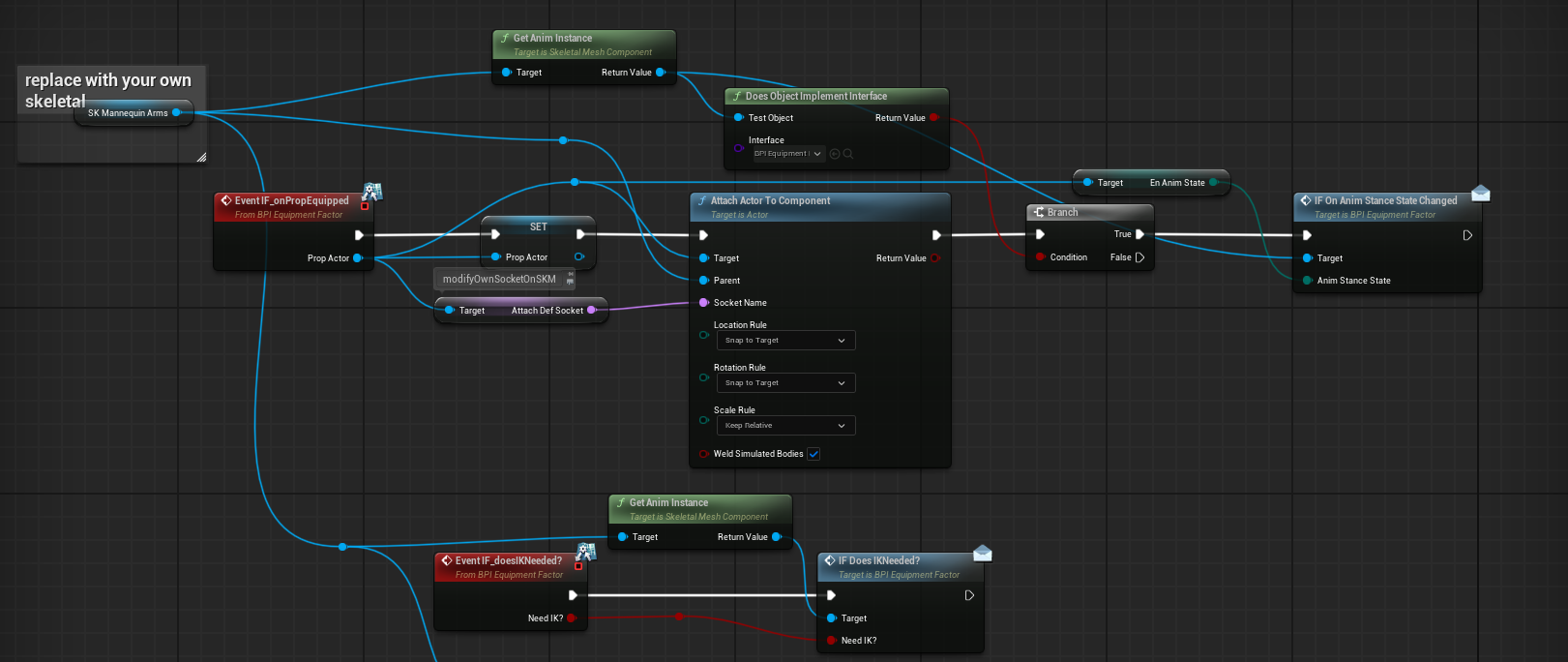
and verify if Enabled

  
compare if already have same props, and chose if UnequipCurrent or spawn immediately

gearThePropsOnCharacter

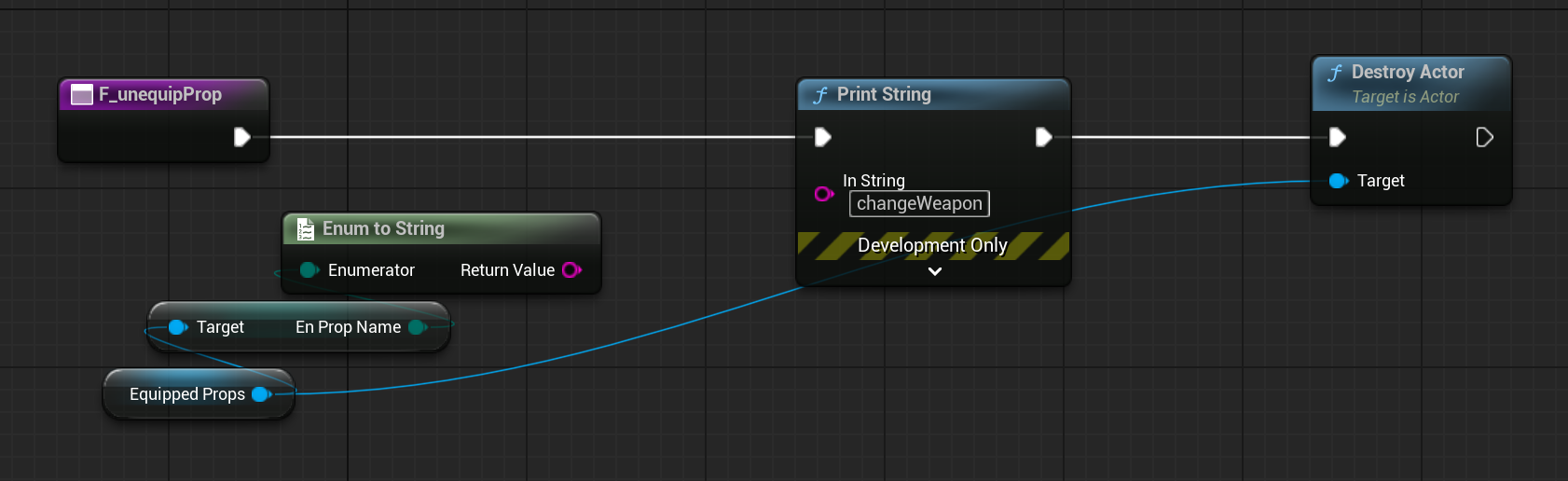


built-in Event, use interface function (input only) to call Owner, and passing propActor, and ik infos

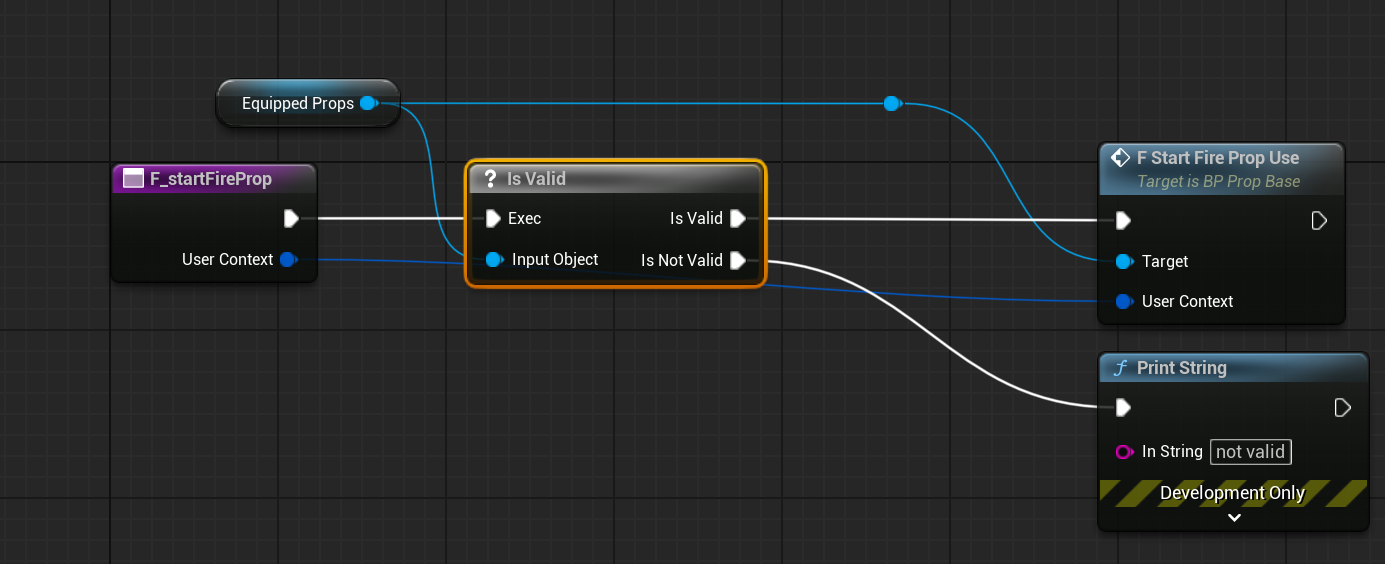


on actor that Actually calling the EquipEvent:  
notify via InterfaceFunction, and attach actor, send IK\AnimStance to animInstant

F\_unequipProp

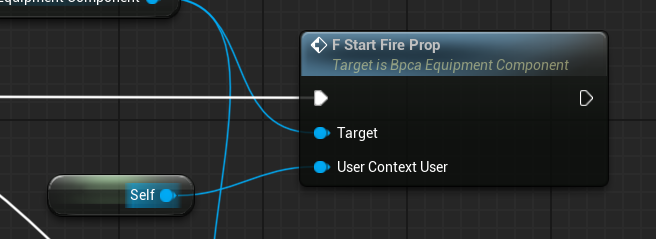


F\_start\end\FireProp



just a notify using,

passing the “trigger on\off” message to ChildActors



user Context contains like Actors.(owner)

can be use as owner, when spawn actor (bullets)

weapons

hierarchy：bp\_propC\_weaponBase

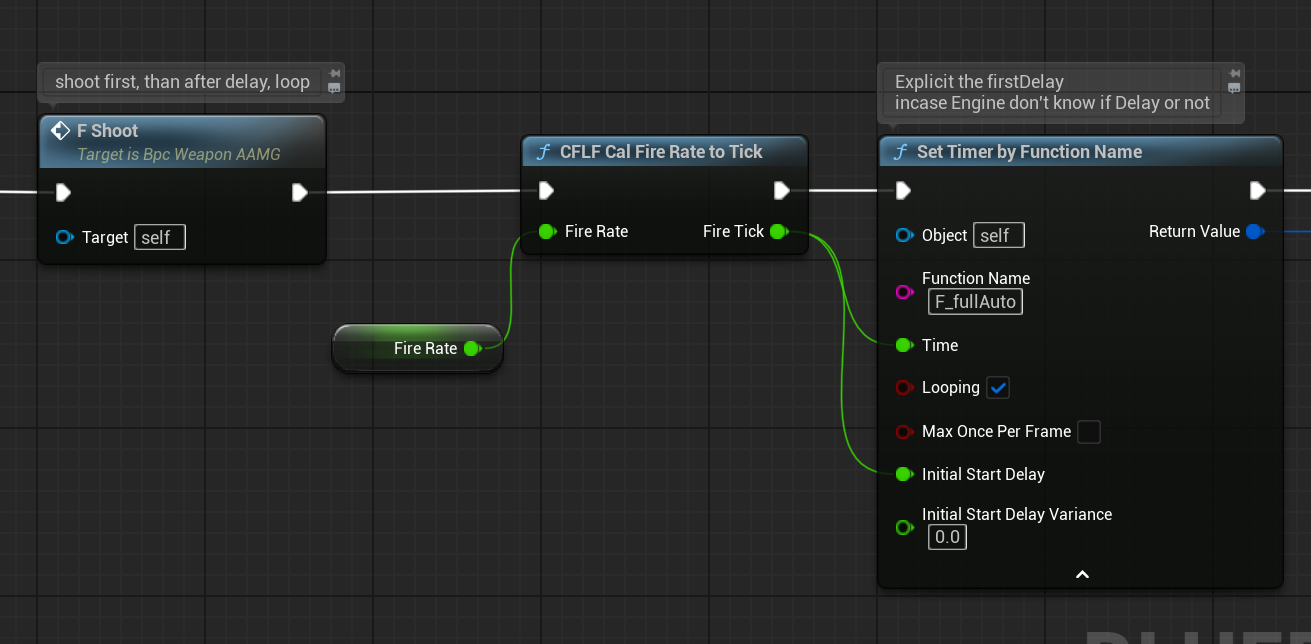
child actor-bp\_weaponHHG…

fullAuto

delay node in event, event is not inheritable,

yet, functions are time sens, no delay node allow

so use TimerByFunctionName to loop, to full auto

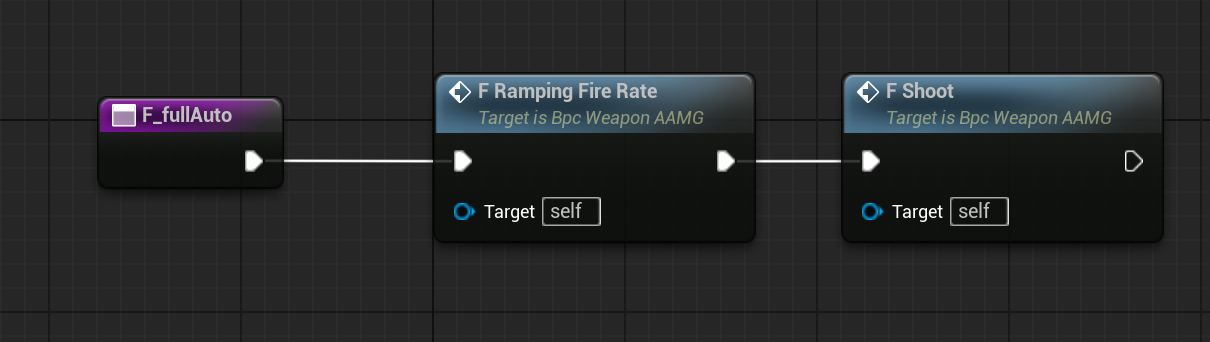


start delay are essential, remember first delay, shot first, then loop

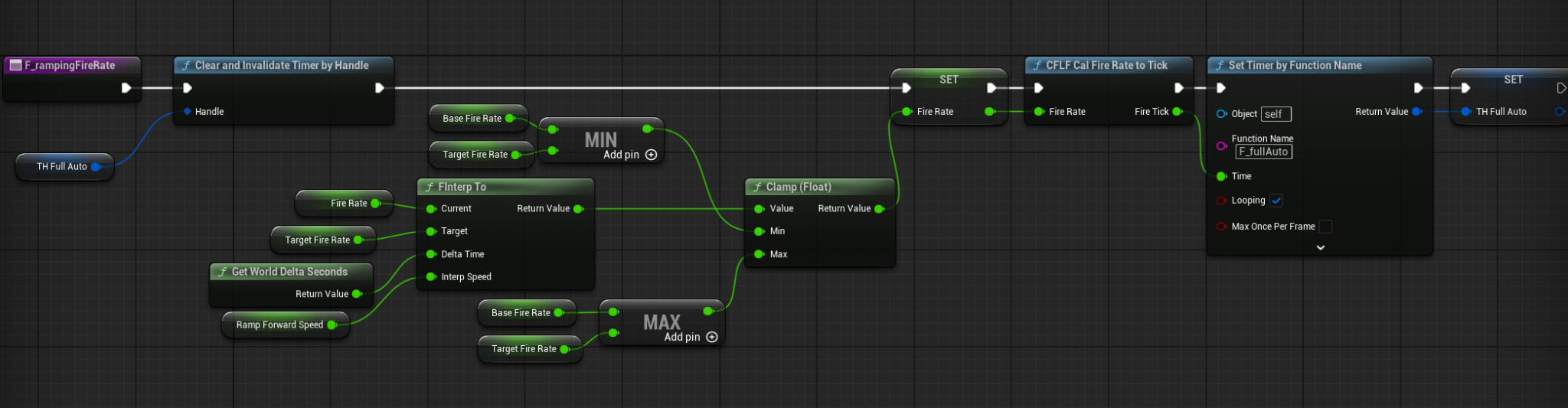
On FireReleased:

fireParaModify

some weapon Ramp-up the fireRate, some narrowing the Spray



on fullAuto or other notify,



use timer as well, but if effect same Timer, clear first

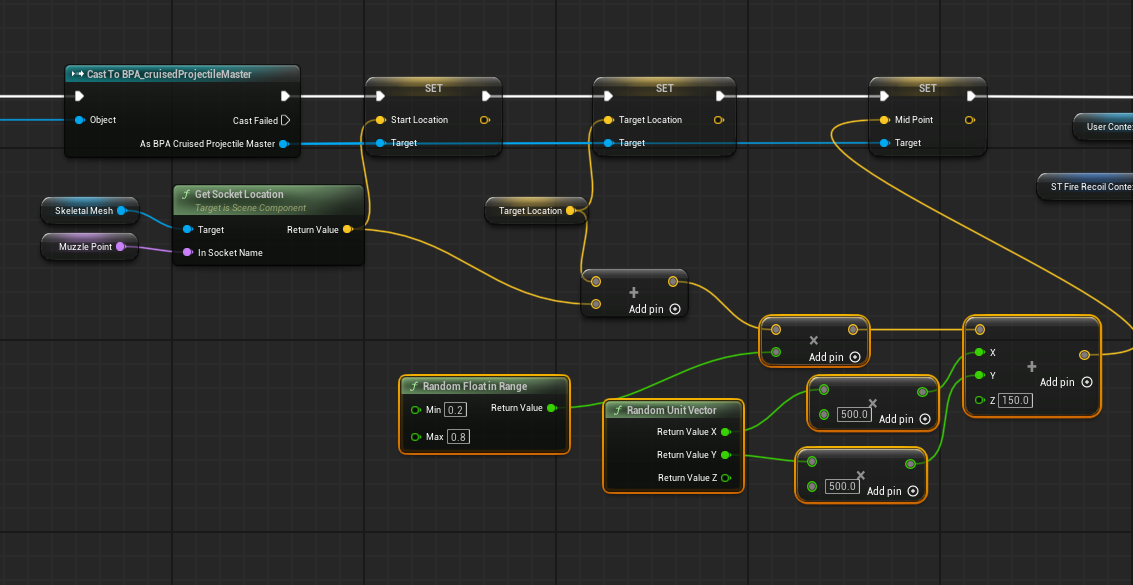
for time are locked since it launched

interp the current to target value, clamp after.

homing bullets

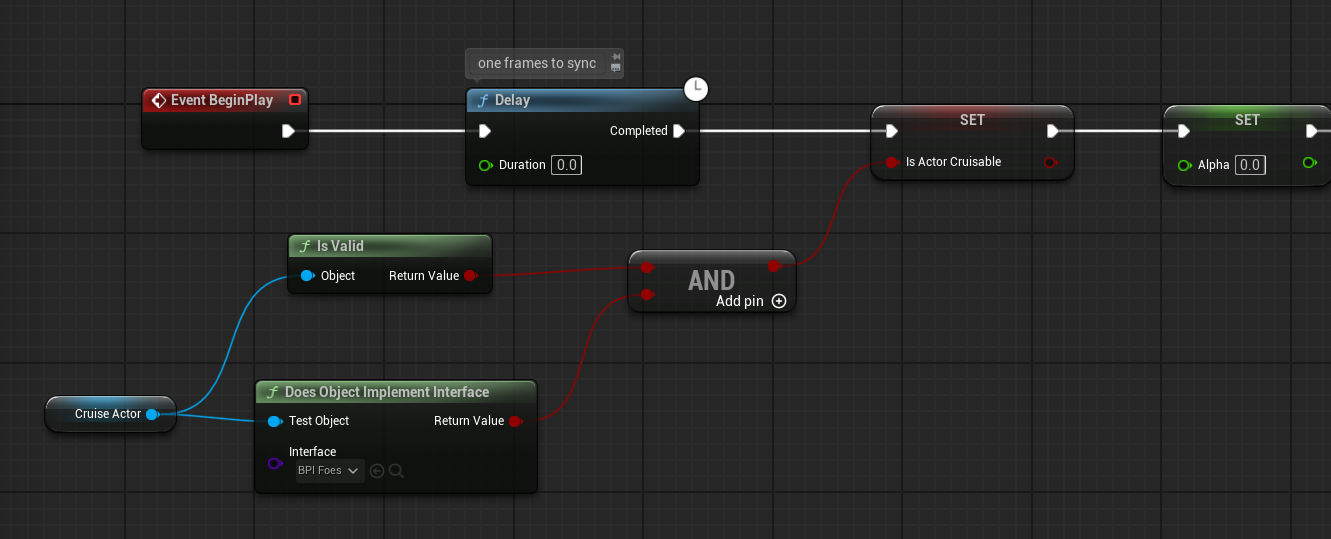
use Spline component and set actor location to trace Spline knot

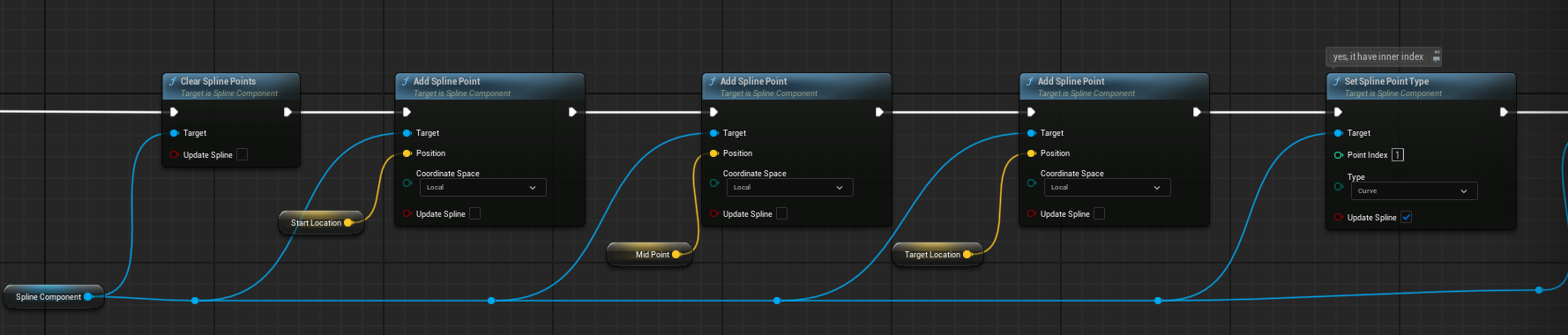
to pass the Spline point, either use InterfaceFunction, or cast to BP\_homingProjectilBase

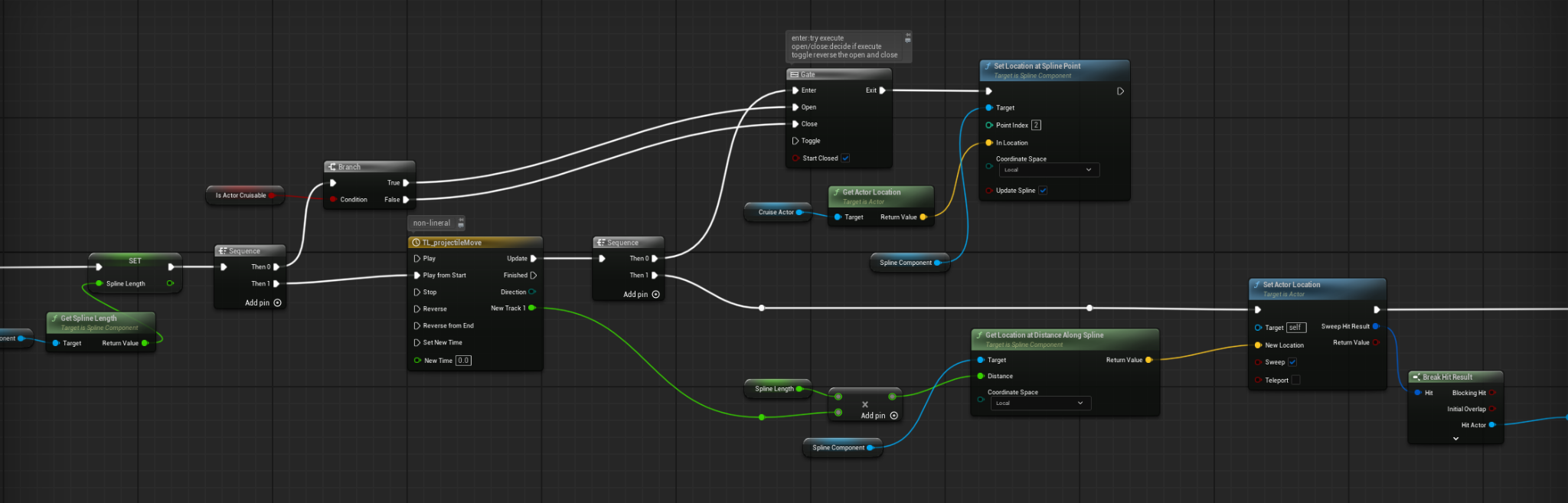


don’t forget use sphere trace, and pass the actor

inside the BP\_homingBulletBase



check if needs to trace



set three point, use timeLine to lerp trace speed, use gate to decide if needs to Trace

Animations

first person arms, so no need for layers blueprint interface or something

**Default slot** as Final point, to make sure animation montage use

others, **Arms\upperBody** can use as **LayeredBlend**

[State Machine / Locomotion] ↓

(cached Pose) ↓

[IK / Aim Offset / 修正] ↓

Slot "Arms" ↓

Slot "DefaultSlot" ↓

Output Pose