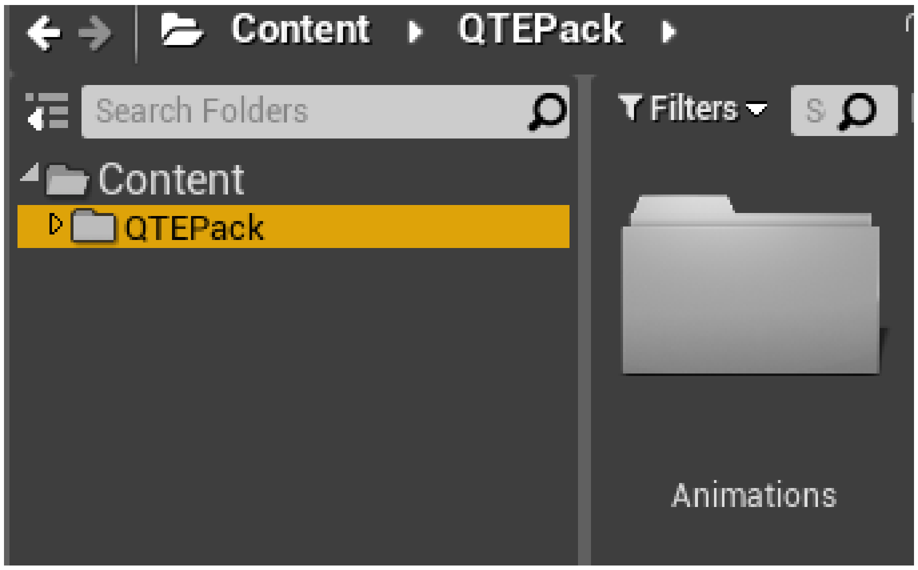
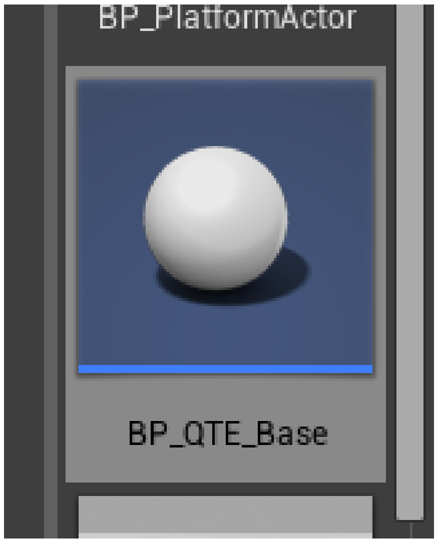
QTE System Documentation

made by dinozavr

to migrate QTE system to your custom project copy this QTEPack folder to your project's content folder

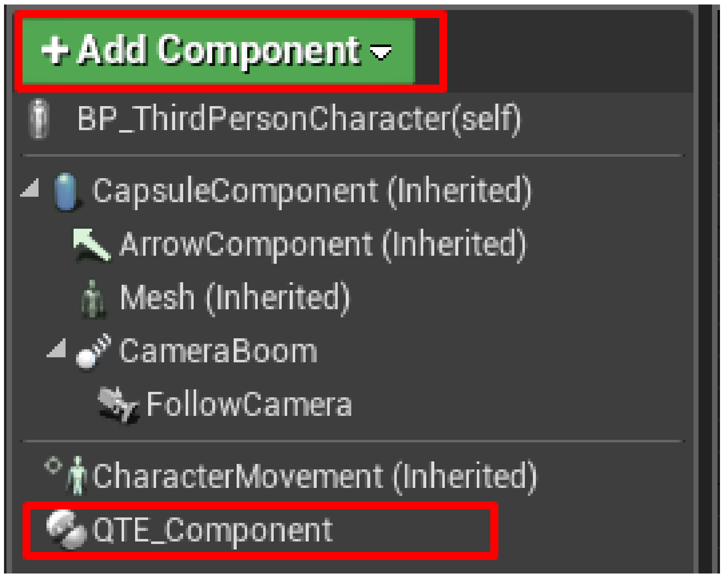


now add only 1 actor of this class to your level, all your levels where QTE is used must have only 1 actor of this class



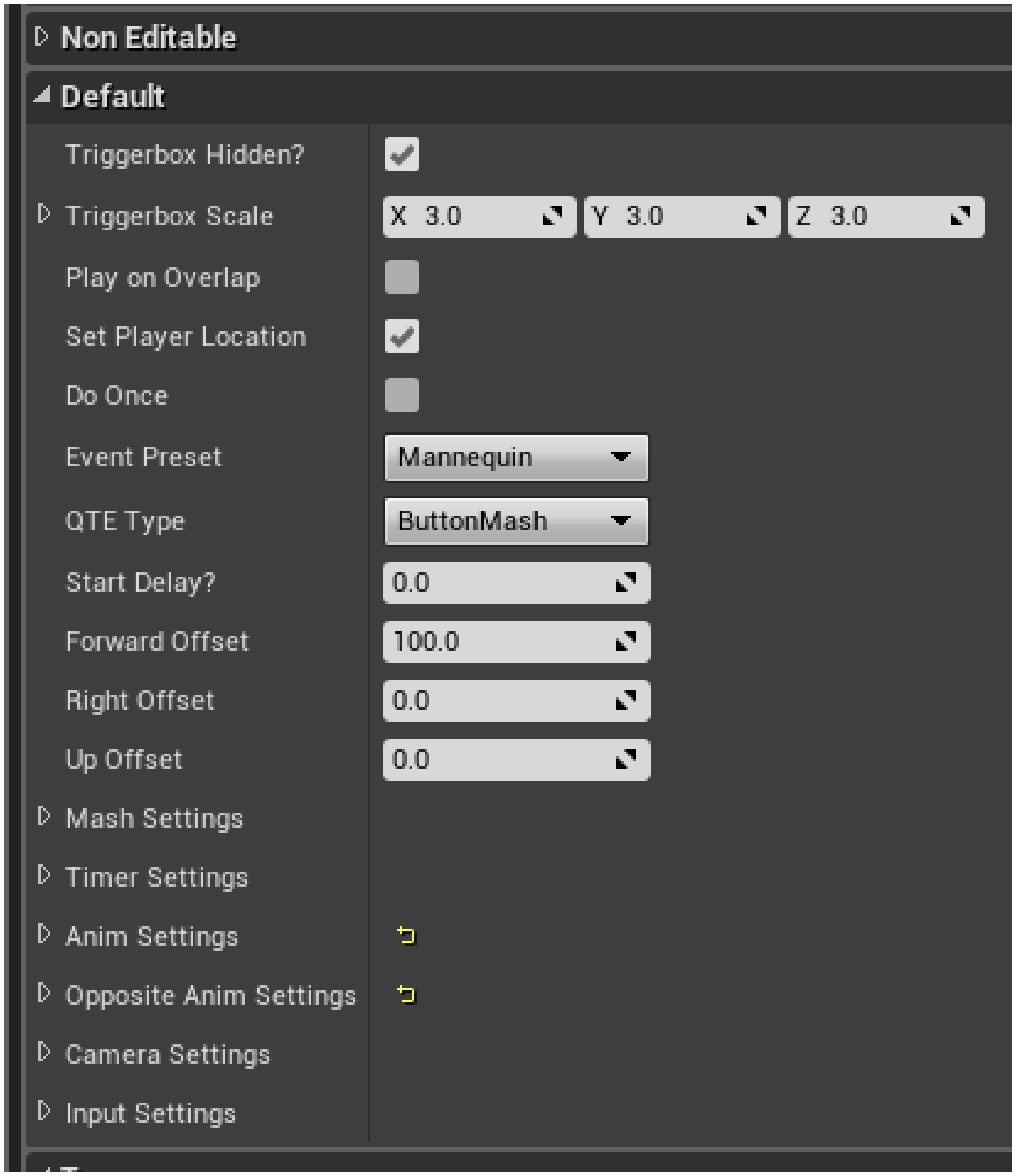
this actor does receive all input for QTE's, so when you press any button this actor will check it

now select any actor, you want to be QTE interactable, for example TPS character


+ Add Component
BP_ThirdPersonCharacter(self)
CapsuleComponent (Inherited)
ArrowComponent (Inherited)
Mesh (Inherited)
CameraBoom
FollowCamera
CharacterMovement (Inherited)
QTE_Component

add an QTE_Component to your actor, now you can engage a (quick time events) with this actor

there is various settings i'll try to explain them all

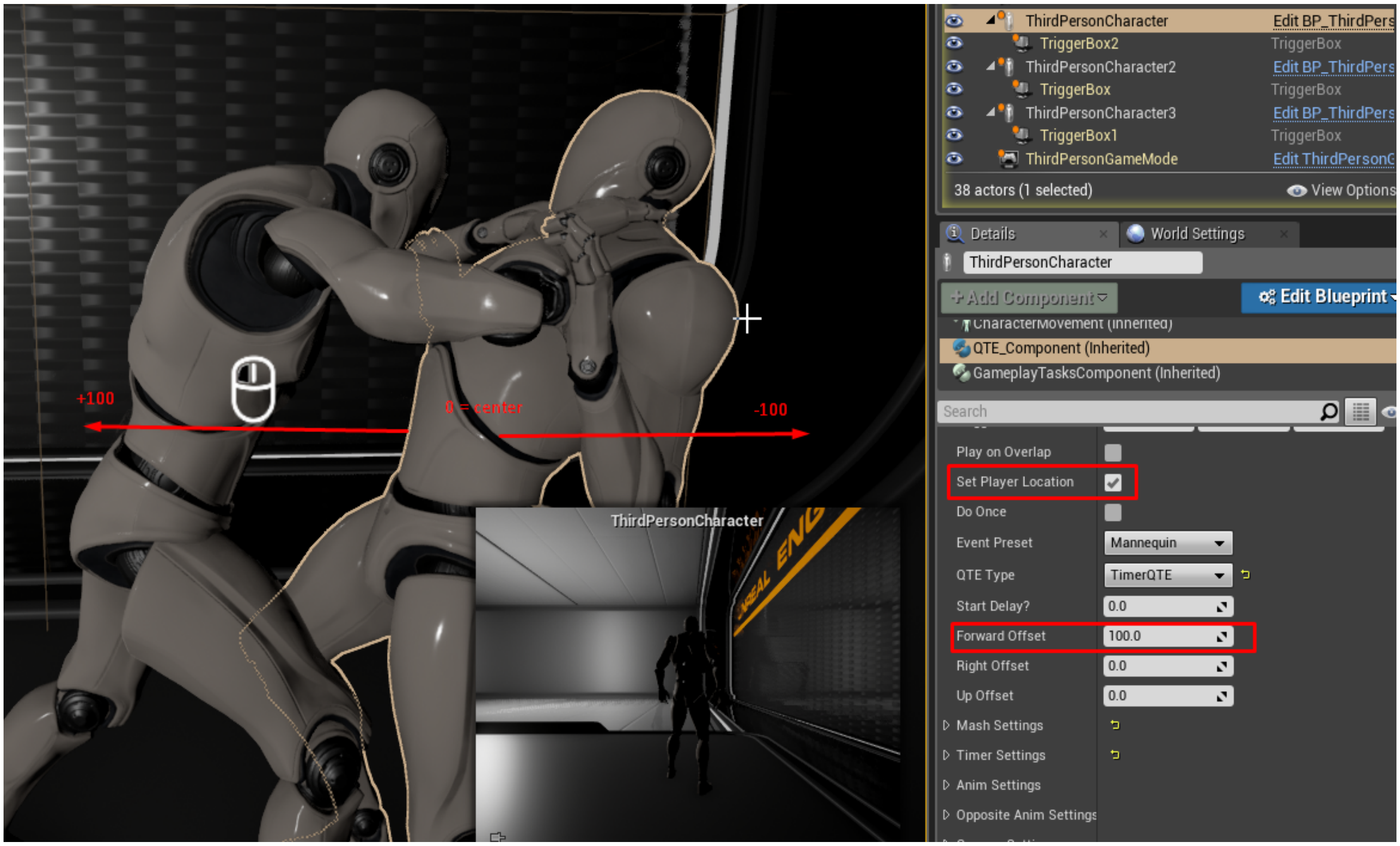


Non Editable	
Default	
Triggerbox Hidden?	✓
Triggerbox Scale	X 3.0 Y 3.0 Z 3.0
Play on Overlap	☐
Set Player Location	✓
Do Once	☐
Event Preset	Mannequin
QTE Type	ButtonMash
Start Delay?	0.0
Forward Offset	100.0
Right Offset	0.0
Up Offset	0.0
Mash Settings	
Timer Settings	
Anim Settings	
Opposite Anim Settings	
Camera Settings	
Input Settings	

you can hide and unhide trigger box for debug set trigger box scale(it attachs to owning actor) play on overlap, will engage QTE when player enters trigger box(otherwise we enter box and press button.

set player location, interps player to qte actor + with additive offsets(forward,right,up) to fix position

so like, set player location +100 forward
interps us to face enemy while choking him
to fix animation view



ThirdPersonCharacter	Edit BP_ThirdPers
TriggerBox2	TriggerBox
ThirdPersonCharacter2	Edit BP_ThirdPers
TriggerBox	TriggerBox
ThirdPersonCharacter3	Edit BP_ThirdPers
TriggerBox1	TriggerBox
ThirdPersonGameMode	Edit ThirdPersonG

38 actors (1 selected)

Details — ThirdPersonCharacter

CharacterMovement (Inherited)
QTE_Component (Inherited)
GameplayTasksComponent (Inherited)

Play on Overlap	☐
Set Player Location	✓
Do Once	☐
Event Preset	Mannequin
QTE Type	TimerQTE
Start Delay?	0.0
Forward Offset	100.0
Right Offset	0.0
Up Offset	0.0

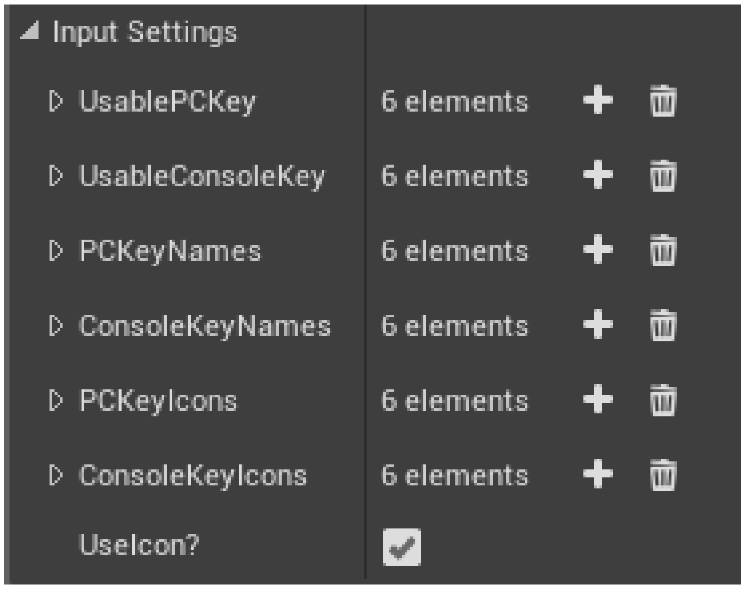
Mash Settings
Timer Settings
Anim Settings
Opposite Anim Settings

do once bool, will do QTE once if you finish,
you can't play it again but if you fail you can try
again

Start Delay will start QTE sequence after given
delay, so you can't fail or success until it ends
animations will ignore this delay,
(so example, QTE buttons will show only when we
start choking, not instantly)

QTE have 2 types, button mashing, and Timer sequence,(timer have non-timer version also) they both have their struct settings, with dozens of editable variables.

i’ll show first necessary things, like Input struct



Input Settings

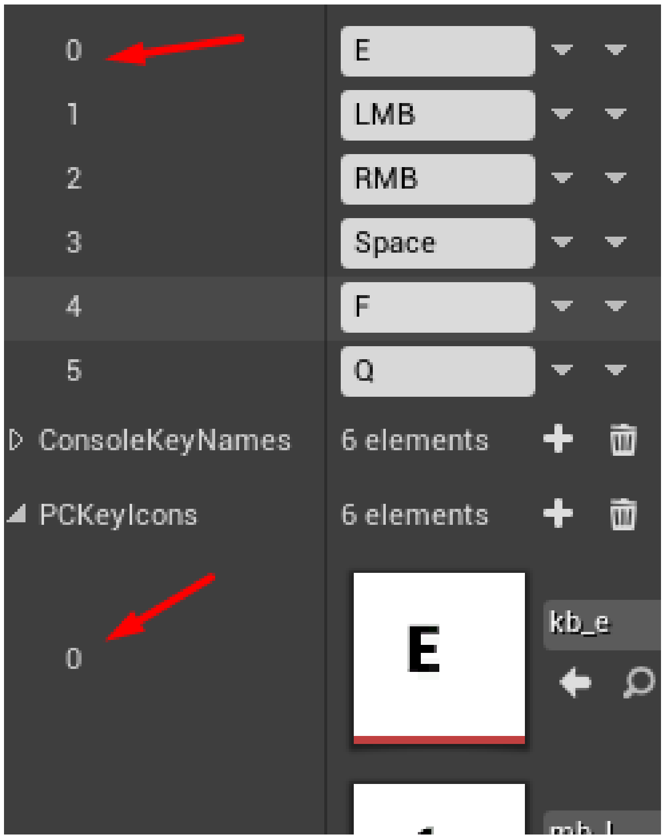
UsablePCKey	6 elements
UsableConsoleKey	6 elements
PCKeyNames	6 elements
ConsoleKeyNames	6 elements
PCKeyIcons	6 elements
ConsoleKeyIcons	6 elements
UseIcon?	✓

input settings store, usable PC and Console keys, with their display names and Icons aswell their index are equal

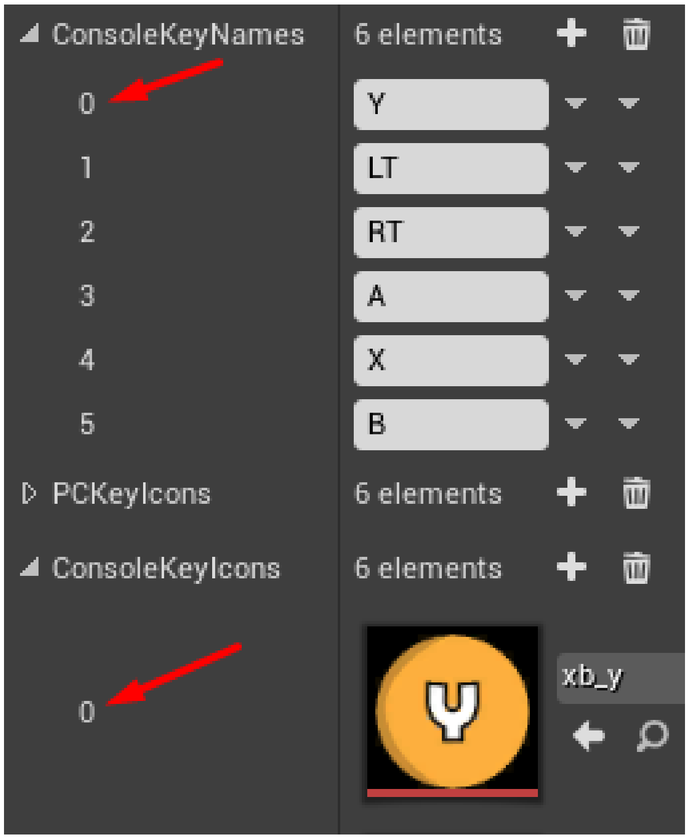
so for example PC key index 0 is E
Console key index 0 is Y(xbox)
they both engage QTE, so they work in same way

you can add new keys to array to expand usable keys in qte sequence

Display name will be used if UseIcons bool is disabled otherwise icons will be used gamepad is auto-recognized so when you press gamepad button, QTE_Base actor will swap all input to gamepad ones from array



0	E
1	LMB
2	RMB
3	Space
4	F
5	Q
ConsoleKeyNames	6 elements
PCKeyIcons	6 elements
0	kb_e

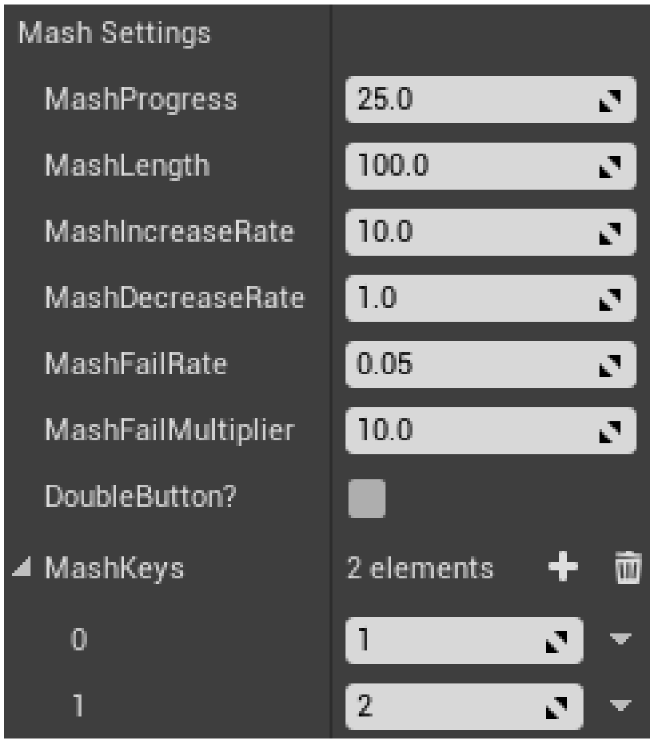


ConsoleKeyNames	6 elements
0	Y
1	LT
2	RT
3	A
4	X
5	B
PCKeyIcons	6 elements
ConsoleKeyIcons	6 elements
0	xb_y

make sure the index order is right

you can add whatever many buttons you want separately for PC and Console layout

Now ButtonMash settings



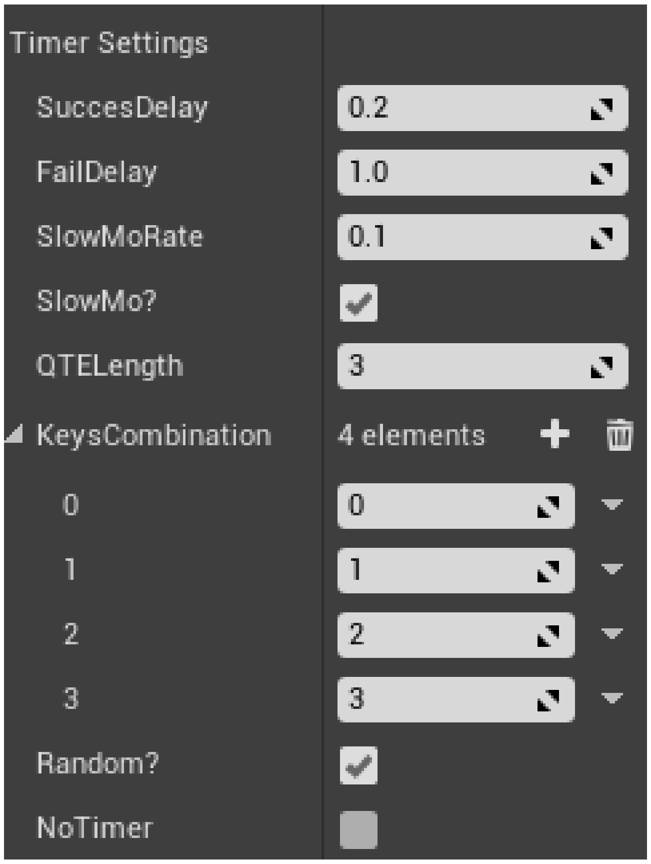
Mash Settings

MashProgress	25.0
MashLength	100.0
MashIncreaseRate	10.0
MashDecreaseRate	1.0
MashFailRate	0.05
MashFailMultiplier	10.0
DoubleButton?	☐
MashKeys	2 elements
0	1
1	2

-MashProgress - starting progress of bar.
-MashLength is the actual length of bar.
-Progress bar increases each time you press right button by IncreaseRate.
-Each MashFailRate(seconds), bar decreases by MashDecreaseRate.
-Each time we press wrong button, bar decreases by FailMultiplier

-Double button bool makes need to press 2 buttons in sequence, one by one, otherwise only 1 button is used.
-Buttons is going from MashKeys array (equal to Input Settings),
first and second button it is(1)LMB and (2)RMB by index from input struct

Now Timer settings



Timer Settings

SuccesDelay	0.2
FailDelay	1.0
SlowMoRate	0.1
SlowMo?	✓
QTELength	3
KeysCombination	4 elements
0	0
1	1
2	2
3	3
Random?	✓
NoTimer	

-Succes delay, this delay happens each time we press right button, before next will be shown

-FailDelay, QTE will be failed if we don't press any button after that delay pass(doesn't work if NoTimer bool is active)

-SlowMo bool slows, the time while input button is shown

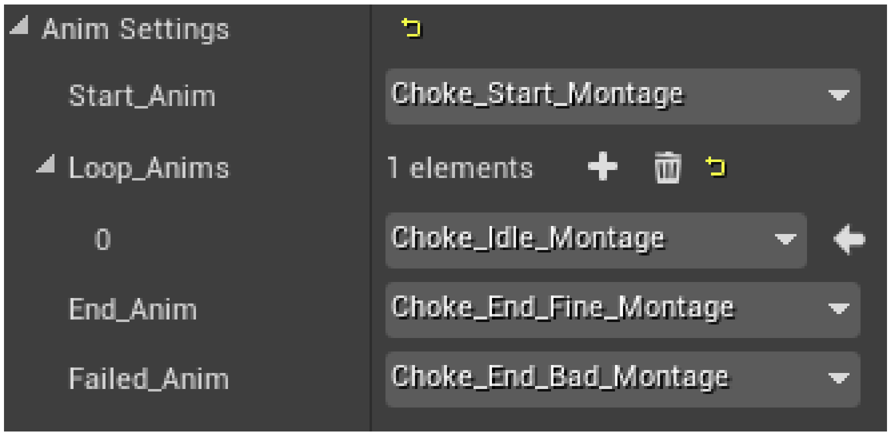
-QTE Length, is the actual length of QTE, how much right buttons we need to press to win QTE

-if random bool active, buttons will go in random order from Input settings array (gamepad or PC)

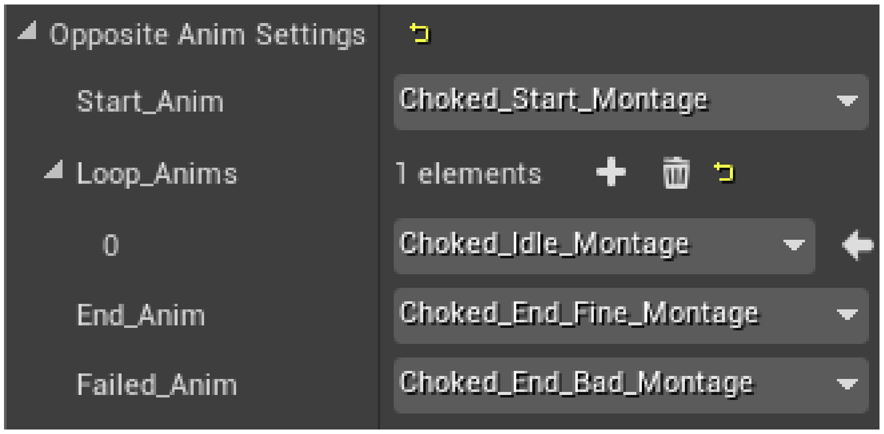
SlowMoRate is the time dilation while in slowMo

-KeysCombination is the order of keys from input struct like
(0) = E
(1) = LMB
(2) = RMB
and etc, everything is equal to Input settings
(doesn't work if random bool is active)

Now animation settings, there is 2 struct one for player(anim settings) and other for opposite(enemy or object) they play synchronously



Anim Settings	
Start_Anim	Choke_Start_Montage
Loop_Anims	1 elements
0	Choke_Idle_Montage
End_Anim	Choke_End_Fine_Montage
Failed_Anim	Choke_End_Bad_Montage



Opposite Anim Settings	
Start_Anim	Choked_Start_Montage
Loop_Anims	1 elements
0	Choked_Idle_Montage
End_Anim	Choked_End_Fine_Montage
Failed_Anim	Choked_End_Bad_Montage

we have start loop’s, end and fail anim for both of them

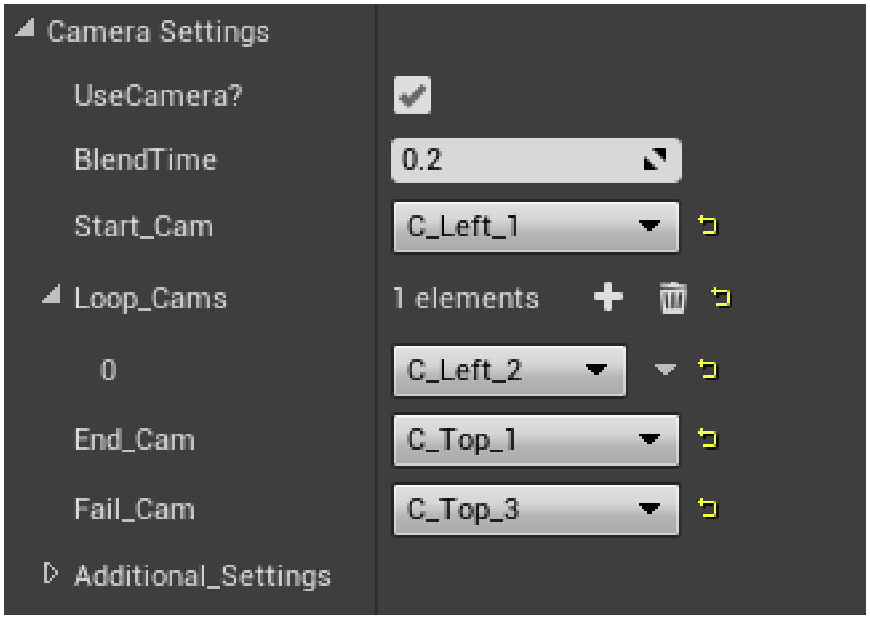
-every anim is montage, start fires first, then it goes to loop
-loop is array because we can progress through them by qte progress,
-so each time we press success button, loop anim progress forward(if exist)

when we finish qte successfully both End anim happens, if we fail it both Failed_anim fires

camera’s work almost the same way like animations

-blend time between cameras, if 0 sharp transition will be

-we have start cam, loop cams, end and fail cam.
-we choose camera from enum value in relation to player(left/right/top) etc

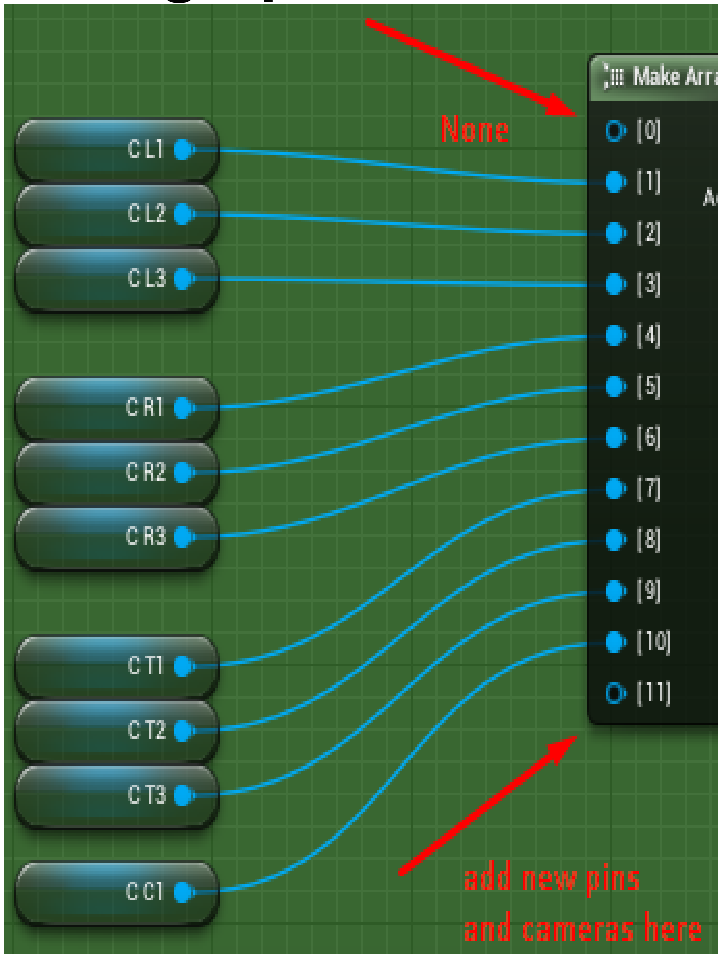


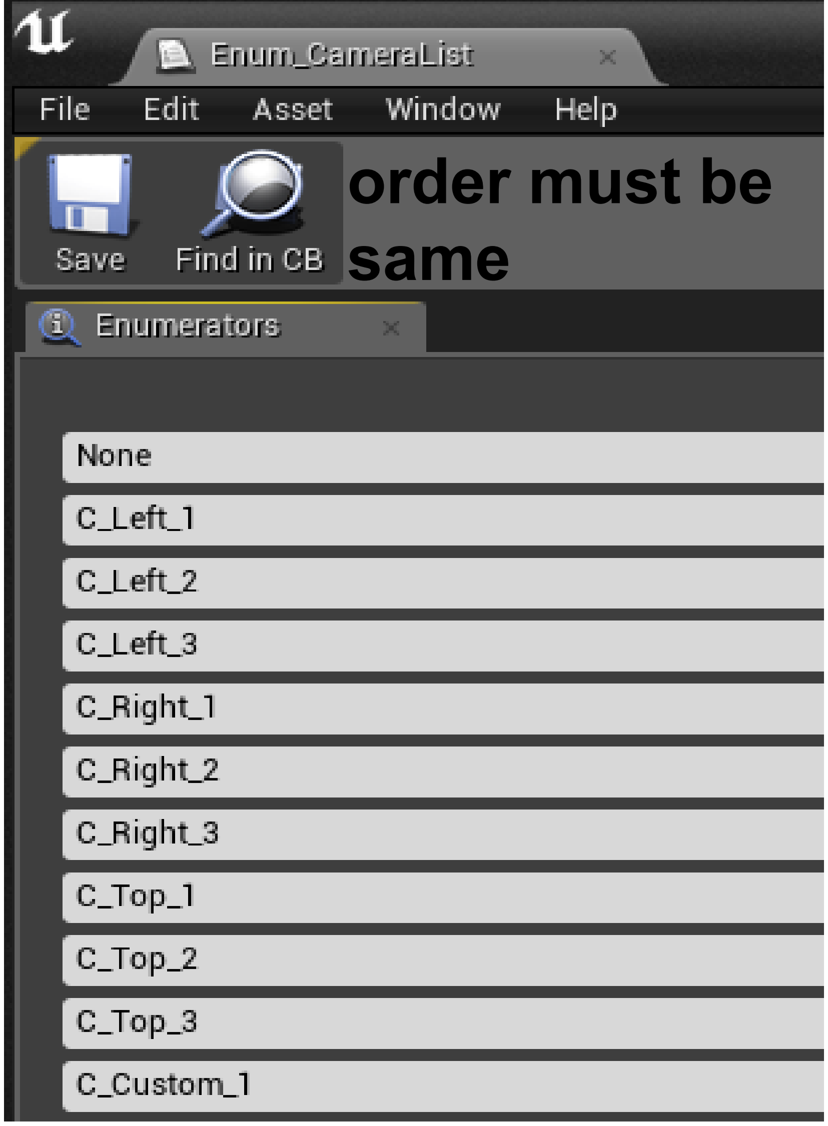
Camera Settings

UseCamera?	✓
BlendTime	0.2
Start_Cam	C_Left_1
Loop_Cams	1 elements
0	C_Left_2
End_Cam	C_Top_1
Fail_Cam	C_Top_3
Additional_Settings	

all those camera’s exist in BP_QTE_Cams

so we add new camera view to Enum_CameraList, and we create that new cam in bp_qte cams, then we put it to array in the same order, to use it in this struct this is in bp_qte_cams event graph

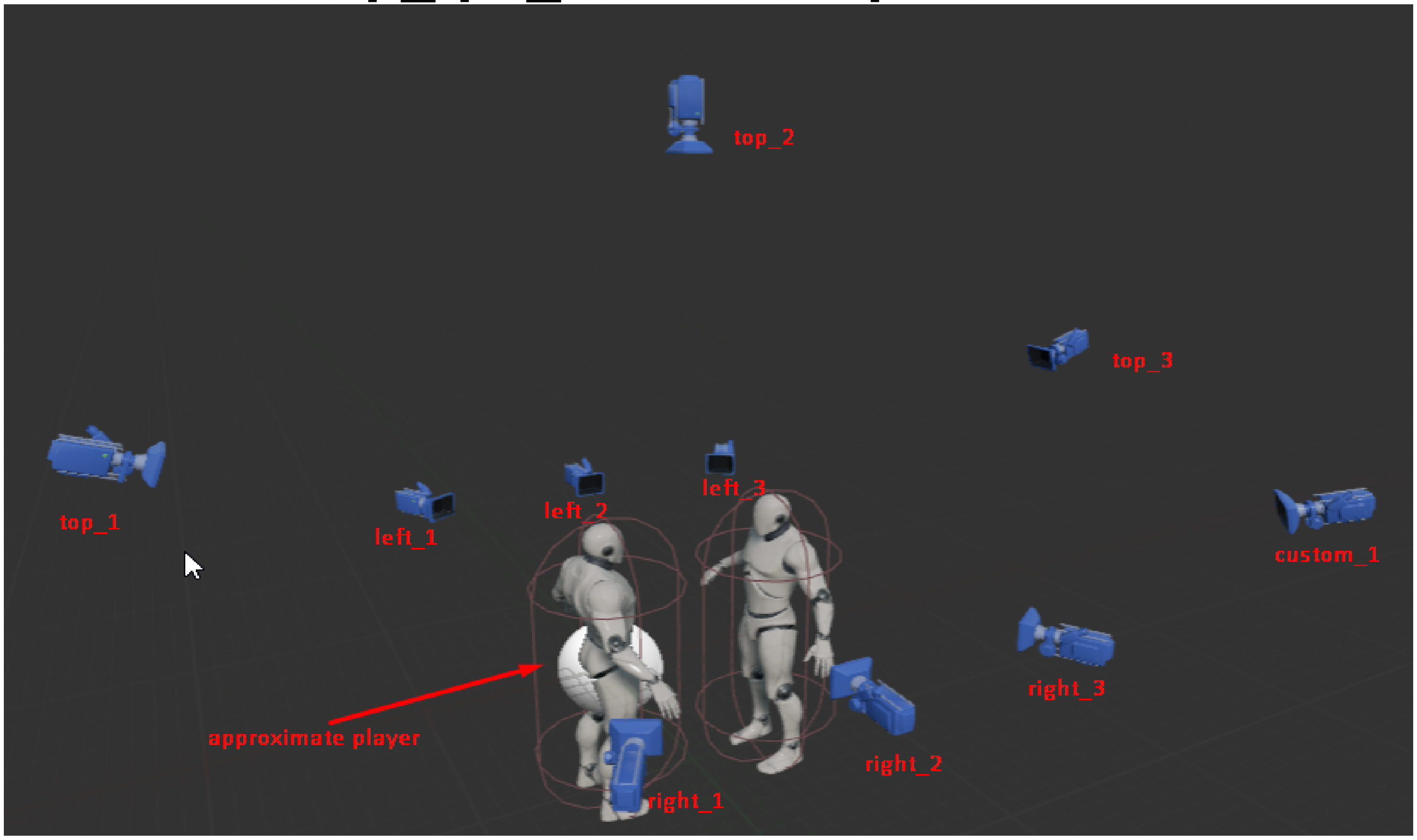




None
C_Left_1
C_Left_2
C_Left_3
C_Right_1
C_Right_2
C_Right_3
C_Top_1
C_Top_2
C_Top_3
C_Custom_1

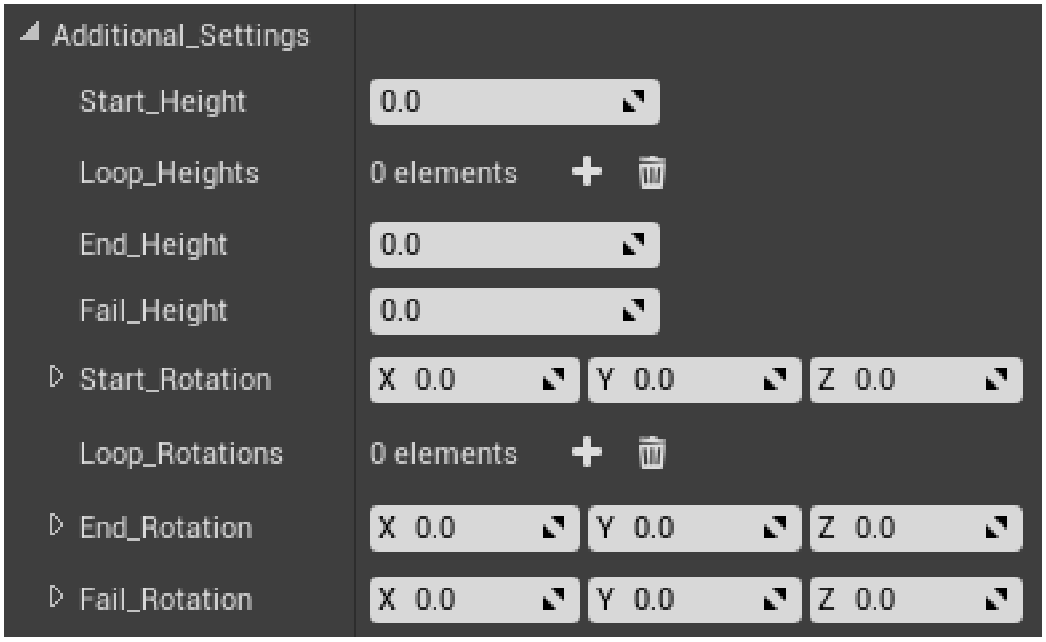
order must be same

and this is bp_qte_cams viewport



bp_qte_cams, spawn and attaches to player
then we switch between cameras by qte progress

also we do have additional camera settings to save time
we can set each camera height here
and also add additional rotation to current



Additional_Settings

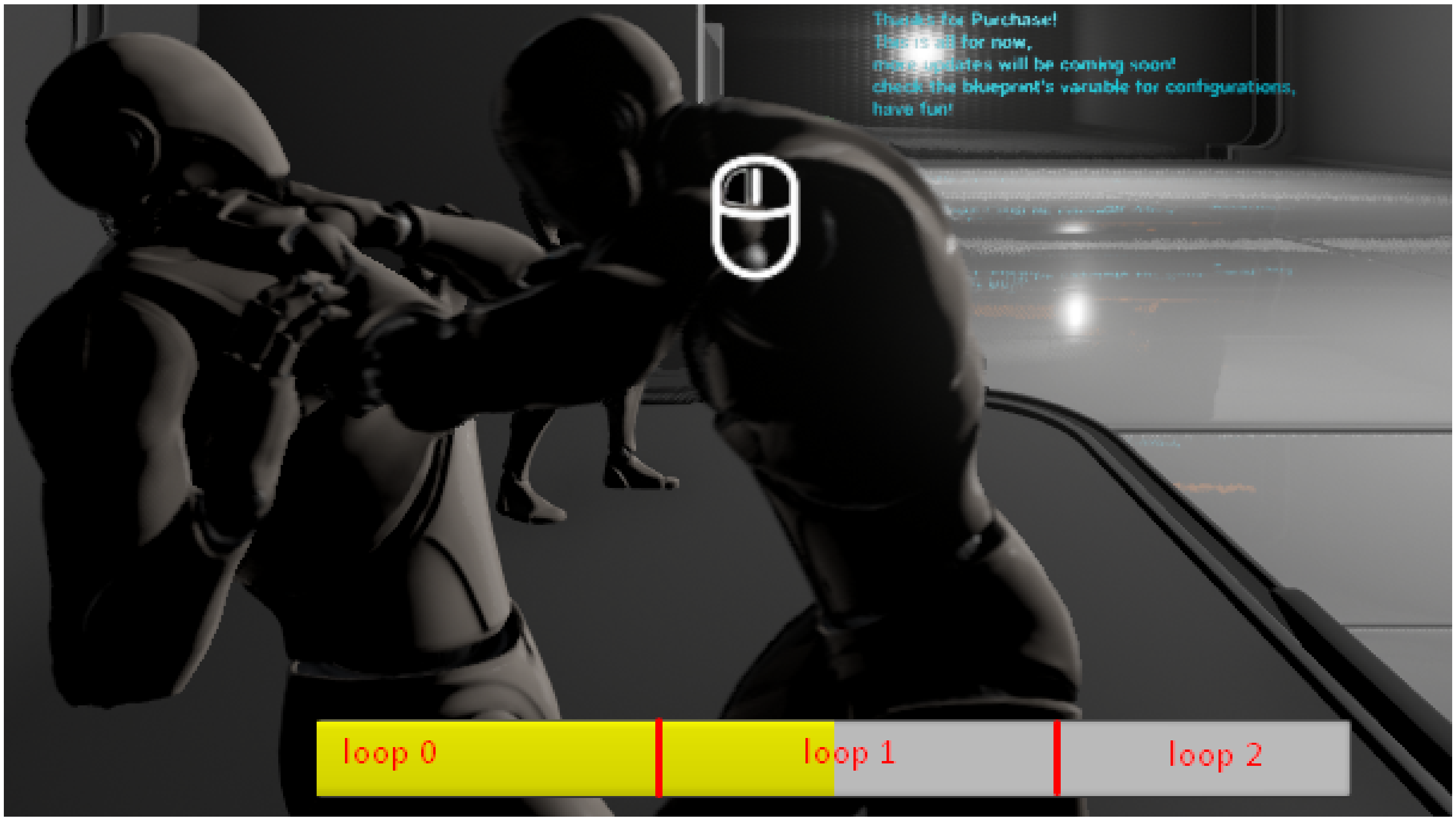
Start_Height	0.0		
Loop_Heights	0 elements		
End_Height	0.0		
Fail_Height	0.0		
Start_Rotation	X 0.0	Y 0.0	Z 0.0
Loop_Rotations	0 elements		
End_Rotation	X 0.0	Y 0.0	Z 0.0
Fail_Rotation	X 0.0	Y 0.0	Z 0.0

so you don’t need to create many additional camera views, to make little changes

so on platform QTE, we set height to -50 so platform doesn’t clip through cameras

and a few words about loop progress

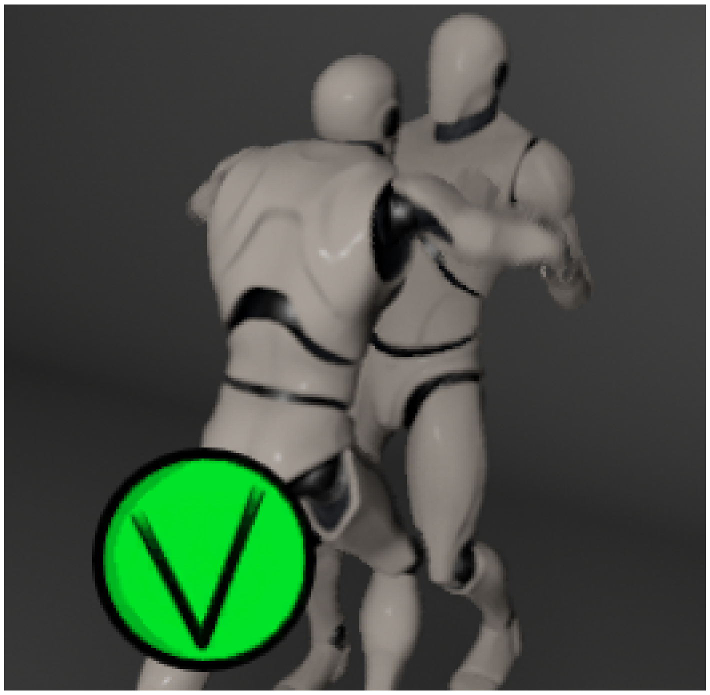
on button mash, qte progress is split to 3 parts


Thanks for Purchase!
This is all for now,
more updates will be coming soon!
check the blueprint's variable for configurations,
have fun!
loop 0
loop 1
loop 2

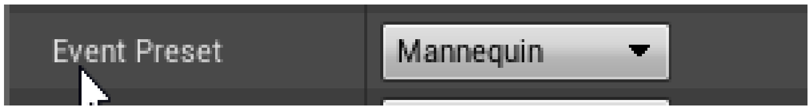
so if we have 3 loop animations or cameras, they will be split to button mash's progress bar
-begin till 33% center till 66% and finish till the end

so loop anim or camera from array will be engage by index, in those parts if they exist at timer QTE, our progress goes forward, each time we press success button

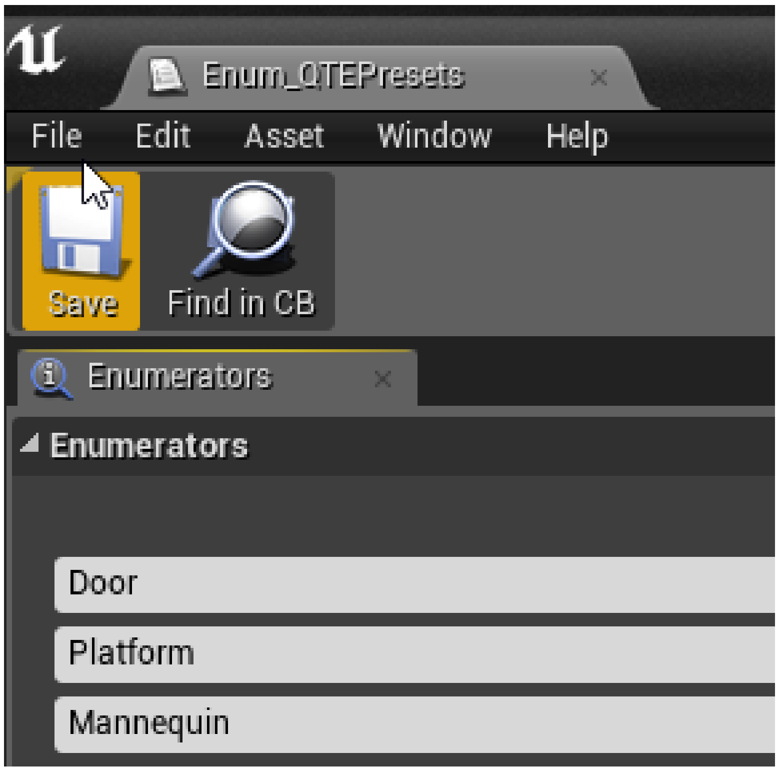
so our loop anims or cams, will progress forward until we reach qte end by qte length



otherwise you can still use one loop anim or one loop camera, anim will not stop or change if there is no more in array and last word about event presets


Event Preset
Mannequin

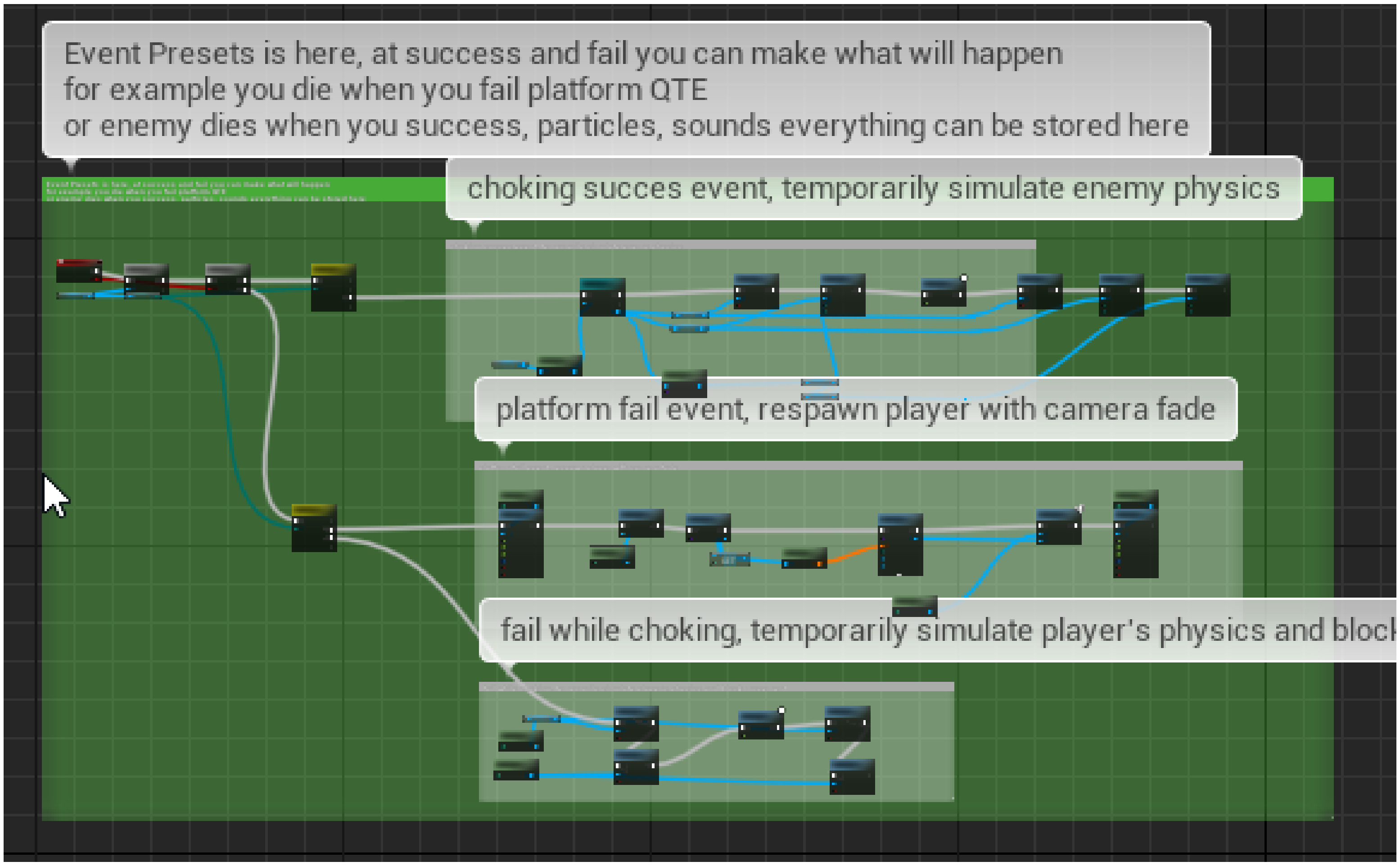
we can add many event presets into Enum_QTEPresets



Door
Platform
Mannequin

and create events what will happen on QTE end fail or success

we can choose preset for each actor that have qte component inside QTE_Base actor's event graph we do have place for Event presets firing as in this image


Event Presets is here, at success and fail you can make what will happen
for example you die when you fail platform QTE
or enemy dies when you success, particles, sounds everything can be stored here

choking succes event, temporarily simulate enemy physics

platform fail event, respawn player with camera fade

fail while choking, temporarily simulate player's physics and block

we can add events such like, enemy dies when we successfully choke him, or we are dying when we fail platform event, and etc

spawn particles, sounds and more

and again, all your levels must have at least one QTE_Base actor, don't spawn more than 1 to prevent bugs

Thanks everyone that’s all, if you have any problems or questions please contact me.

godofwar8080@gmail.com