Dynamic Combat System

Contact email: szewczykgrzesiek@gmail.com

Dynamic Combat System

Player

- InputBuffer
- Guard Stance
- Ignore Root Motion
- Montage Actions used by player
  - Important Montages
  - Additional Montages

AI

- Creating new AI
- Animations used by AI
  - Important Montages
  - Additional Montages
- Stun System
- Block System
- Patrol
- Rotate Towards Target
- Dissolve Effect
- Parry

Interfaces

- IsAttackable
- CanBeImmortal
- IsTargetable
- IsSusceptibleToParry

Components

- CollisionHandlerComponent
- DynamicTargetingComponent
- InputBufferComponent
- MontageManagerComponent
- StateMachineComponent
- StatisticComponent

Animation Notifies

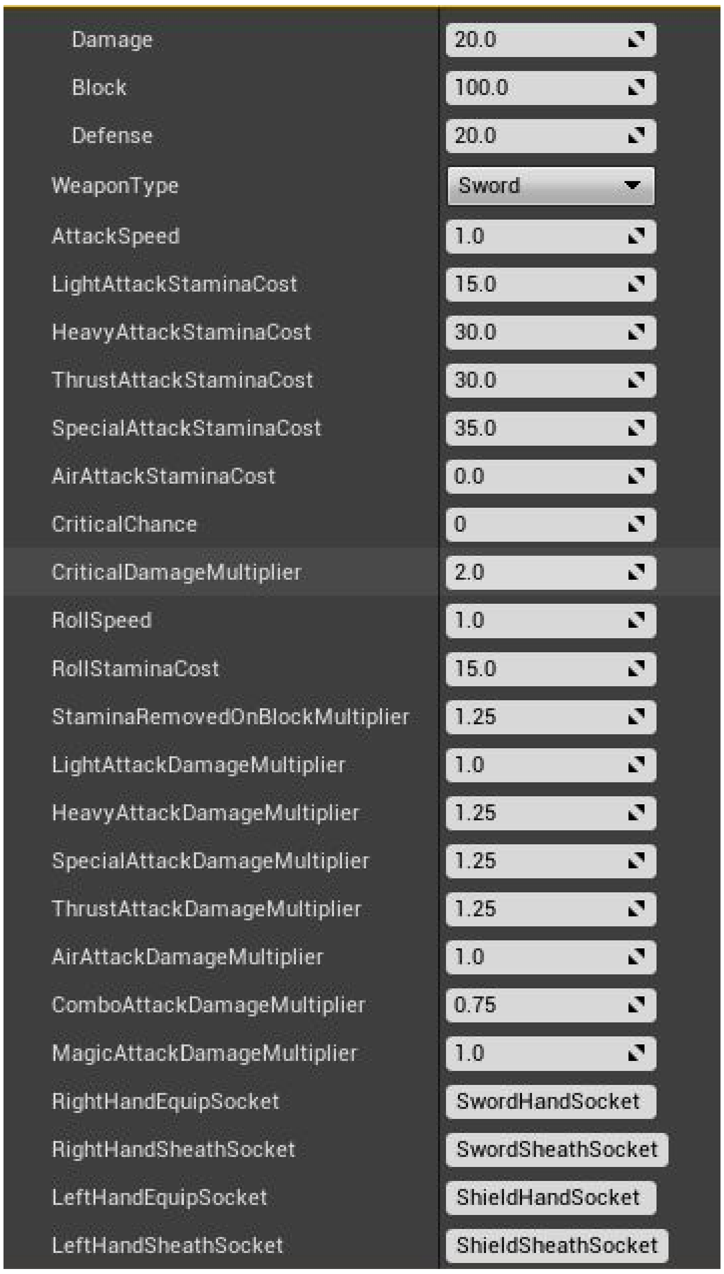
- AN_IgnoreRootMotion-Player

- ANS_IgnoreRootMotion-Player
- AN_ResetCombat-Player
- ANS_ParrySusceptible
- ANS_HitBox
- ANS_InputBuffer-Player
- ANS_RotateTowardsTarget-AI
- ANS_ToggleImmortality
- ANS_ToggleWeaponAttachment-Player

Enums
- EAICombatBehavior
- EAIHealthState
- EAIPatrolBehavior
- EAIStatus
- EAttackType
- EInputBufferKey
- EMontageAction
- ERotationMode
- EState

Player

All combat stats which can be modifiable are stored in one struct.



Damage	20.0
Block	100.0
Defense	20.0
WeaponType	Sword
AttackSpeed	1.0
LightAttackStaminaCost	15.0
HeavyAttackStaminaCost	30.0
ThrustAttackStaminaCost	30.0
SpecialAttackStaminaCost	35.0
AirAttackStaminaCost	0.0
CriticalChance	0
CriticalDamageMultiplier	2.0
RollSpeed	1.0
RollStaminaCost	15.0
StaminaRemovedOnBlockMultiplier	1.25
LightAttackDamageMultiplier	1.0
HeavyAttackDamageMultiplier	1.25
SpecialAttackDamageMultiplier	1.25
ThrustAttackDamageMultiplier	1.25
AirAttackDamageMultiplier	1.0
ComboAttackDamageMultiplier	0.75
MagicAttackDamageMultiplier	1.0
RightHandEquipSocket	SwordHandSocket
RightHandSheathSocket	SwordSheathSocket
LeftHandEquipSocket	ShieldHandSocket
LeftHandSheathSocket	ShieldSheathSocket

InputBuffer

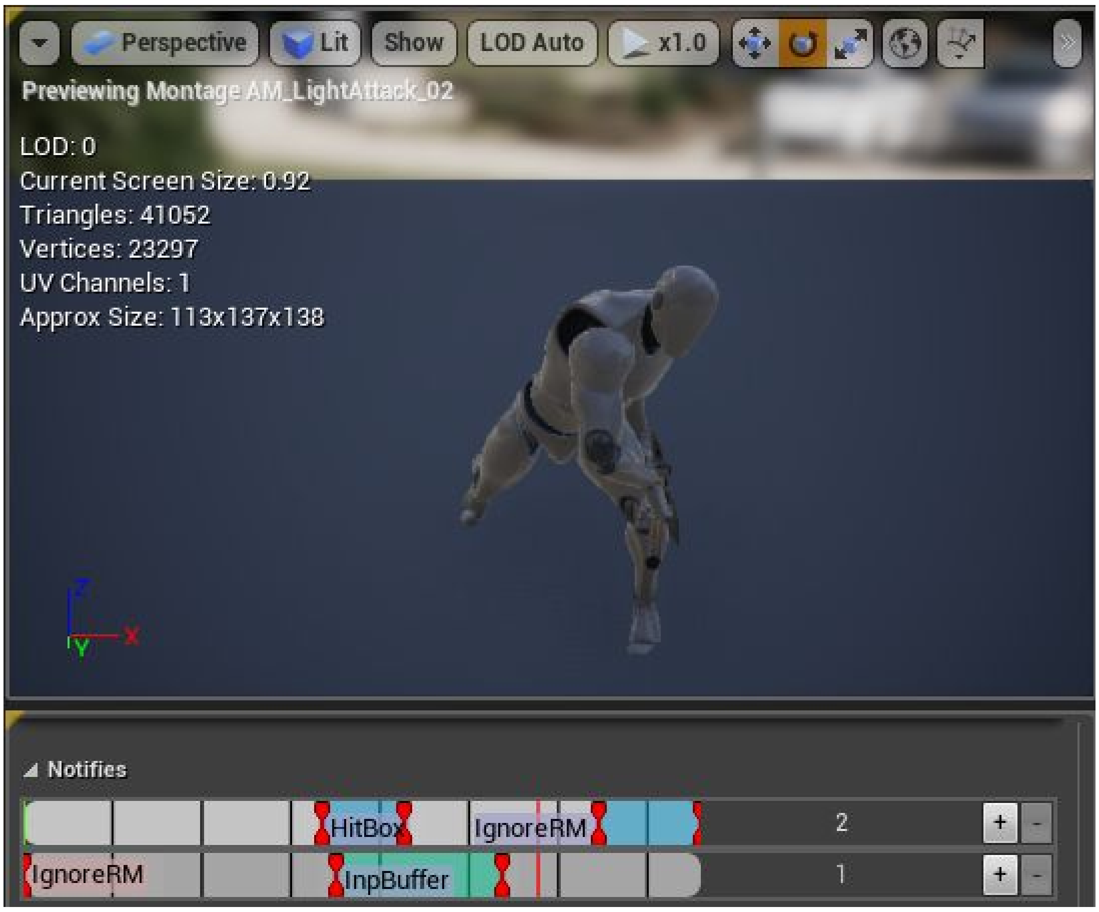
Player is using InputBufferComponent.

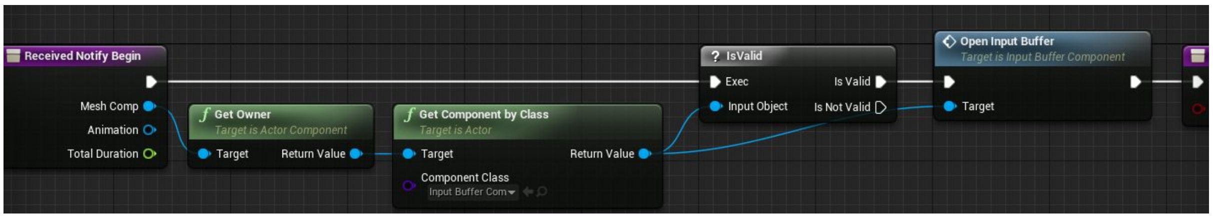
This component allows to store key(EInputBufferKey) in buffer and perform action based on that key when buffer is closed.

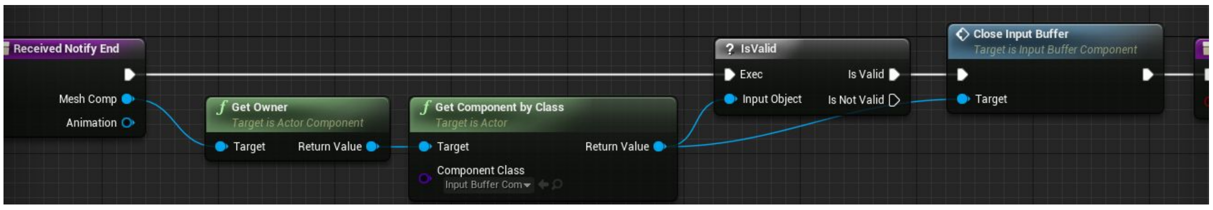
e.g. when input buffer is open, and character is attacking, you can press roll input, then roll key will be stored in buffer, and when buffer closes,

checks if there is any key inside, in this case there would be roll key and Roll function will be called.
It gives feeling of better control over the character.

To open and close InputBuffer, use ANS_InputBuffer-Player.
In this example InputBuffer is open from 50-70% of the montage.

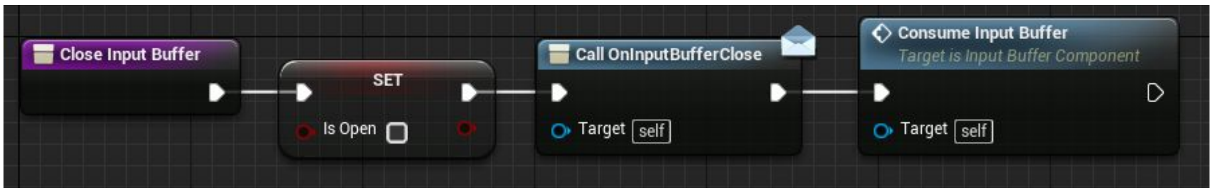

Previewing Montage AM_LightAttack_02
LOD: 0
Current Screen Size: 0.92
Triangles: 41052
Vertices: 23297
UV Channels: 1
Approx Size: 113x137x138

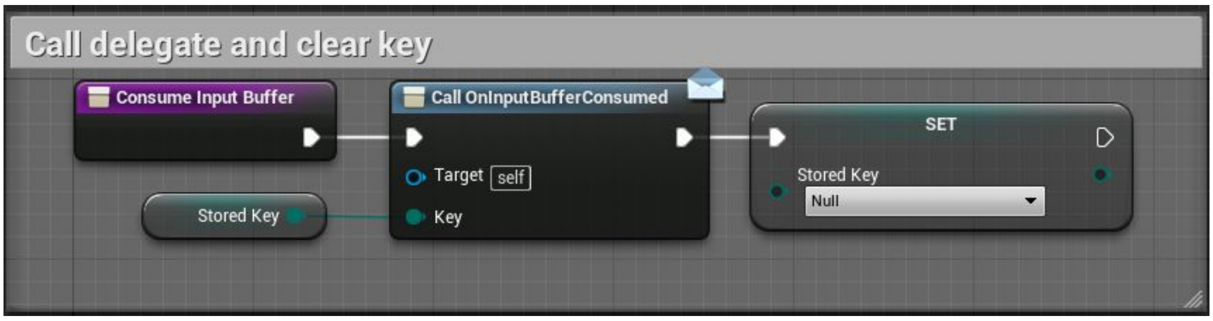


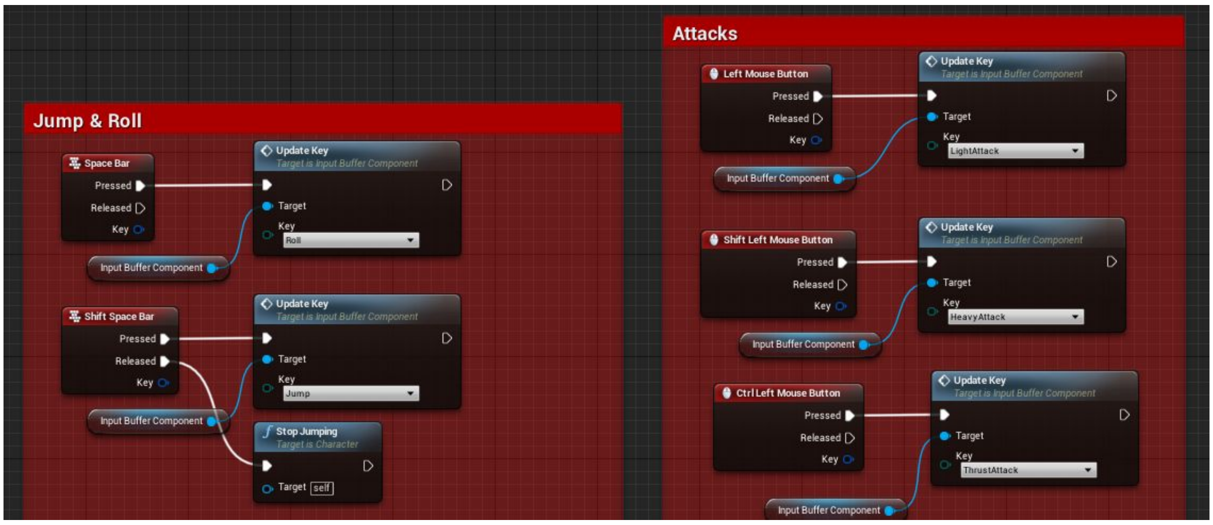


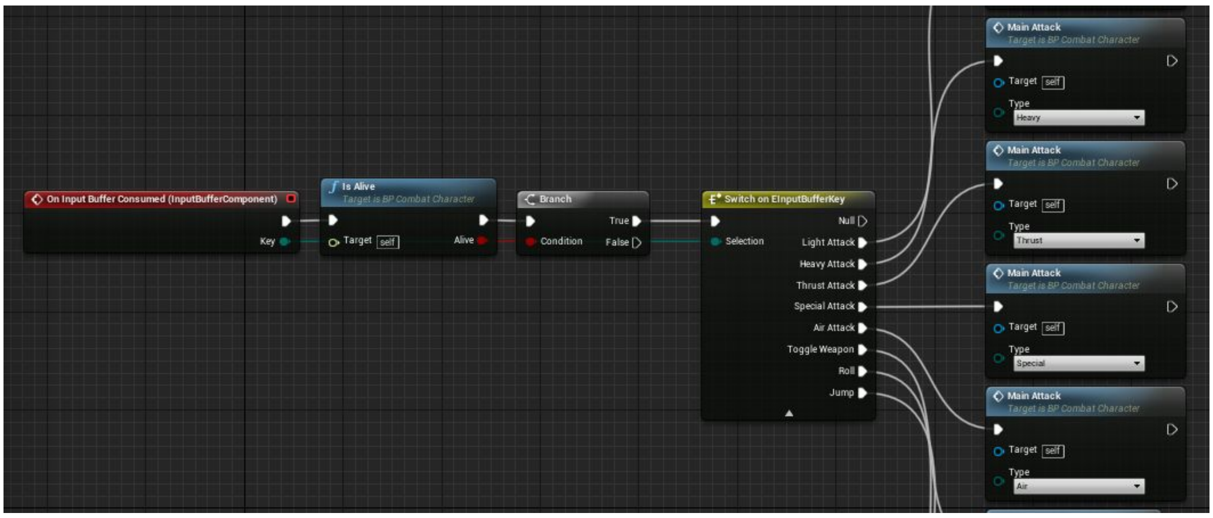
So actions like attack/roll/jump etc. are not calling their functions directly.

They update key stored in InputBuffer, and when it closes, key is consumed, calling the proper function.

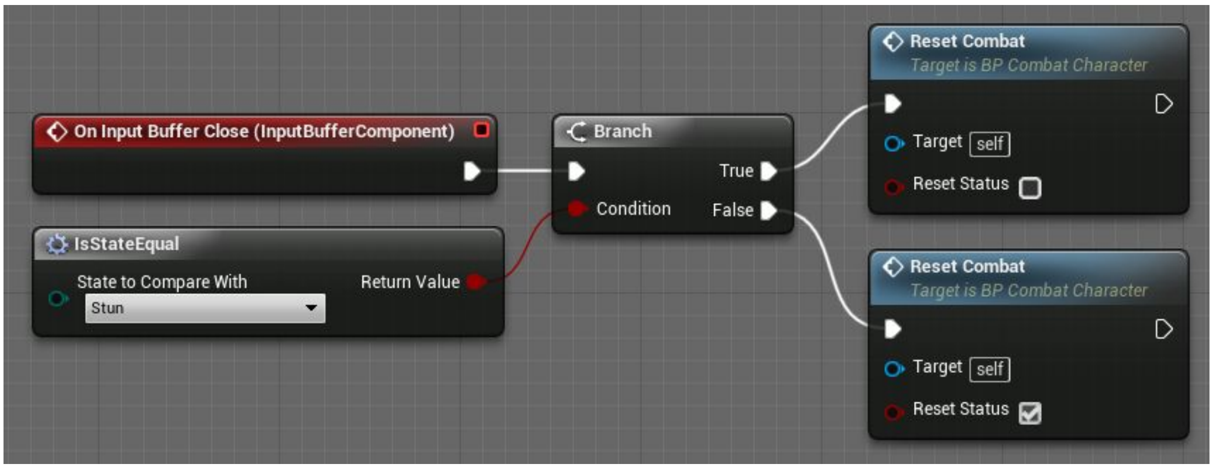


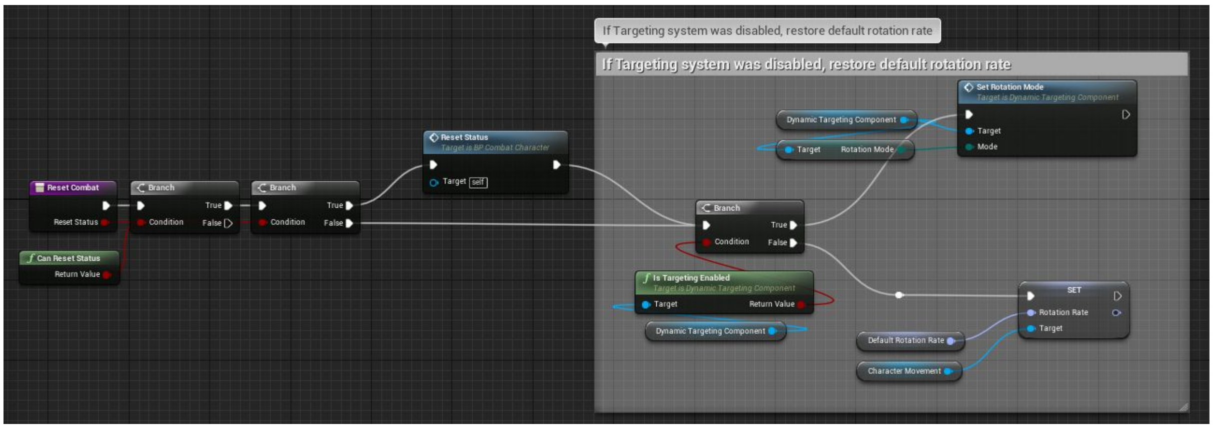


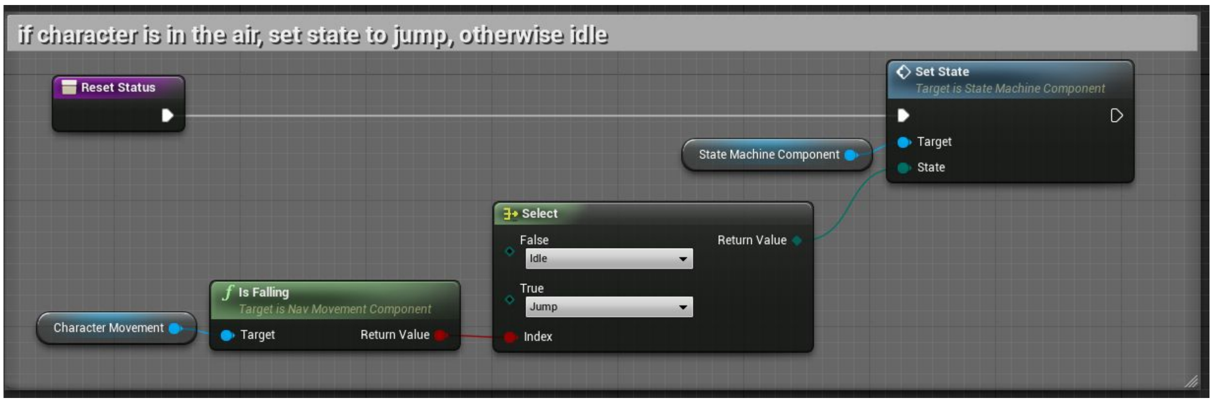




Additionaly when InputBuffer closes, ResetCombat function is called.







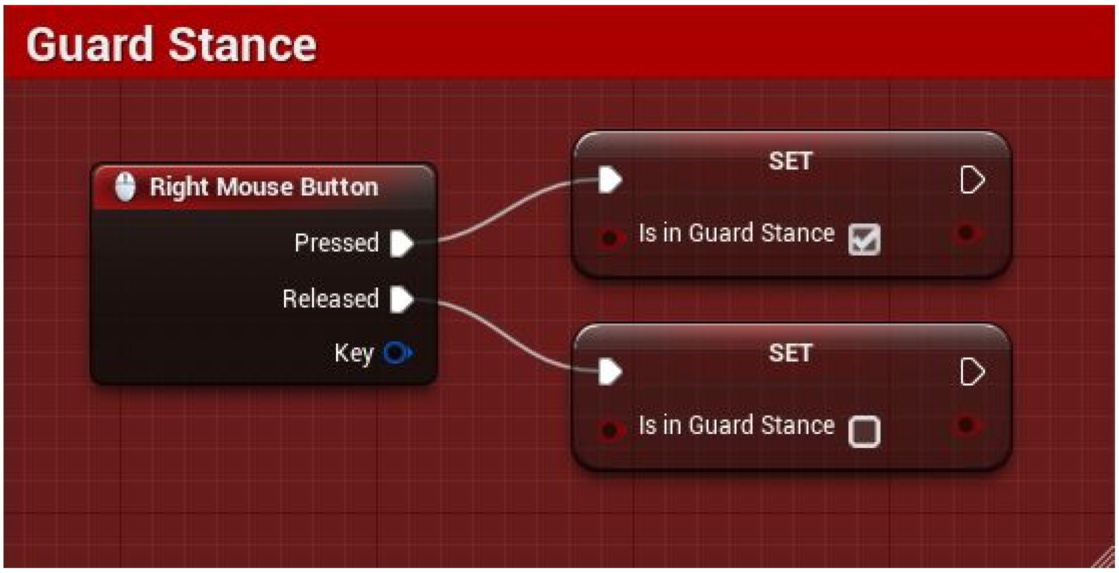
Notice that if character was in Stun state, status won't be resetted. This is problem with anim notify states, when it is interrupted Notify End Event will be called on next frame.

Let's say that character was attacking and Input buffer was open, but then got hit and state changed to stun, also playing stun animation. On the next frame InputBuffer end event would be called, calling function ResetCombat which would reset state back to Idle. It means that stun state would last only for 1 frame.

That’s why on montages like Stun/Impact you shouldn’t use ANS_InputBuffer-Player, but AN_ResetCombat-Player instead.

Guard Stance

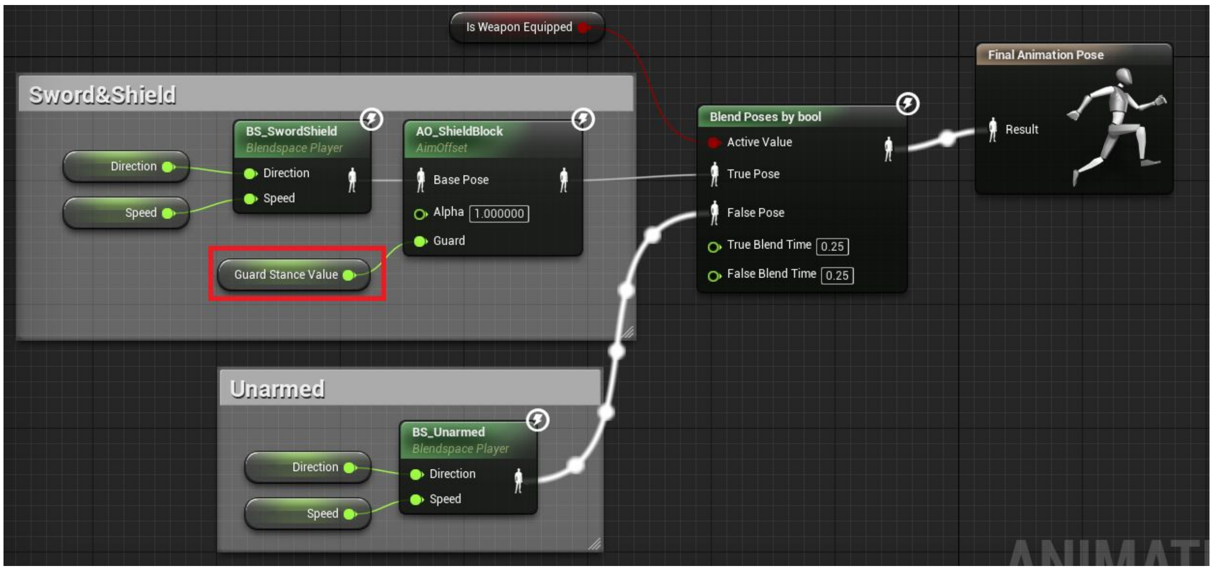
Enabled and disabled by pressing input key.



Based on that bool value, variable GuardStanceValue is smoothly changed between 0 and 1.

When character gets hit while GuardStanceValue equals 1, it means that hit was blocked.
If there is enough stamina, it will be removed instead of health and stun effect won’t be applied.

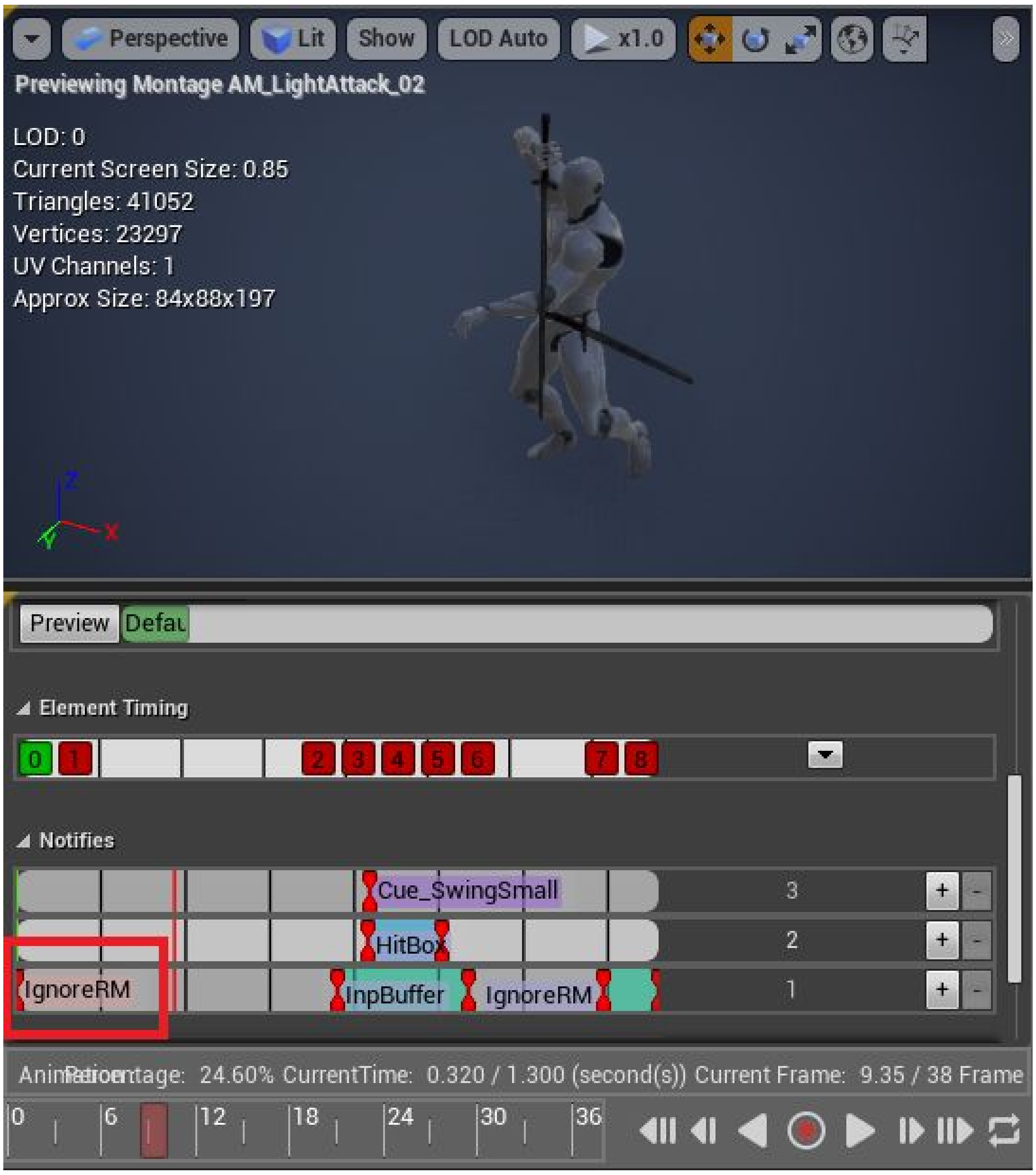
This value is also used in animation blueprint to rise/lower shield



Ignore Root Motion

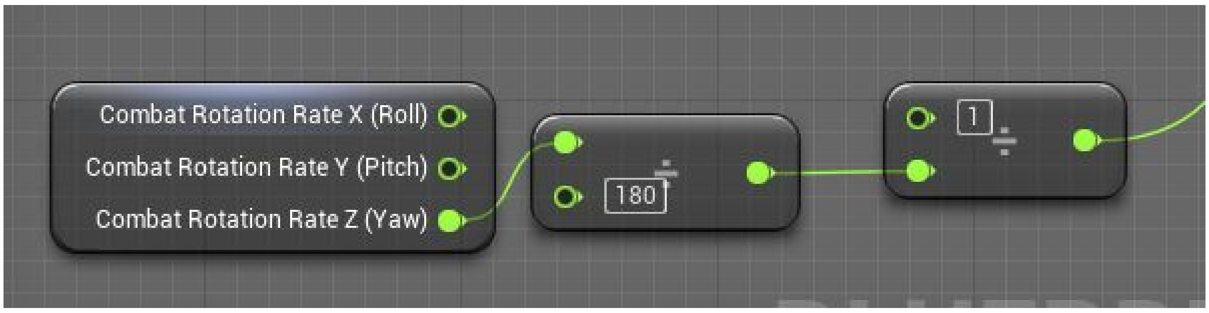
System which allows character to turn into desired direction at the beginning of the montage like attack/roll etc.

Firstly AN_IgnoreRootMotion-Player placed at the begging of the montage is calling IgnoreRootMotion function from BP_CombatCharacter.

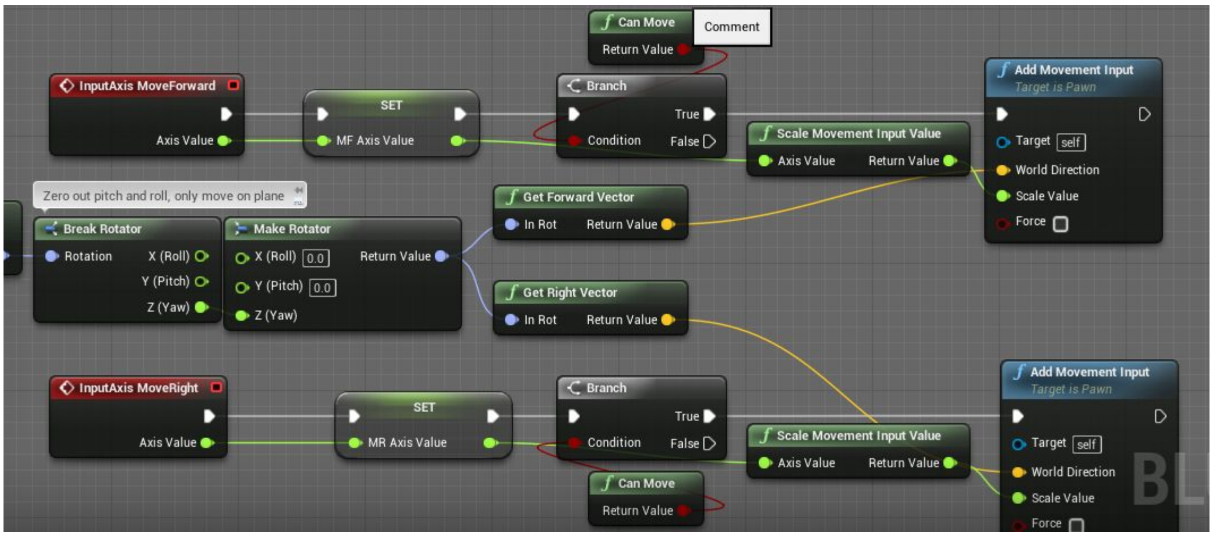


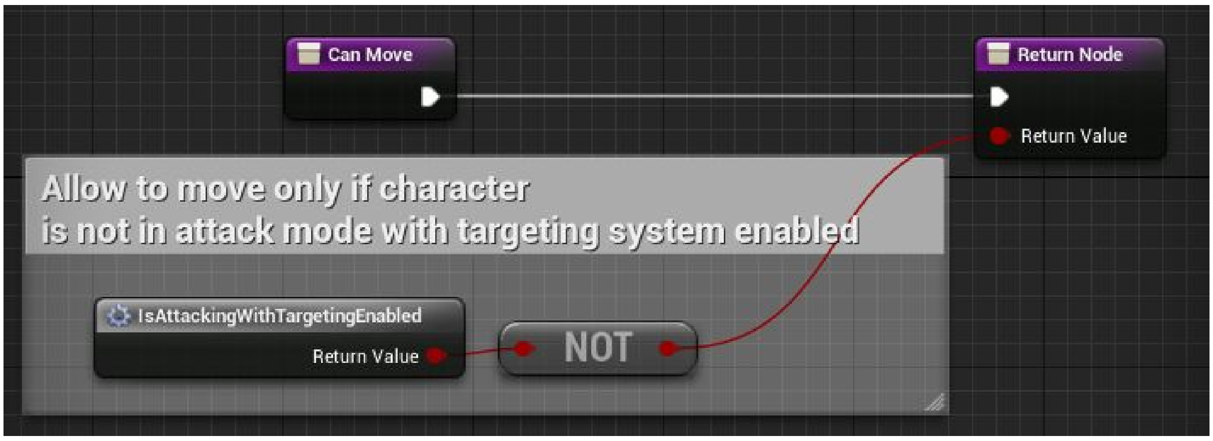
Function is calculating for how long root motion should be ignored based on variable CombatRotationRate (Rotator used as RotationRate in MovementComponent when character enters Roll/Attack state, by default it is 0,0,900)

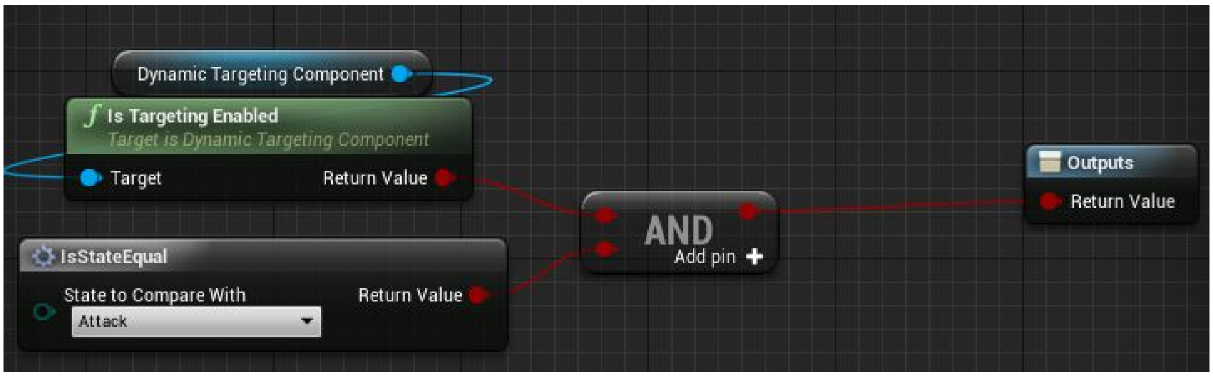
With 900 rotation rate yaw, because we only care about that, it takes 0.2sec to turn 180 degrees(max)
[1/(900/180)].


Combat Rotation Rate X (Roll)
Combat Rotation Rate Y (Pitch)
Combat Rotation Rate Z (Yaw)
180
÷
1
÷

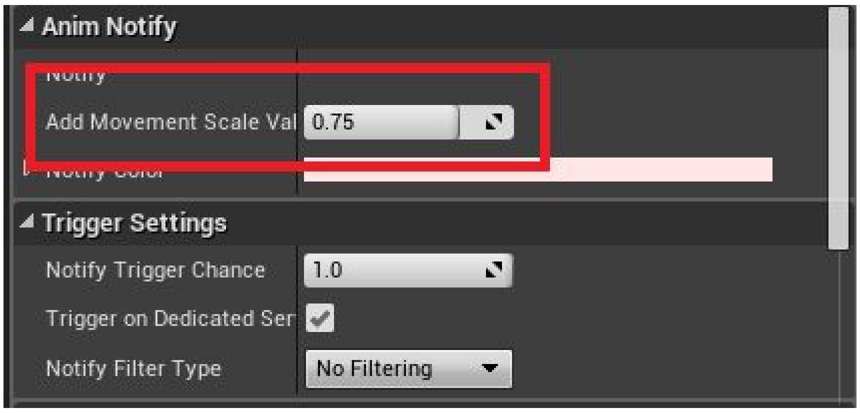
During that time, if targeting system is enabled and player is in attack state, character can't be moved using axis inputs, to prevent moving right/left/back.




Can Move
Return Node
Return Value
Allow to move only if character
is not in attack mode with targeting system enabled
IsAttackingWithTargetingEnabled
Return Value
NOT



Instead character will be moved by ticking function using AddMovementScaleValue provided by AN_IgnoreRootMotion-Player.



Anim Notify

Add Movement Scale Val 0.75

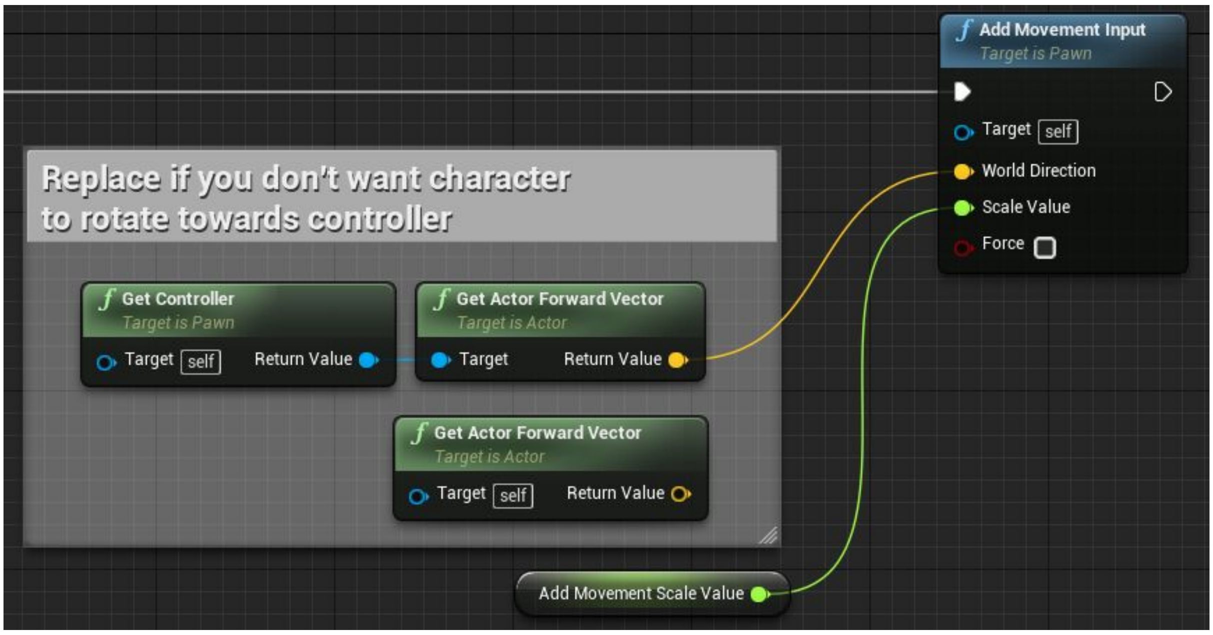
Trigger Settings

Notify Trigger Chance 1.0

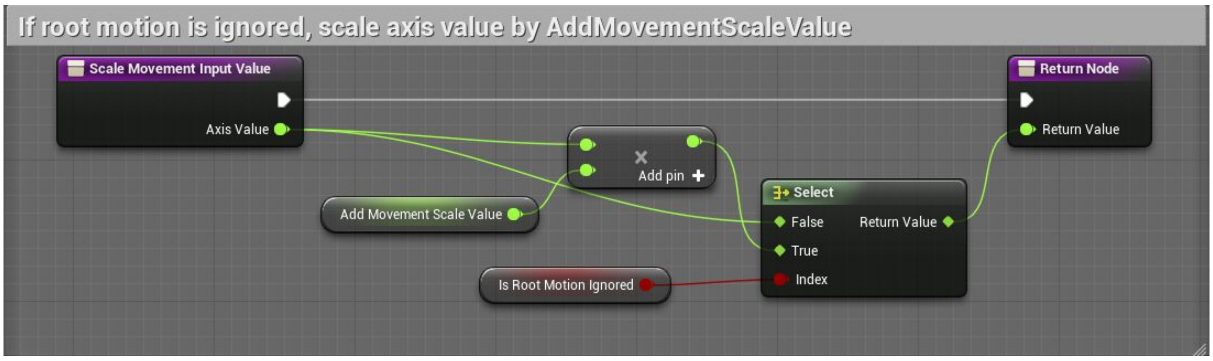
Trigger on Dedicated Ser ✓

Notify Filter Type No Filtering

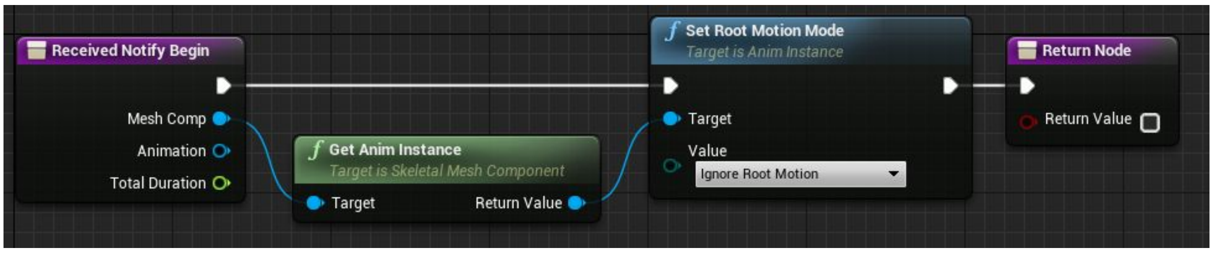
1 means move character(useful if you want character to move while root motion is ignored e.g in roll)
0 don't move at all (useful if you have InPlace attack animations)

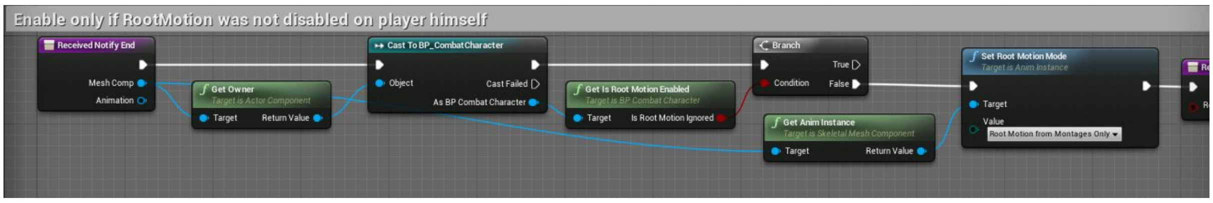


Otherwise if player is not in attack state, axis value will be scaled by AddMovementScaleValue, it means that if value on notify was 0, we won't be able to move character anyway.

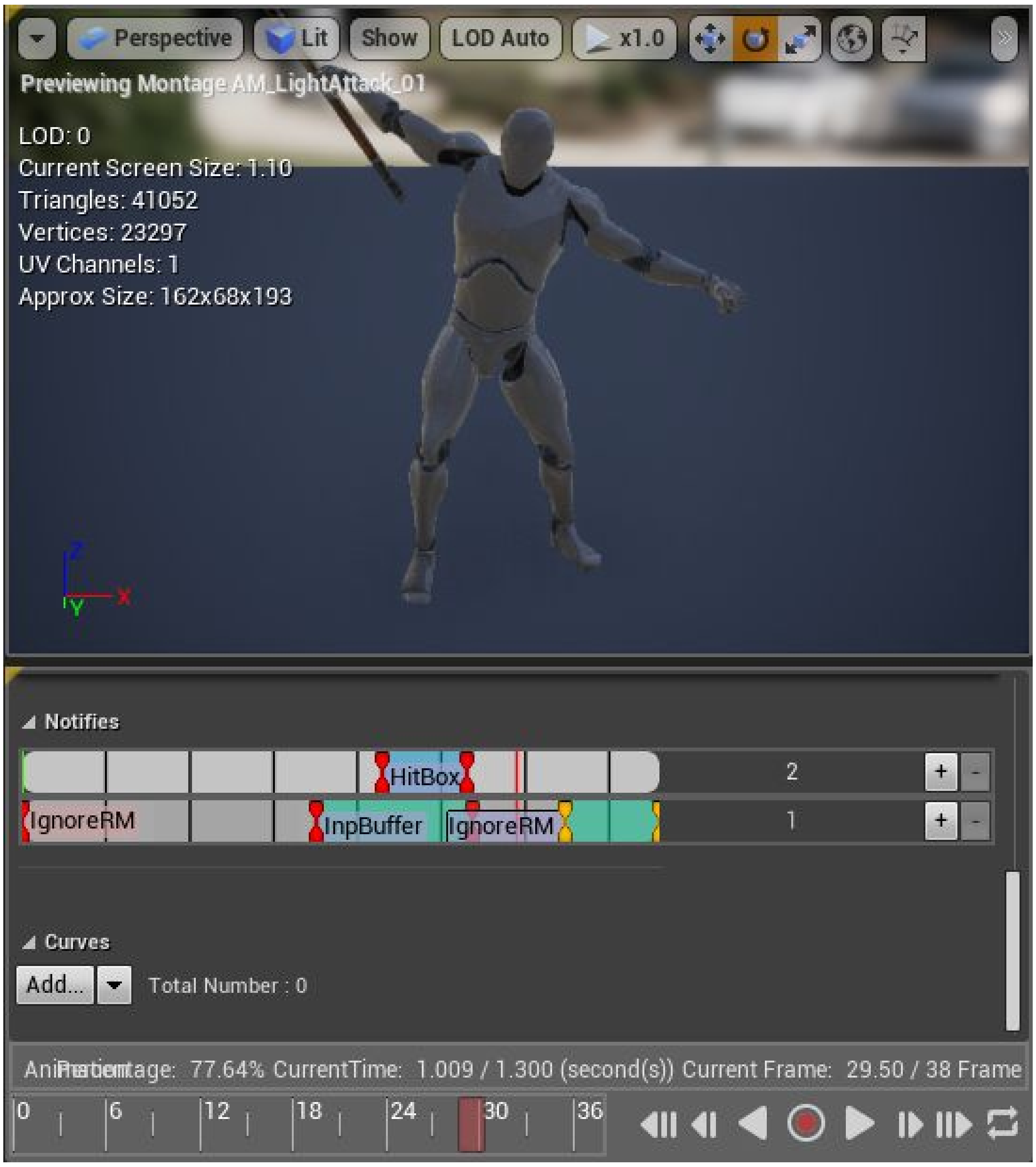


There is also another notify(state) which in contrast to AN_IgnoreRootMotion-Player, is not calling player’s blueprint function, but just ignores root motion on begin event, and enable it back on end event, if root motion was not already disabled by player himself.





Using this notify is optional, it is most useful at the end (last 10-15%) of montages using root motion, like attack, roll to give player faster control over the character while montage is blending out.



Previewing Montage AM_LightAttack_01

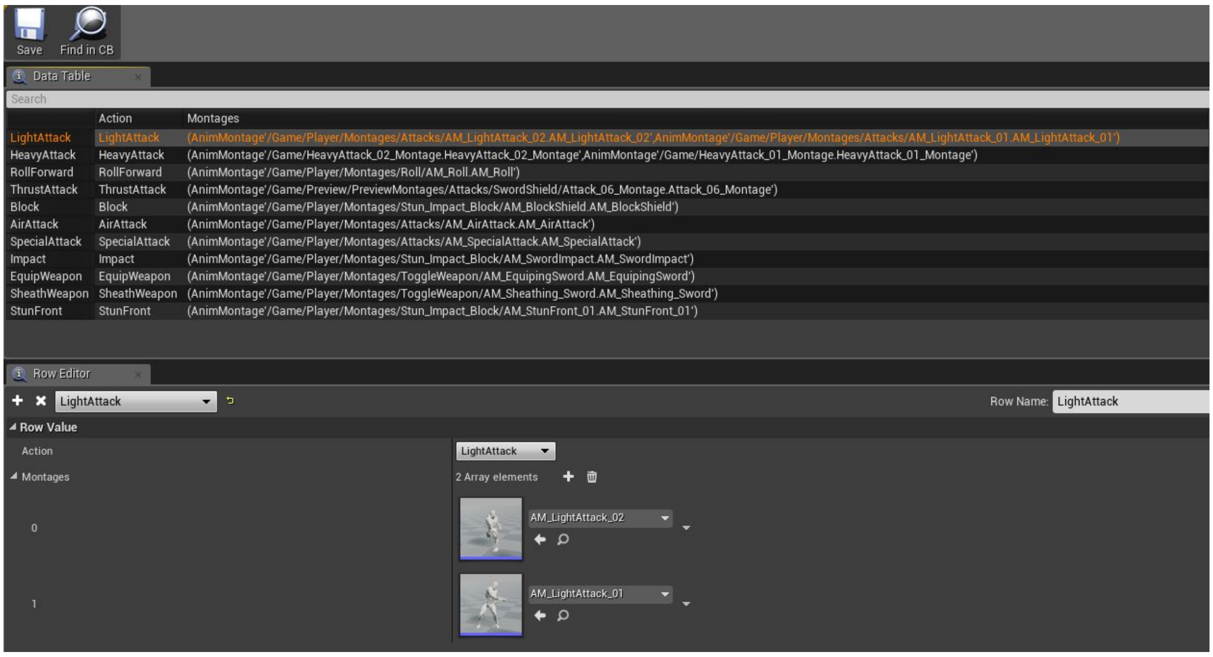
LOD: 0
Current Screen Size: 1.10
Triangles: 41052
Vertices: 23297
UV Channels: 1
Approx Size: 162x68x193

Total Number : 0

Percentage: 77.64% CurrentTime: 1.009 / 1.300 (second(s)) Current Frame: 29.50 / 38 Frame

Montage Actions used by player

All montages used by player are stored in DataTable (DataTables/DT_CombatCharacter), using MontageManagerComponent.
I highly recommend to keep that convection if you are going to add new montages.



	Action	Montages
LightAttack	LightAttack	(AnimMontage'/Game/Player/Montages/Attacks/AM_LightAttack_02.AM_LightAttack_02',AnimMontage'/Game/Player/Montages/Attacks/AM_LightAttack_01.AM_LightAttack_01')
HeavyAttack	HeavyAttack	(AnimMontage'/Game/HeavyAttack_02_Montage.HeavyAttack_02_Montage',AnimMontage'/Game/HeavyAttack_01_Montage.HeavyAttack_01_Montage')
RollForward	RollForward	(AnimMontage'/Game/Player/Montages/Roll/AM_Roll.AM_Roll')
ThrustAttack	ThrustAttack	(AnimMontage'/Game/Preview/PreviewMontages/Attacks/SwordShield/Attack_06_Montage.Attack_06_Montage')
Block	Block	(AnimMontage'/Game/Player/Montages/Stun_Impact_Block/AM_BlockShield.AM_BlockShield')
AirAttack	AirAttack	(AnimMontage'/Game/Player/Montages/Attacks/AM_AirAttack.AM_AirAttack')
SpecialAttack	SpecialAttack	(AnimMontage'/Game/Player/Montages/Attacks/AM_SpecialAttack.AM_SpecialAttack')
Impact	Impact	(AnimMontage'/Game/Player/Montages/Stun_Impact_Block/AM_SwordImpact.AM_SwordImpact')
EquipWeapon	EquipWeapon	(AnimMontage'/Game/Player/Montages/ToggleWeapon/AM_EquipingSword.AM_EquipingSword')
SheathWeapon	SheathWeapon	(AnimMontage'/Game/Player/Montages/ToggleWeapon/AM_Sheathing_Sword.AM_Sheathing_Sword')
StunFront	StunFront	(AnimMontage'/Game/Player/Montages/Stun_Impact_Block/AM_StunFront_01.AM_StunFront_01')

Row Name: LightAttack

Row Value
Action: LightAttack
Montages: 2 Array elements
0: AM_LightAttack_02
1: AM_LightAttack_01

Take note that RowName must be the same as MotageAction value.

Important Montages

If any of those montages will not be valid, action will simply not be performed.

Light Attack (Slot: FullBody, with root motion)
Heavy Attack (Slot: FullBody, with root motion)
Thrust Attack (Slot: FullBody, with root motion)
Special Attack (Slot: FullBody, with root motion)
Air Attack (Slot: UpperBody, no root motion)
Impact (Slot: FullBody, with root motion)
Stun Front(Slot: FullBody, with root motion)
Block (Slot: UpperBody, no root motion)
Equip Weapon (Slot: UpperBody, no root motion)
Sheath Weapon (Slot: UpperBody, no root motion)
Roll Forward (Slot: FullBody, with root motion)

Additional Montages

For directional stun and roll montages there is already implemented logic inside player character, just set animations with proper enum value, Row Name and it will work.

Stun Back (Slot: FullBody, with root motion)
Stun Left (Slot: FullBody, with root motion)
Stun Right (Slot: FullBody, with root motion)

Roll Forward Left (Slot: FullBody, with root motion)
Roll Forward Right (Slot: FullBody, with root motion)
Roll Left (Slot: FullBody, with root motion)
Roll Right (Slot: FullBody, with root motion)
Roll Back (Slot: FullBody, with root motion)
Roll Back Left (Slot: FullBody, with root motion)
Roll Back Right (Slot: FullBody, with root motion)

AI

Remember to enable EQS.
https://docs.unrealengine.com/latest/INT/Engine/AI/EnvironmentQuerySystem/QuickStart/2

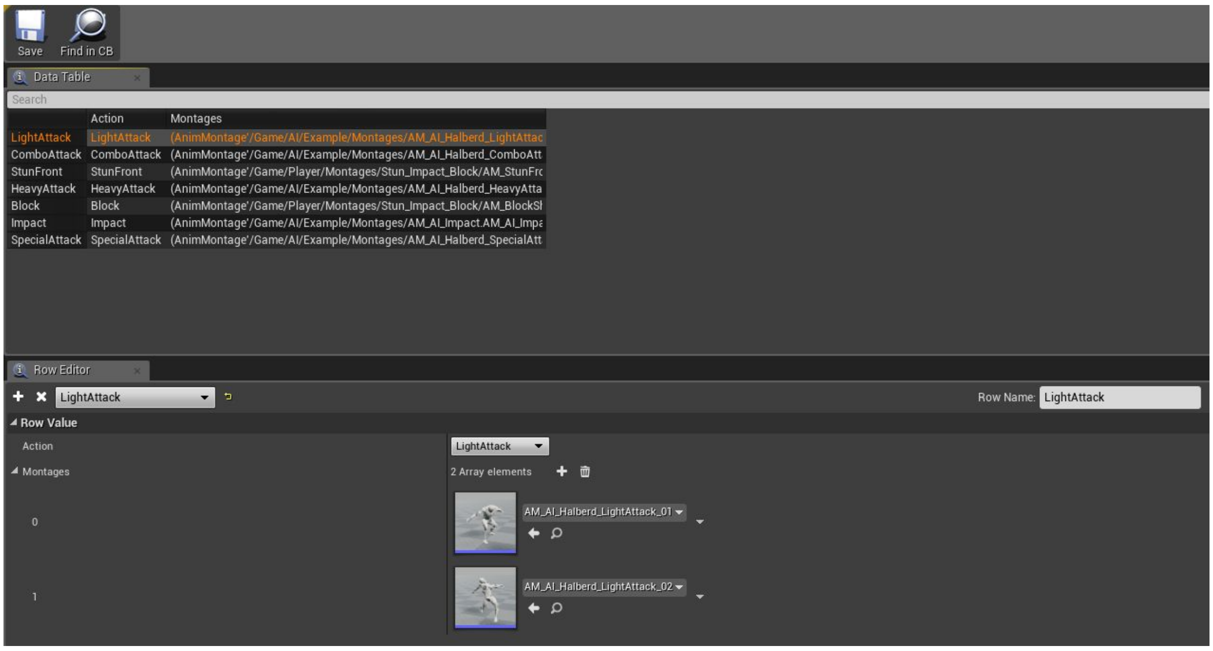
Creating new AI

1. Create new blueprint based on BP_BaseAI
2. Create new BehaviorTree with BB_Base as blackboard or use BT_ExampleAI
3. In AI defaults set BehaviorTree pointer to created one
4. By default AI will have BP_BaseAIController as AIController, it will set player as target value in blackboard on detection

5. Set AI mesh, anim instance, play around with default settings
6. Set Montages DataTable in MontagesManagerComponent to DT_ExampleAI or create your own with different animations
7. After those steps AI should be ready, and you can start building Behavior Tree using included tasks/services/decorators if you didn't use BT_ExampleAI

Animations used by AI

Same as player, all montages used by AI are stored in DataTable (DataTables/DT_ExampleAI), using MontageManagerComponent.
I highly recommend to keep that convection if you are going to add new montages.



	Action	Montages
LightAttack	LightAttack	(AnimMontage'/Game/AI/Example/Montages/AM_AI_Halberd_LightAttac
ComboAttack	ComboAttack	(AnimMontage'/Game/AI/Example/Montages/AM_AI_Halberd_ComboAtt
StunFront	StunFront	(AnimMontage'/Game/Player/Montages/Stun_Impact_Block/AM_StunFrc
HeavyAttack	HeavyAttack	(AnimMontage'/Game/AI/Example/Montages/AM_AI_Halberd_HeavyAtta
Block	Block	(AnimMontage'/Game/Player/Montages/Stun_Impact_Block/AM_BlockSl
Impact	Impact	(AnimMontage'/Game/AI/Example/Montages/AM_AI_Impact.AM_AI_Impa
SpecialAttack	SpecialAttack	(AnimMontage'/Game/AI/Example/Montages/AM_AI_Halberd_SpecialAtt

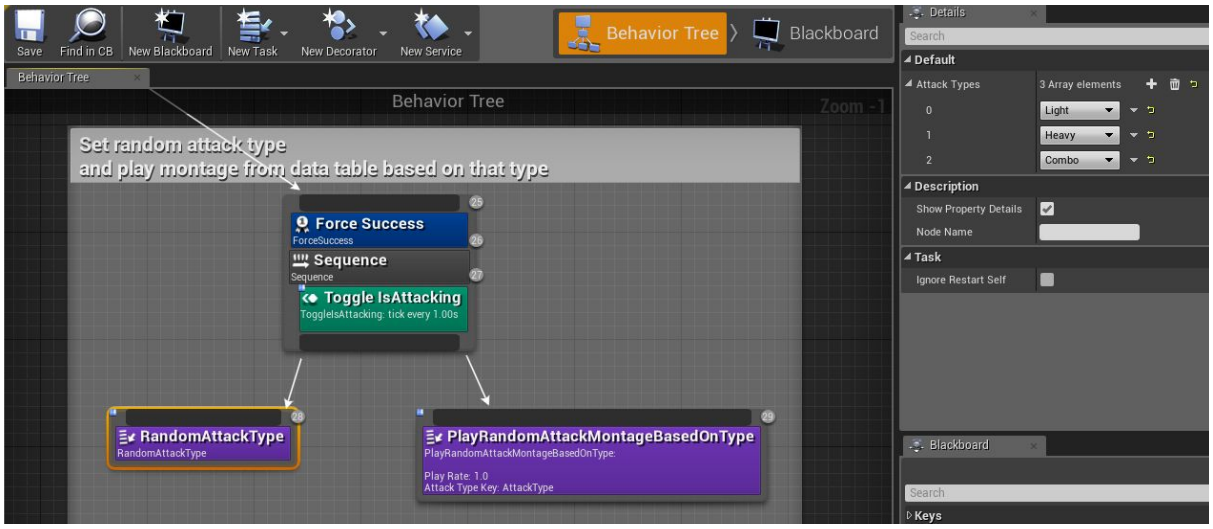
Row Name: LightAttack

Row Value
Action: LightAttack
Montages: 2 Array elements
0: AM_AI_Halberd_LightAttack_01
1: AM_AI_Halberd_LightAttack_02

Take note that RowName must have same name as Action value.

Important Montages

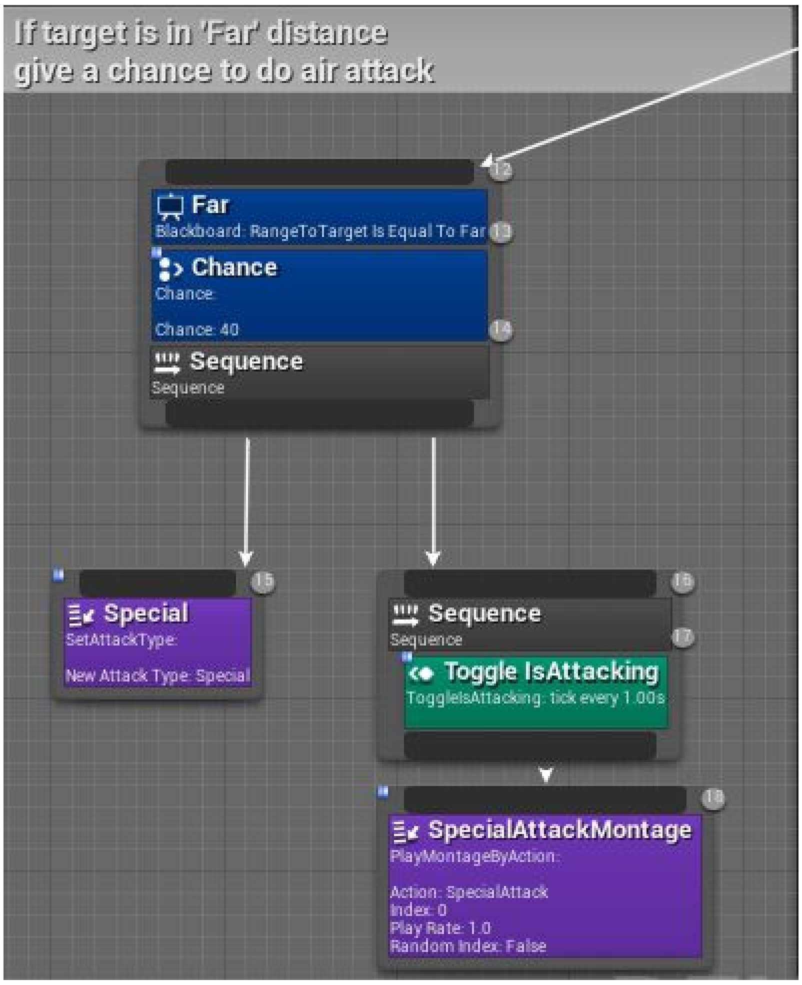
If you take a look at BT_ExampleAI, it uses light/heavy/combo/special attack types, so he will need to set those in DataTable as well.



Set random attack type
and play montage from data table based on that type

Force Success
ForceSuccess
Sequence
Sequence
Toggle IsAttacking
ToggleIsAttacking: tick every 1.00s
RandomAttackType
RandomAttackType
PlayRandomAttackMontageBasedOnType
PlayRandomAttackMontageBasedOnType
Play Rate: 1.0
Attack Type Key: AttackType

Details
Default
Attack Types: 3 Array elements
0: Light
1: Heavy
2: Combo
Description
Show Property Details: ✓
Node Name
Task
Ignore Restart Self
Blackboard
Keys


If target is in 'Far' distance
give a chance to do air attack

Far
Blackboard: RangeToTarget Is Equal To Far
Chance
Chance
Chance: 40
Sequence
Sequence
Special
SetAttackType
New Attack Type: Special
Sequence
Sequence
Toggle IsAttacking
ToggleIsAttacking: tick every 1.00s
SpecialAttackMontage
PlayMontageByAction
Action: SpecialAttack
Index: 0
Play Rate: 1.0
Random Index: False

Light Attack (Slot: FullBody, with root motion)
Heavy Attack (Slot: FullBody, with root motion)
Special Attack (Slot: FullBody, with root motion)
Combo Attack (Slot: FullBody, with root motion)
Impact (Slot: FullBody, with root motion)

Stun Front(Slot: FullBody, with root motion)
Block (Slot: UpperBody, no root motion)

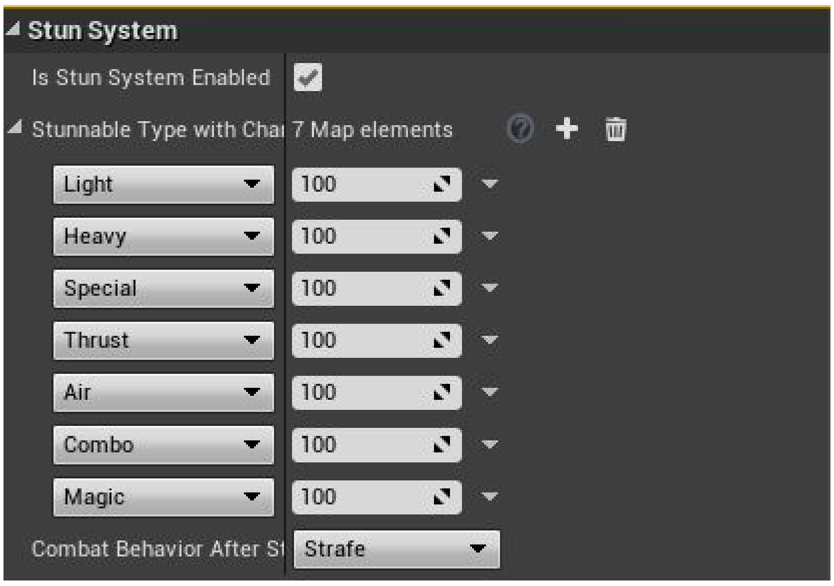
Additional Montages

For directional stun there is already implemented logic inside player character, just set animations with proper enum value and Row Name and it will work.

Stun Back (Slot: FullBody, with root motion)
Stun Left (Slot: FullBody, with root motion)
Stun Right (Slot: FullBody, with root motion)

Stun System

In AI default settings stun system can be enabled/disabled.
If it’s enabled, you can set chances for this character to get stunned and CombatBehavior after stun ends.



Stun System

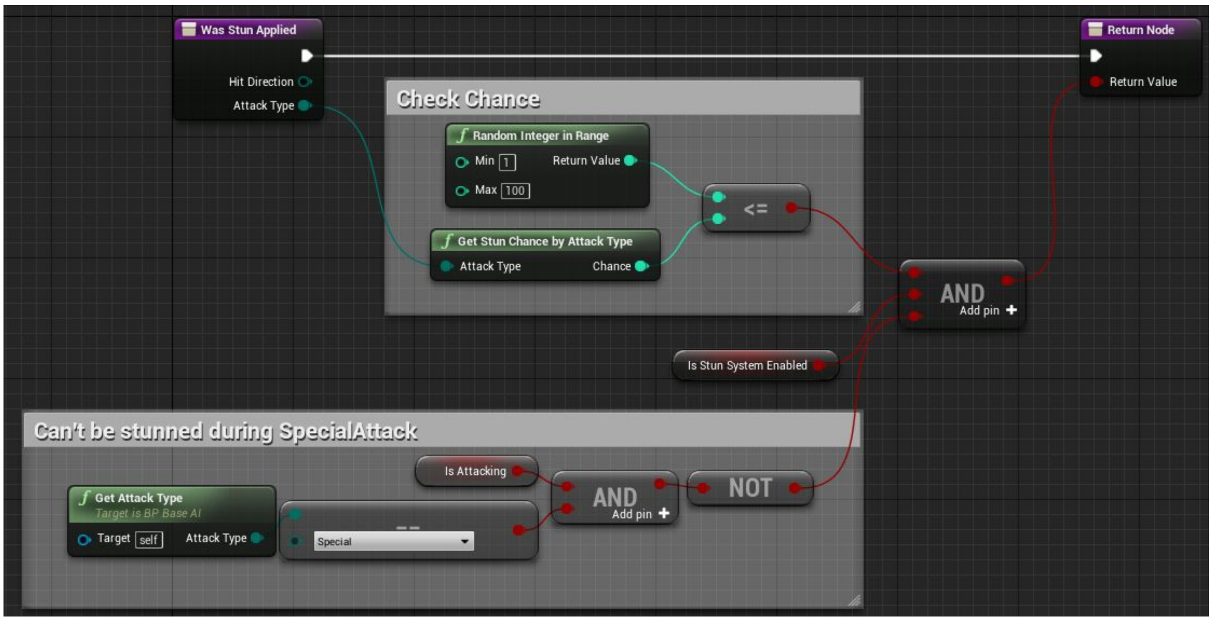
Is Stun System Enabled ✓

Stunnable Type with Char 7 Map elements

Light	100
Heavy	100
Special	100
Thrust	100
Air	100
Combo	100
Magic	100

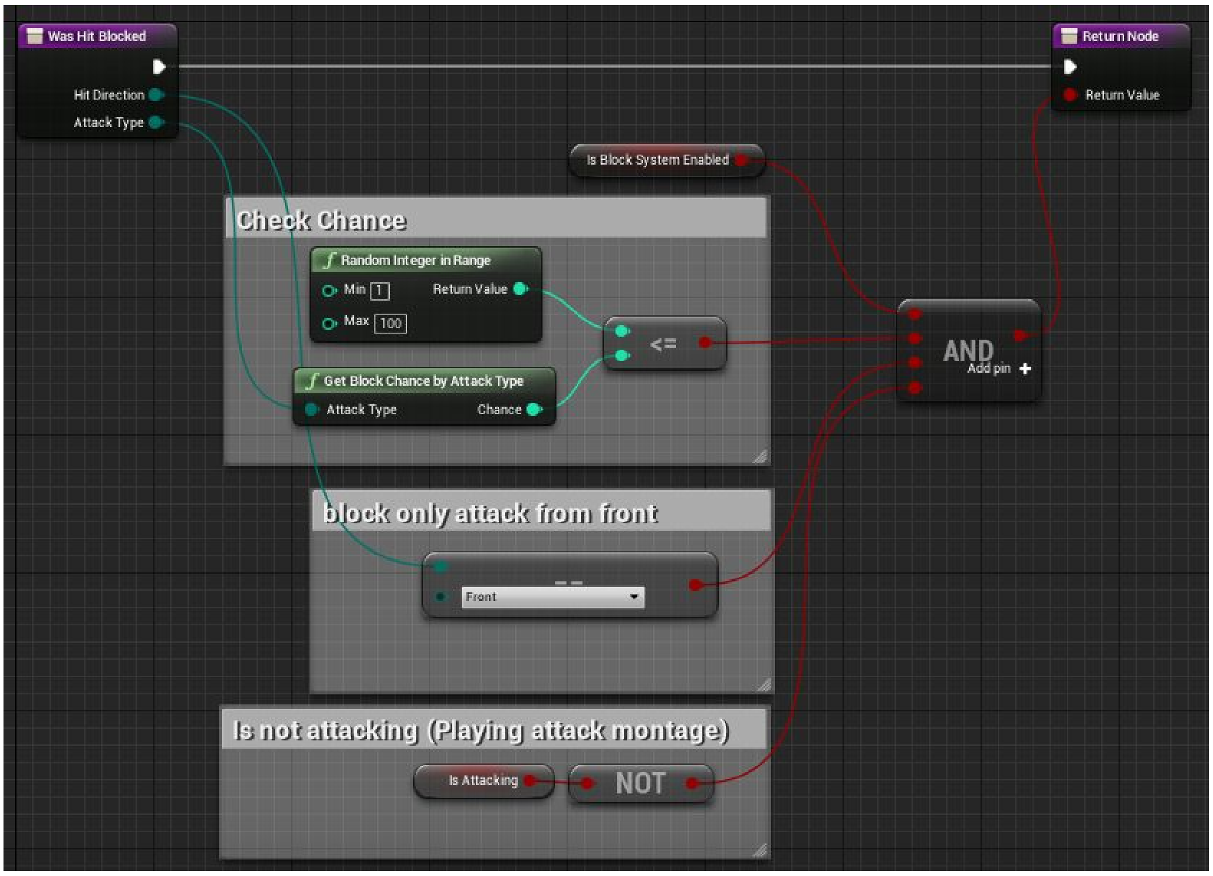
Combat Behavior After St Strafe

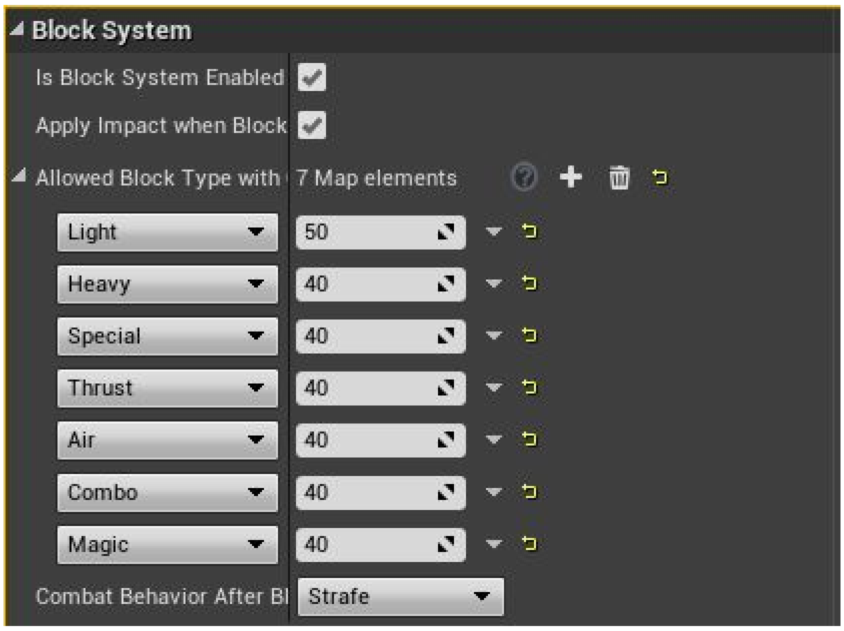
Additionally in function WasStunApplied you can add other conditions which must be fulfilled to apply stun effect
(e.g. never stun character while special attack is performed).



Block System

Very similar to stun system but to change block conditions use WasHitBlocked function instead.





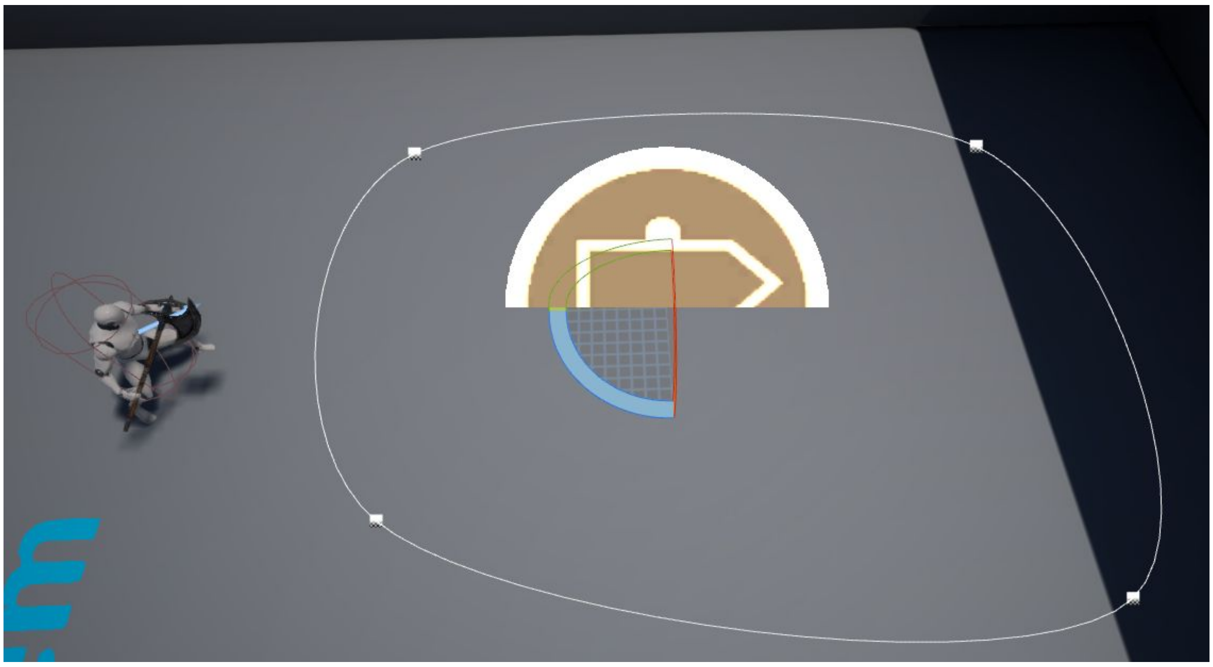
Block System

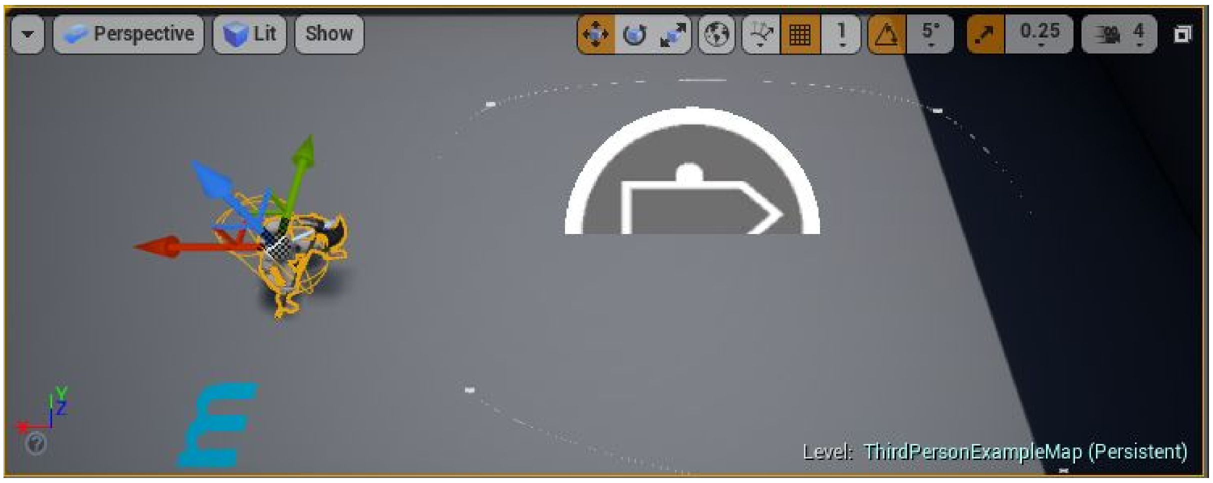
Is Block System Enabled	✓
Apply Impact when Block	✓
Allowed Block Type with	7 Map elements
Light	50
Heavy	40
Special	40
Thrust	40
Air	40
Combo	40
Magic	40
Combat Behavior After Bl	Strafe

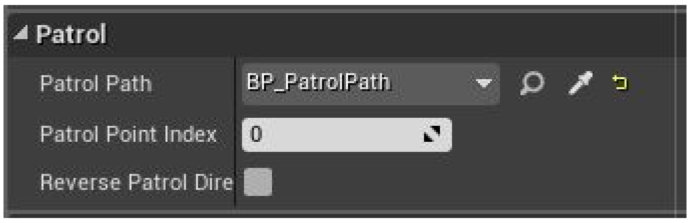
If ApplyImpactWhenBlock is enabled, Impact function from IsAttackable interface will be called on attacker.
It can be disabled in case there is Dodge/Evade block montage played instead of blocking with shield/sword etc.

Patrol

Allows AI to patrol using spline from BP_PatrolPath blueprint.
Drag and drop it into the level creating patrol spline, then set it with eye dropper in AI default settings.

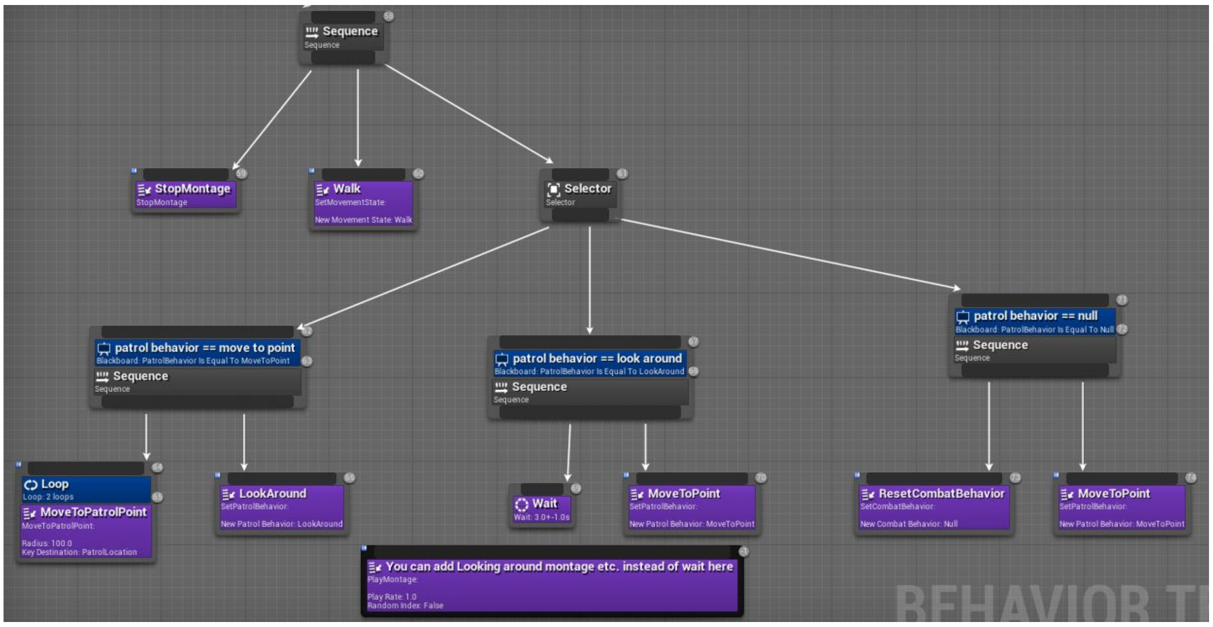





Patrol

Patrol Path	BP_PatrolPath
Patrol Point Index	0
Reverse Patrol Dire	☐

In BT_ExampleAI by default if there is not any enemy around, character will be moving towards 2 patrol points, wait and repeat.

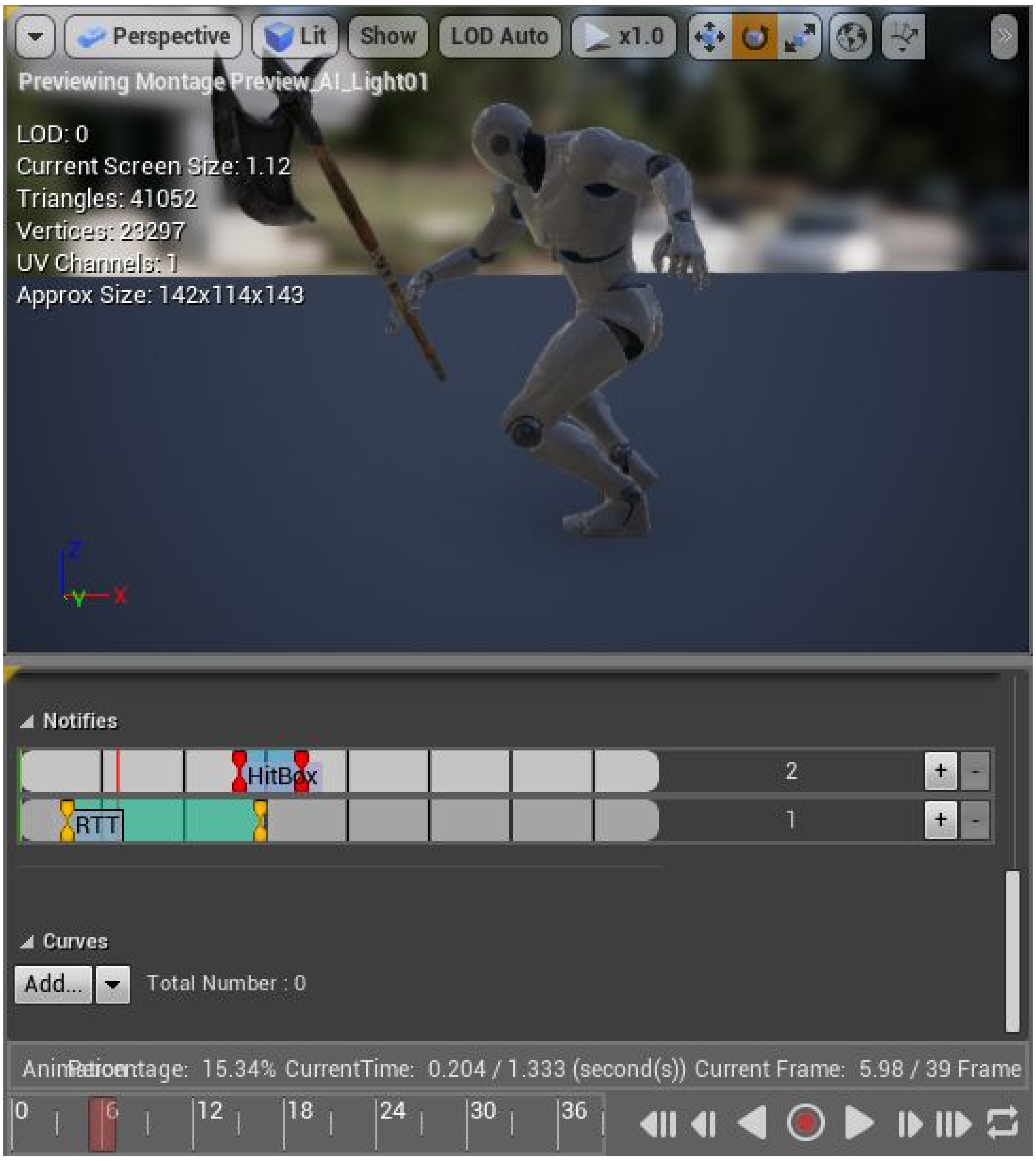


Rotate Towards Target

Allows to rotate AI smoothly towards target from blackboard.

Mostly used on attack animations.

To enable this system use ANS_RotateTowardsTarget-AI

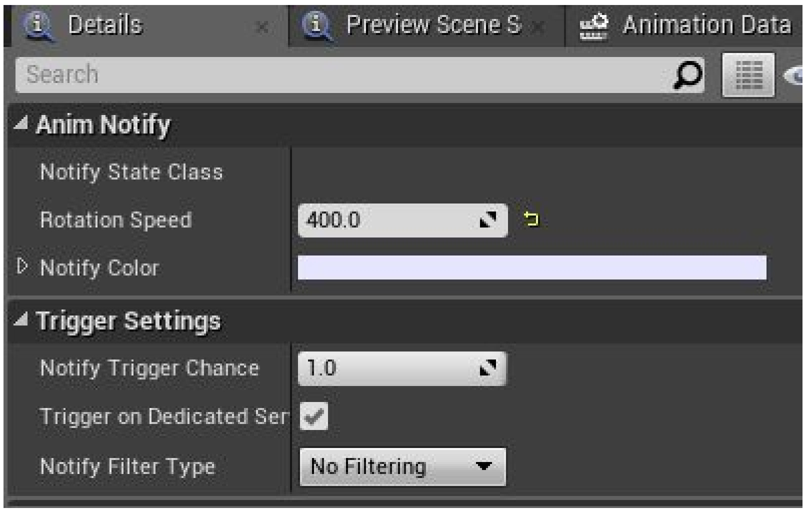


LOD: 0
Current Screen Size: 1.12
Triangles: 41052
Vertices: 23297
UV Channels: 1
Approx Size: 142x114x143

Total Number : 0

AnimMontage: 15.34% CurrentTime: 0.204 / 1.333 (second(s)) Current Frame: 5.98 / 39 Frame

Rotation speed(how fast character will be rotating) can be specifed in anim notify section.


Details
Preview Scene S
Animation Data
Search

Anim Notify

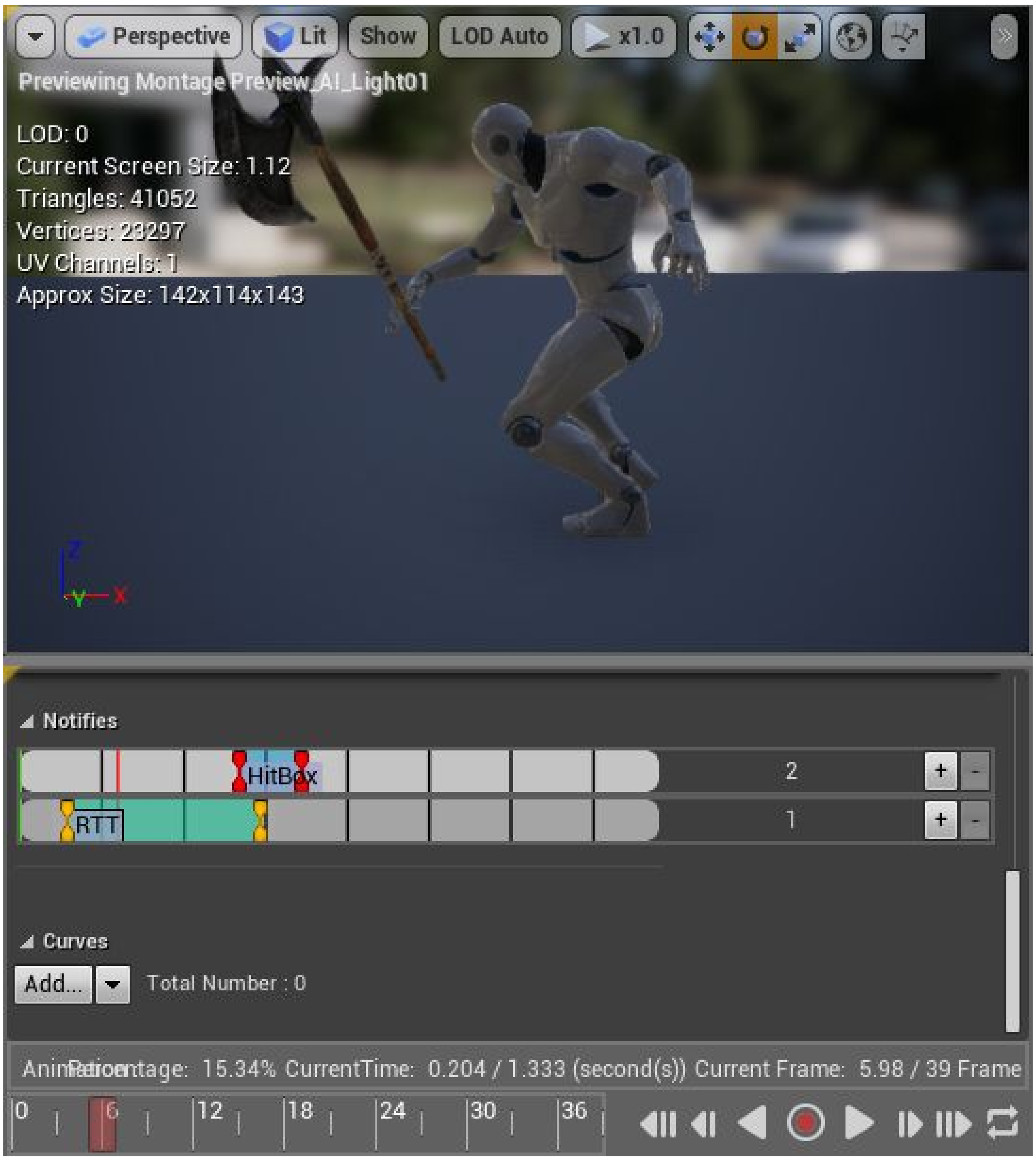
Notify State Class	
Rotation Speed	400.0
Notify Color	

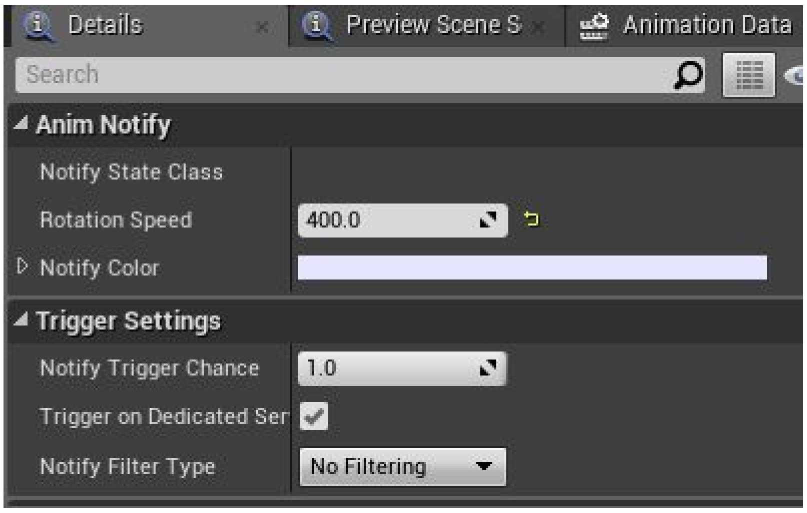
Trigger Settings

Notify Trigger Chance	1.0
Trigger on Dedicated Ser	✓
Notify Filter Type	No Filtering

It is recommended to add this state in the moment when character is moving to avoid foot sliding effect.

In example above state is added when character has 1 feet up, with rotation speed of 400.





Details | Preview Scene S | Animation Data

Search

Anim Notify

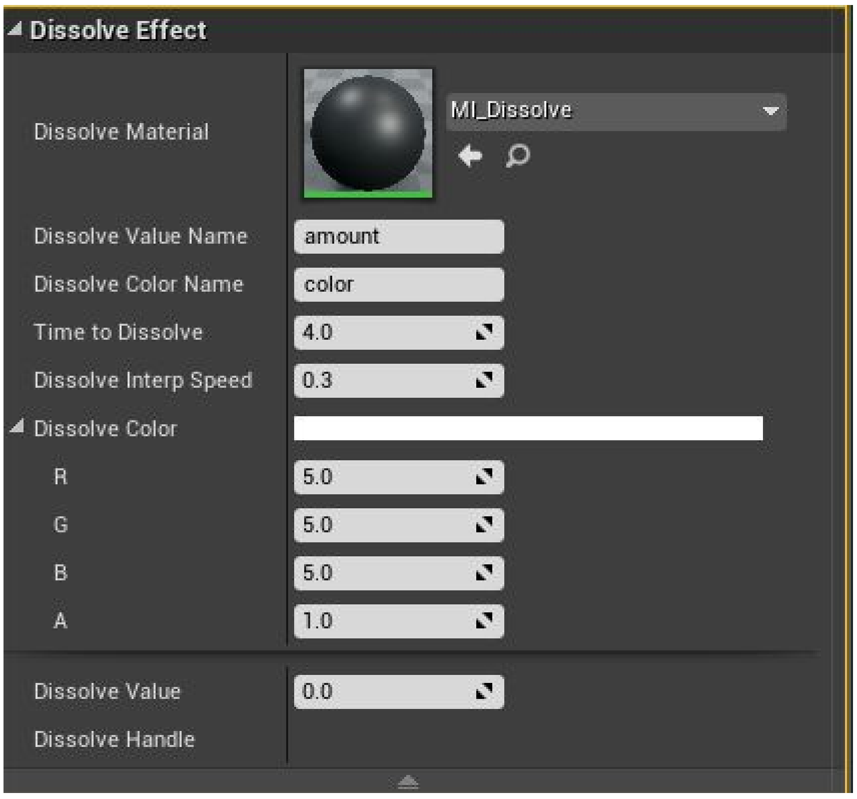
Notify State Class	
Rotation Speed	400.0
Notify Color	

Trigger Settings

Notify Trigger Chance	1.0
Trigger on Dedicated Ser	✓
Notify Filter Type	No Filtering

Dissolve Effect

Effect applied after character dies.



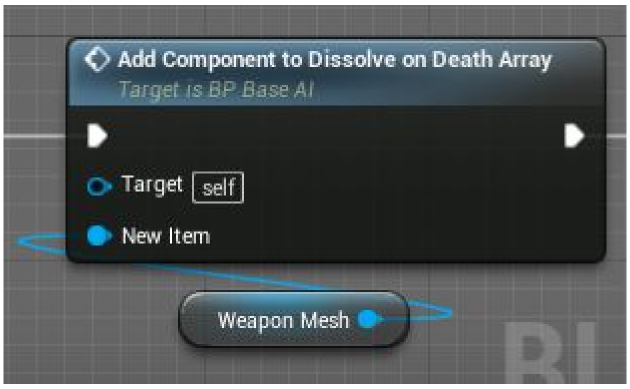
Dissolve Effect

Dissolve Material	MI_Dissolve
Dissolve Value Name	amount
Dissolve Color Name	color
Time to Dissolve	4.0
Dissolve Interp Speed	0.3
Dissolve Color	
R	5.0
G	5.0
B	5.0
A	1.0
Dissolve Value	0.0
Dissolve Handle	

DissolveValueName - parameter name of dissolve effect value material

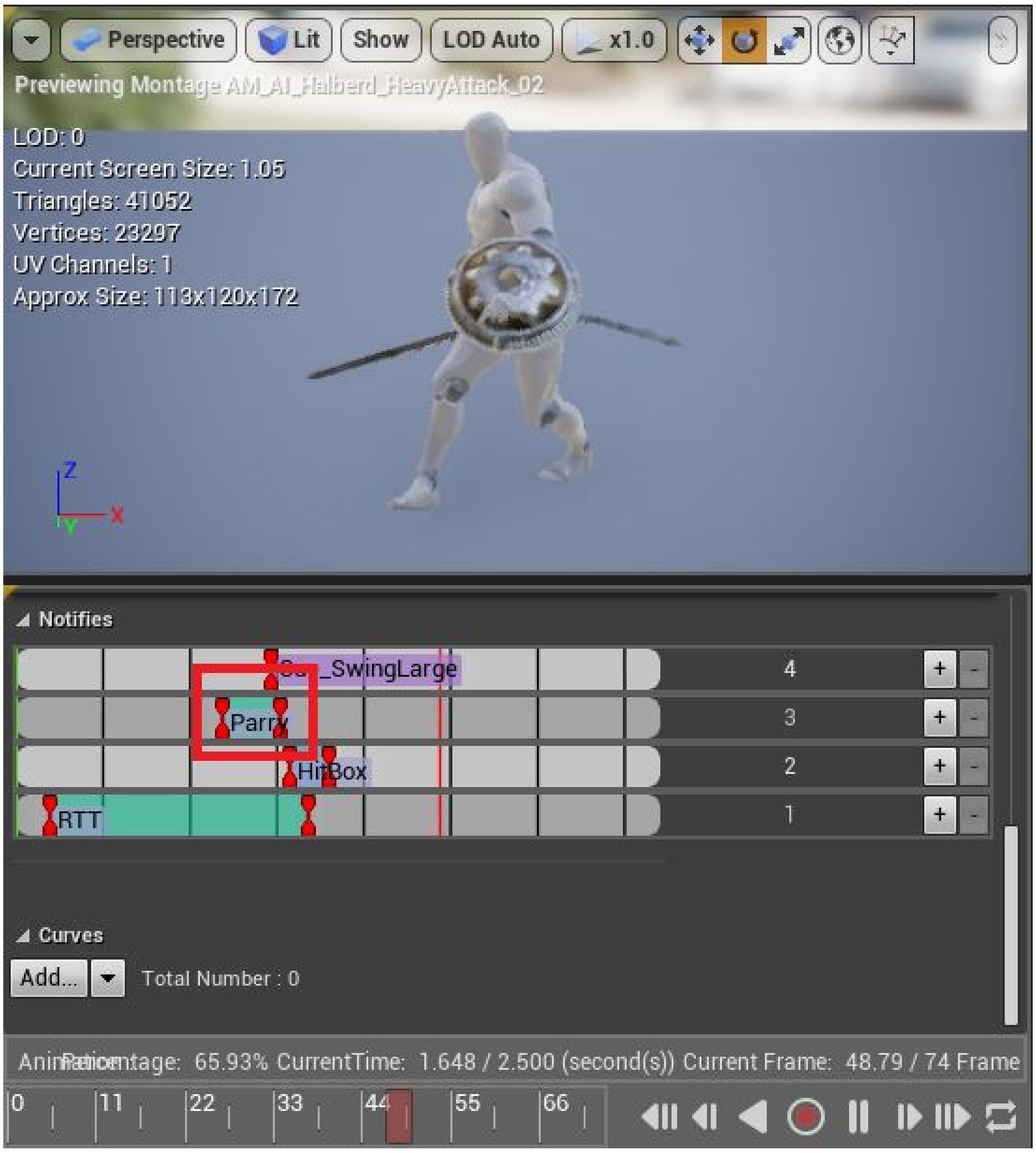
DissolveColorName - parameter name of dissolve color in material
TimeToDissolve - time after dissolve effect will start be applying
DissolveInterpSpeed - how fast dissolve material will be applied
Dissolve Color - dissolve material color

New Primitive Components(attachments like swords/shields) can be added with AddComponentToDissolveOnDeathArray

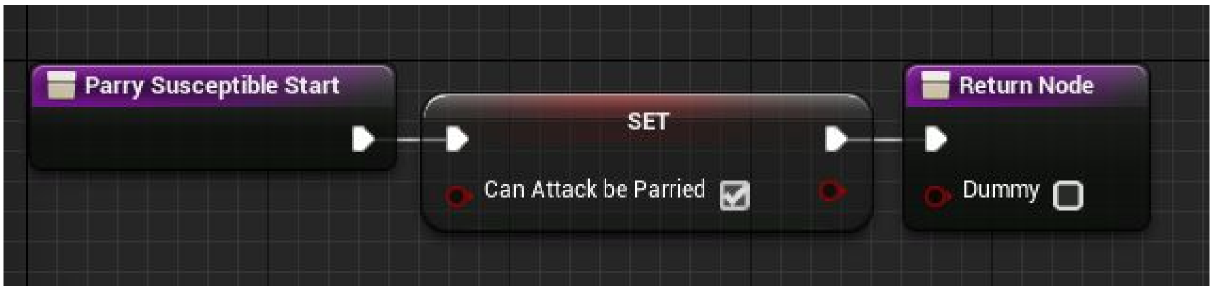


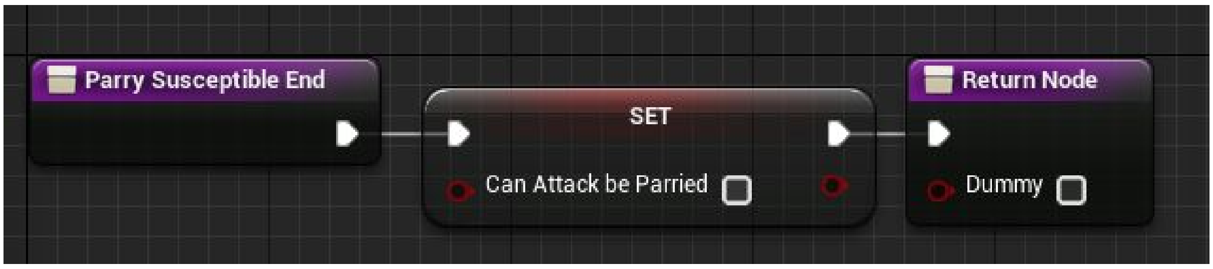
Parry

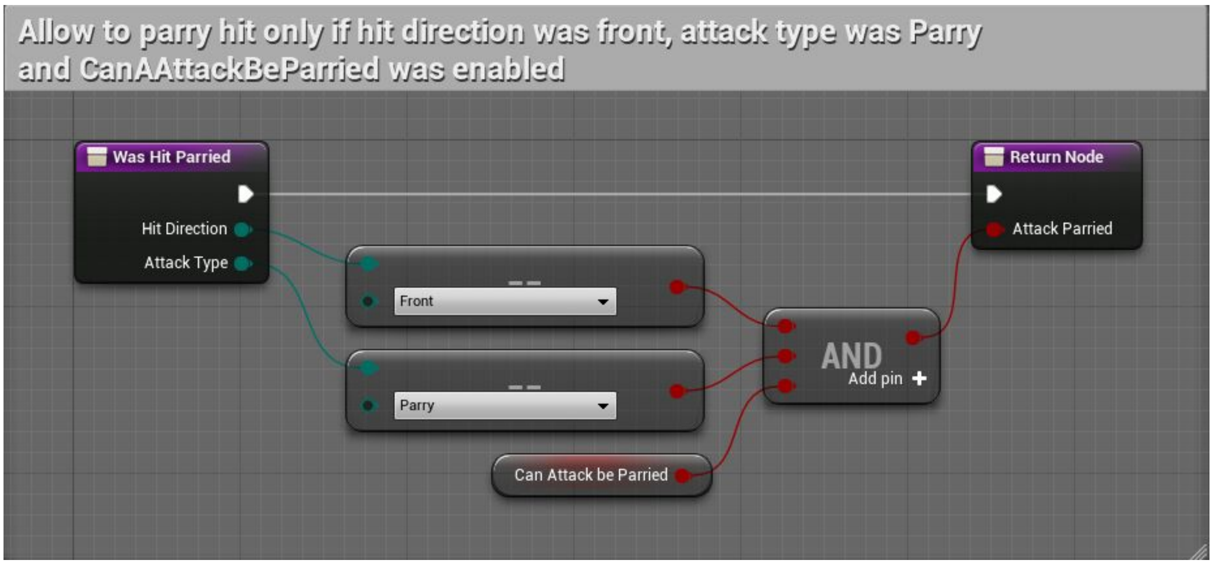
If AI is attacking and gets hit while is in notify state ANS_ParrySusceptible (and attack type was Parry), then AI gets stun and plays ParryGetHit montage from DataTable.



ANS_ParrySusceptible is calling functions from interface IsSusceptibleToParry (Parry SusceptibleStart/End)






Allow to parry hit only if hit direction was front, attack type was Parry and CanAAttackBeParried was enabled
Was Hit Parried
Hit Direction
Attack Type
Front
Parry
Can Attack be Parried
AND
Add pin
Return Node
Attack Parried

Interfaces

IsAttackable

Defines if owning class can be attacked.
Functions:
HandleHit - called when character receives damage..
Impact - called e.g. when attacked actor blocked hit.
IsAlive - should be implemented to return true if character is alive, false otherwise.

CanBeImmortal

EnableImmortality - makes character immortal
DisableImmortality - disable immortality on character

IsTargetable

Defines if character can be targeted by dynamic targeting component. More info in DynamicTargetingComponent.

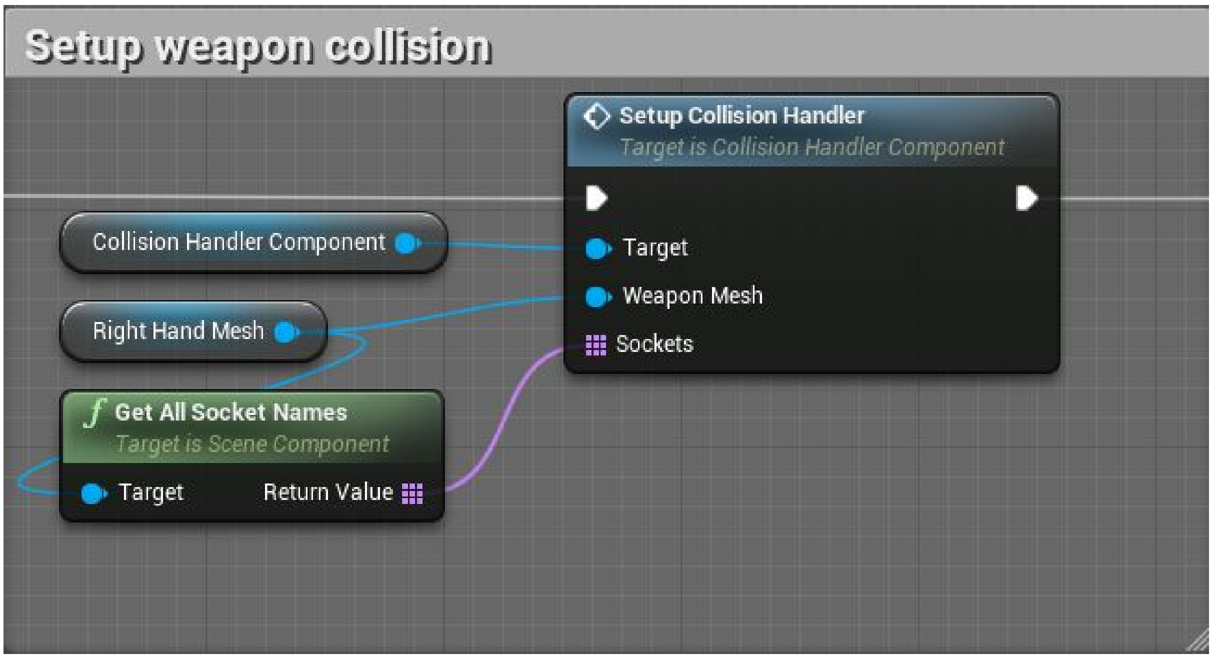
IsSusceptibleToParry

Described in Parry

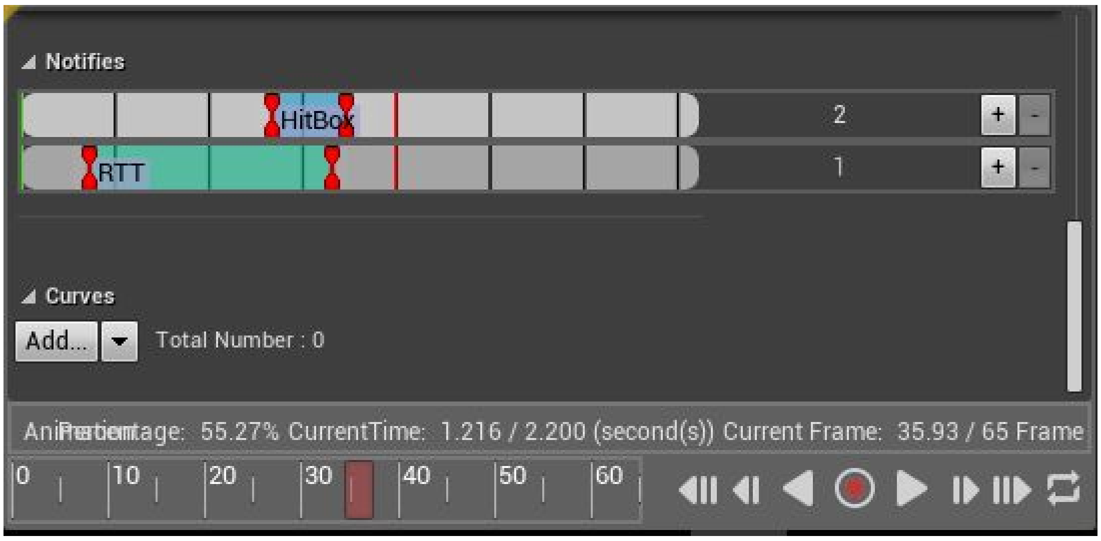
Components

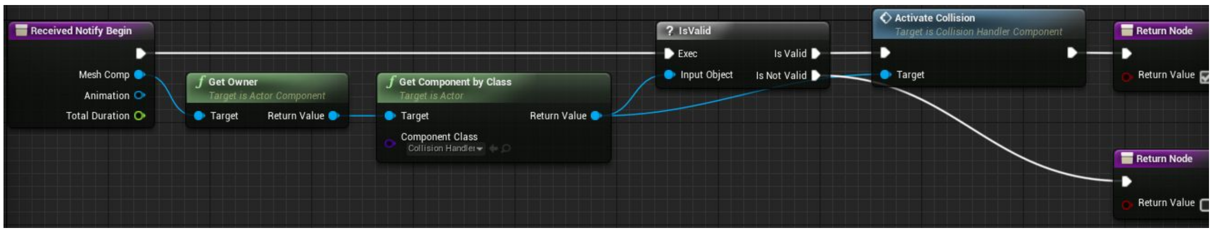
CollisionHandlerComponent

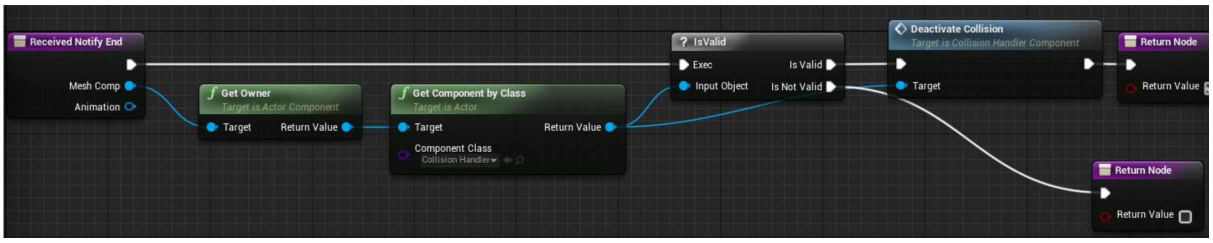
Component used in both Player and AI which allows to detect collision with other actors based on given sockets of skeletal mesh(weapon) e.g. to apply damage on them.



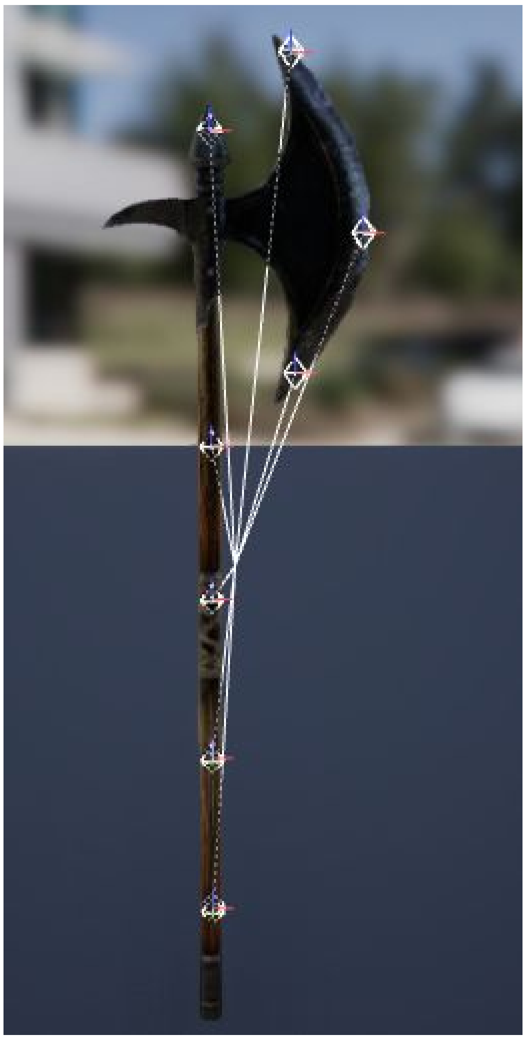
To enable/dissable detecting collision,use ANS_HitBox


Notifies
HitBox
RTT
2
1
Curves
Add...
Total Number : 0
Percentage: 55.27% CurrentTime: 1.216 / 2.200 (second(s)) Current Frame: 35.93 / 65 Frame
0
10
20
30
40
50
60

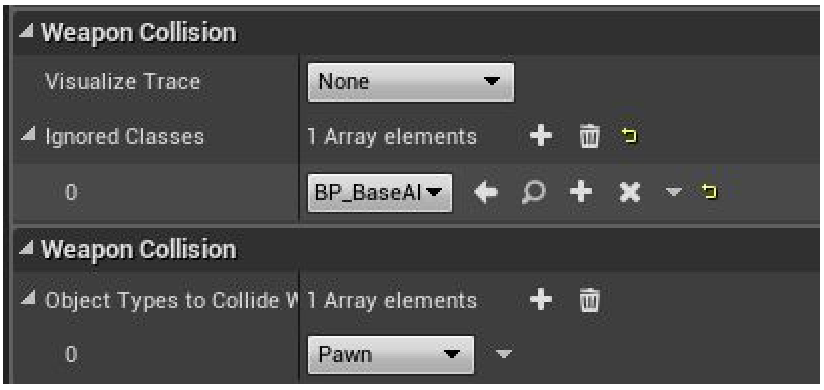




Item used by this component should be skeletal mesh.
Example weapon with few sockets on it.



To avoid calling OnHit event when collision is detected on friendly actors(companion etc.),
their class can be added to IgnoredClasses array(all classes inheriting from this class will also be ignored)



Weapon Collision

Visualize Trace	None
Ignored Classes	1 Array elements
0	BP_BaseAI

Weapon Collision

Object Types to Collide W	1 Array elements
0	Pawn

DynamicTargetingComponent

https://www.unrealengine.com/marketplace/dynamic-targeting

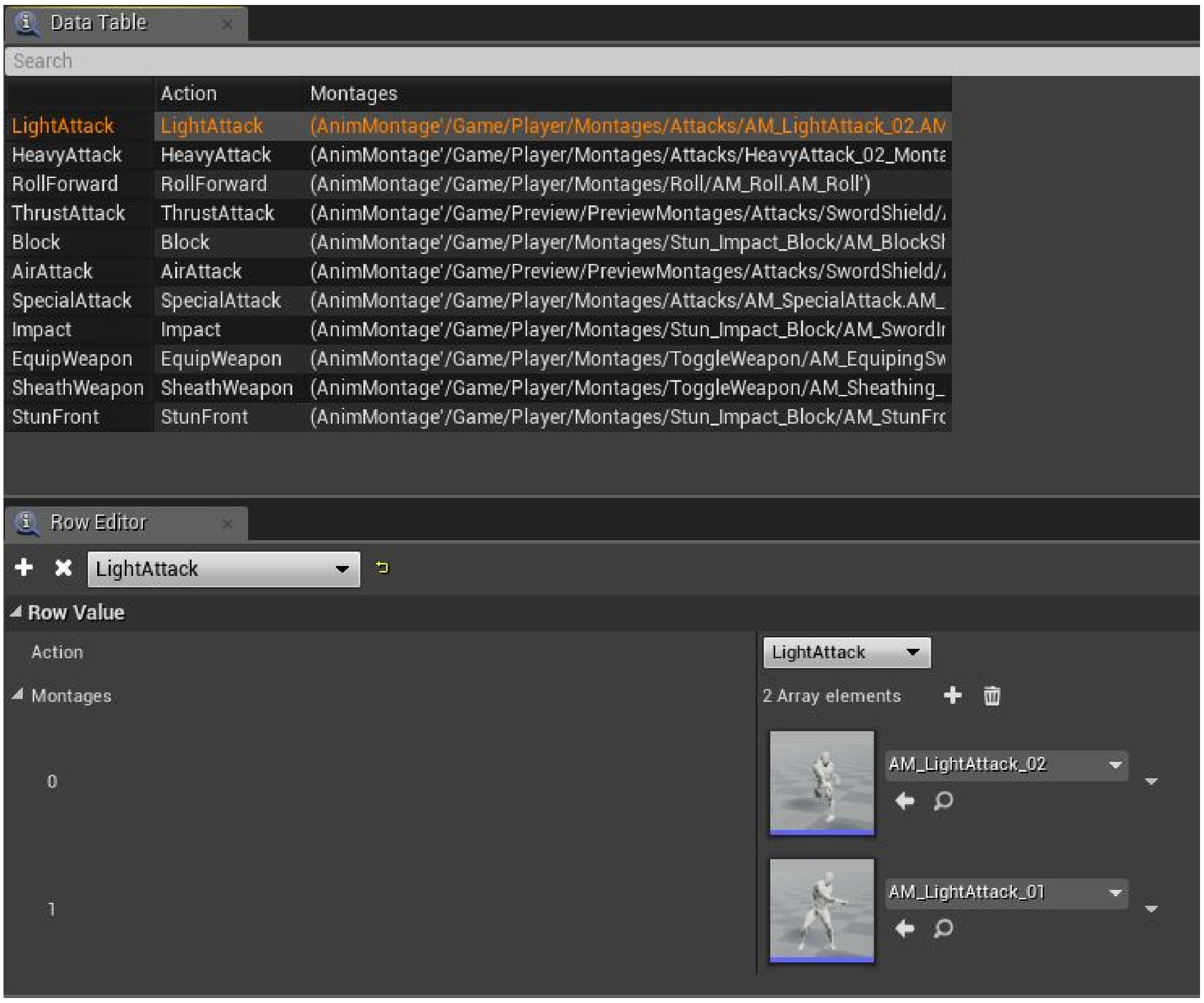
Video explaining how this component works:
https://www.youtube.com/watch?v=sp6SicLCjJQ

InputBufferComponent

Described in InputBuffer section.

MontageManagerComponent

Component used both by AI and Player, it helps to keep code clean and easily modifiable by storing all montages in one place - DataTables

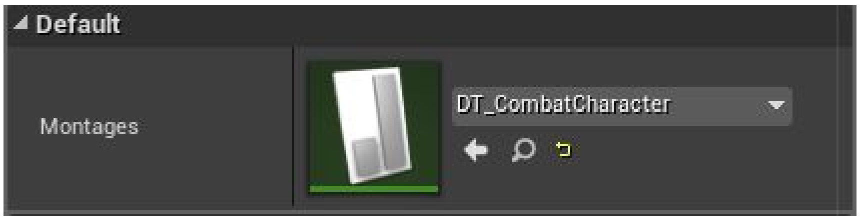


	Action	Montages
LightAttack	LightAttack	(AnimMontage'/Game/Player/Montages/Attacks/AM_LightAttack_02.AM
HeavyAttack	HeavyAttack	(AnimMontage'/Game/Player/Montages/Attacks/HeavyAttack_02_Monta
RollForward	RollForward	(AnimMontage'/Game/Player/Montages/Roll/AM_Roll.AM_Roll')
ThrustAttack	ThrustAttack	(AnimMontage'/Game/Preview/PreviewMontages/Attacks/SwordShield/
Block	Block	(AnimMontage'/Game/Player/Montages/Stun_Impact_Block/AM_BlockSl
AirAttack	AirAttack	(AnimMontage'/Game/Preview/PreviewMontages/Attacks/SwordShield/
SpecialAttack	SpecialAttack	(AnimMontage'/Game/Player/Montages/Attacks/AM_SpecialAttack.AM_
Impact	Impact	(AnimMontage'/Game/Player/Montages/Stun_Impact_Block/AM_SwordIr
EquipWeapon	EquipWeapon	(AnimMontage'/Game/Player/Montages/ToggleWeapon/AM_EquipingSw
SheathWeapon	SheathWeapon	(AnimMontage'/Game/Player/Montages/ToggleWeapon/AM_Sheathing_
StunFront	StunFront	(AnimMontage'/Game/Player/Montages/Stun_Impact_Block/AM_StunFrc

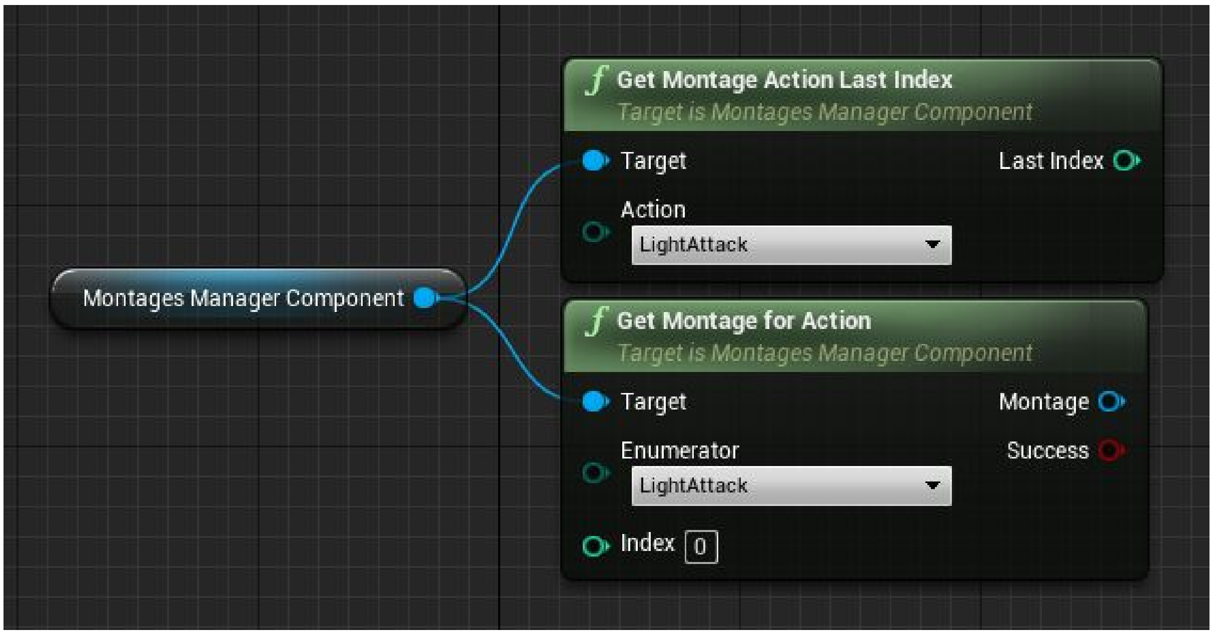
LightAttack

Action	LightAttack
Montages	2 Array elements
0	AM_LightAttack_02
1	AM_LightAttack_01

To use, set Data Table which will be used by owner.



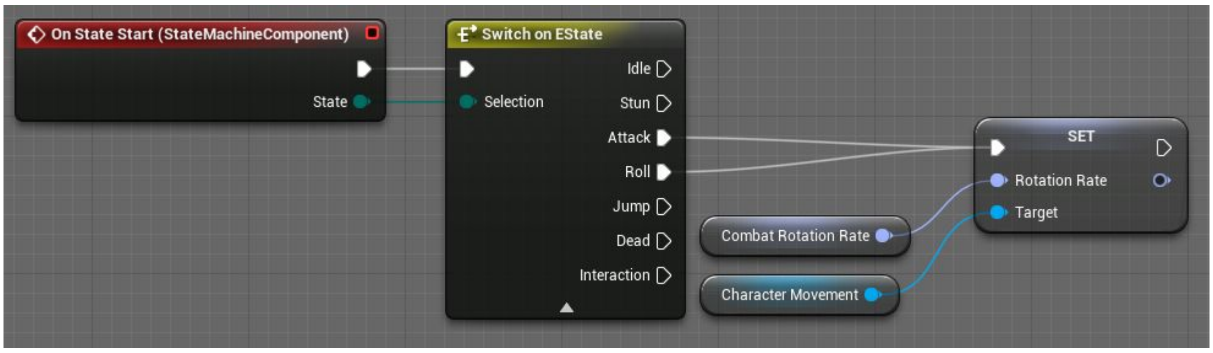
Getting Montage or last index of the array, using EMontageAction value

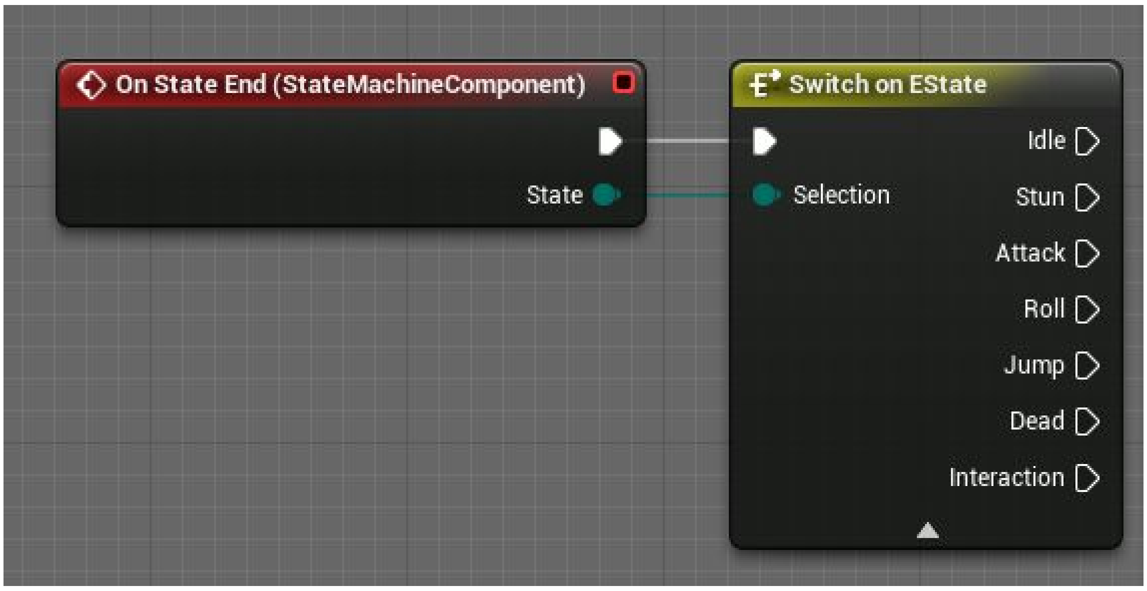

Get Montage Action Last Index
Target is Montages Manager Component
Target
Last Index
Action
LightAttack
Montages Manager Component
Get Montage for Action
Target is Montages Manager Component
Target
Montage
Enumerator
Success
LightAttack
Index
0

StateMachineComponent

Component used by player to define his current state.

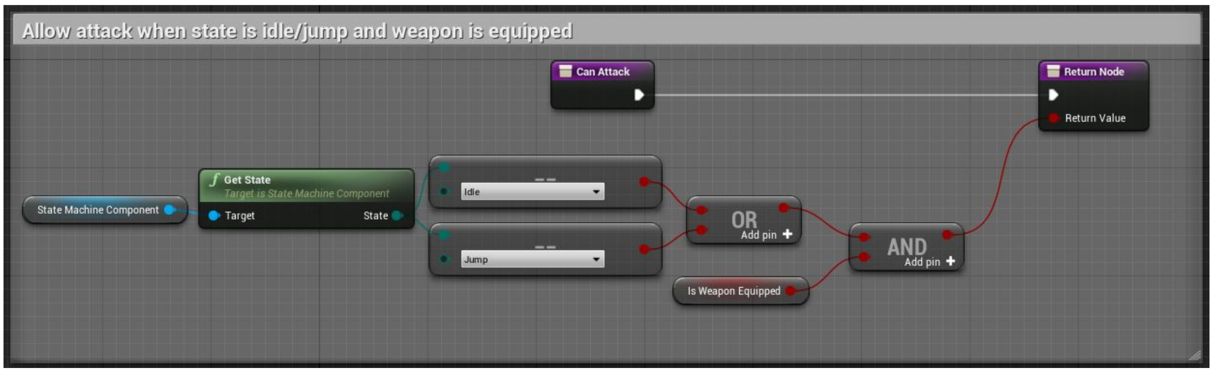
Allow to call additional functions when state ends/starts





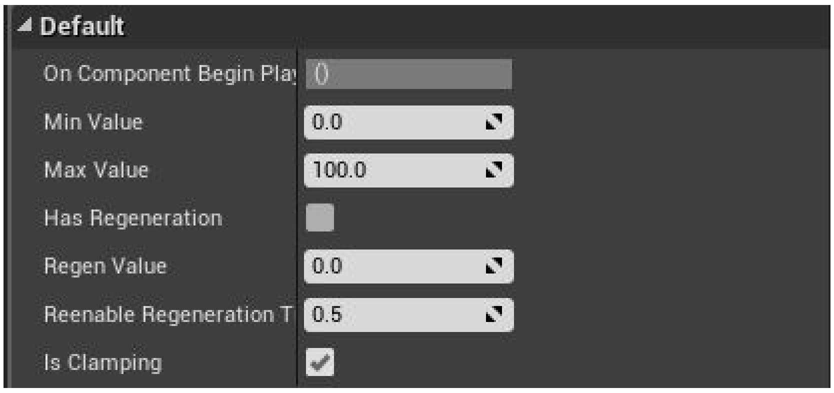
Mostly used to define what player can/can’t do in certain states.

e.g. allow attack only in Idle/Air state if weapon is equipped



StatisticComponent

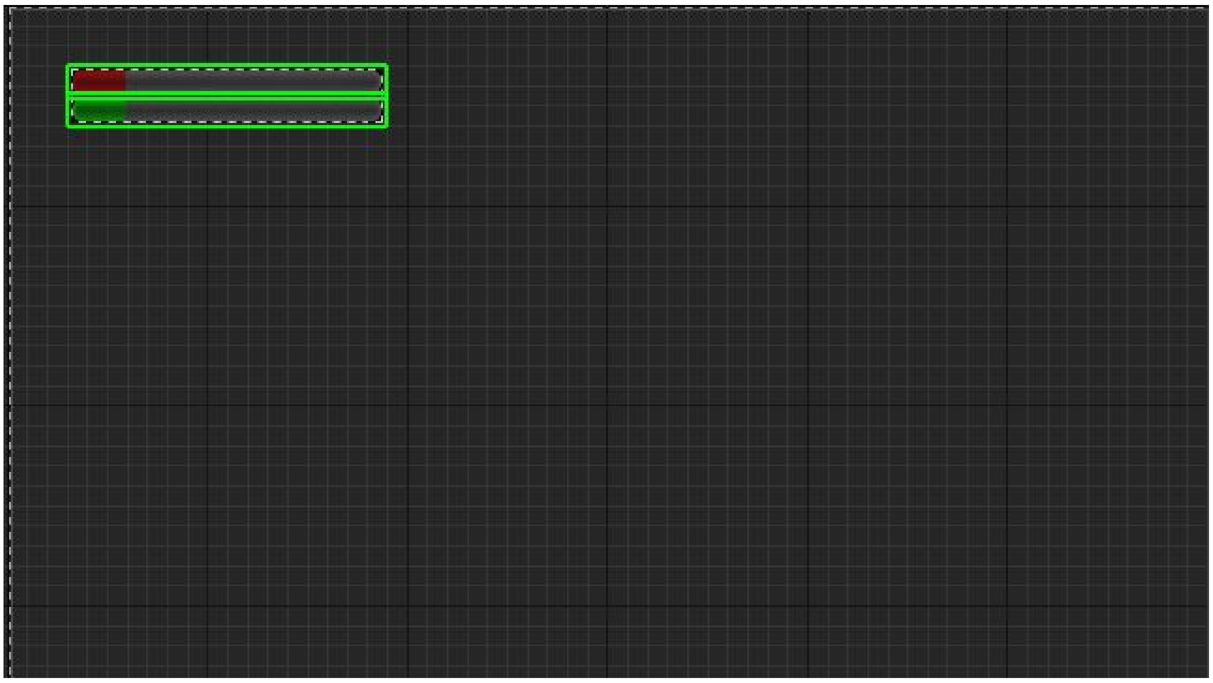
Component used both by player(stamina/health) and AI(health).
New stats like Mana/Exp can be also be added.

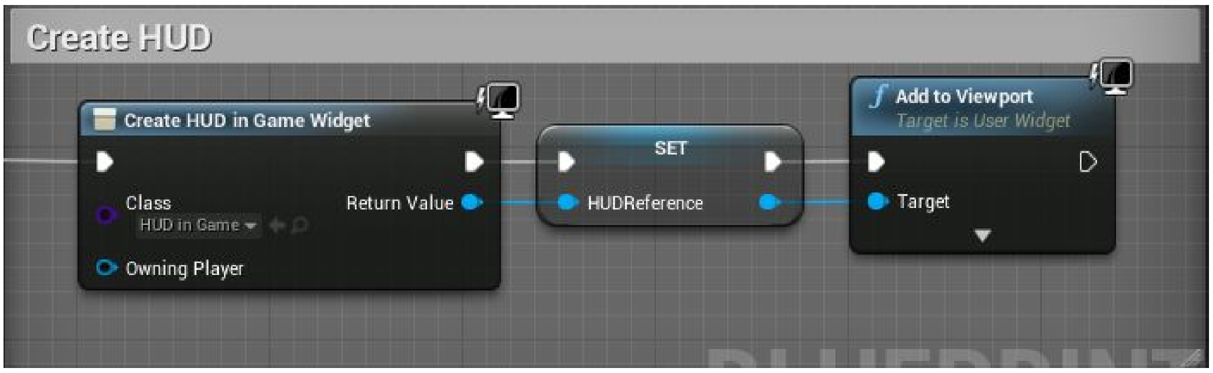


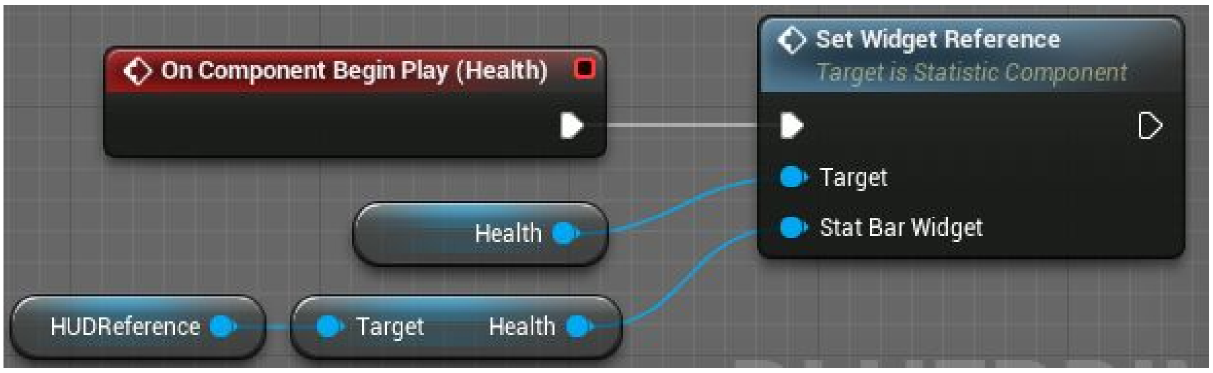
Default	
On Component Begin Play	()
Min Value	0.0
Max Value	100.0
Has Regeneration	☐
Regen Value	0.0
Reenable Regeneration T	0.5
Is Clamping	✓

Component requires to set BaseStatBar pointer in order to visualize it.

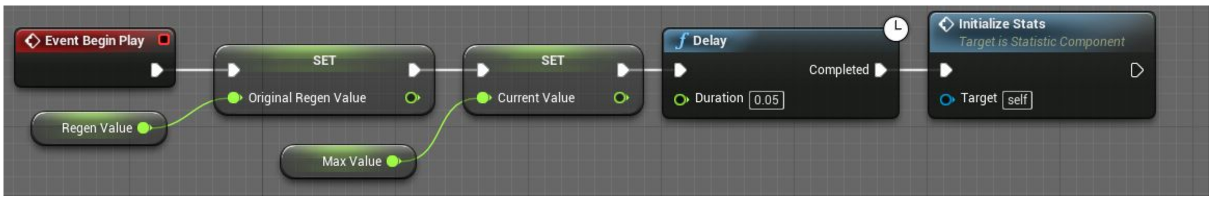
To do that use function SetWidgetReference





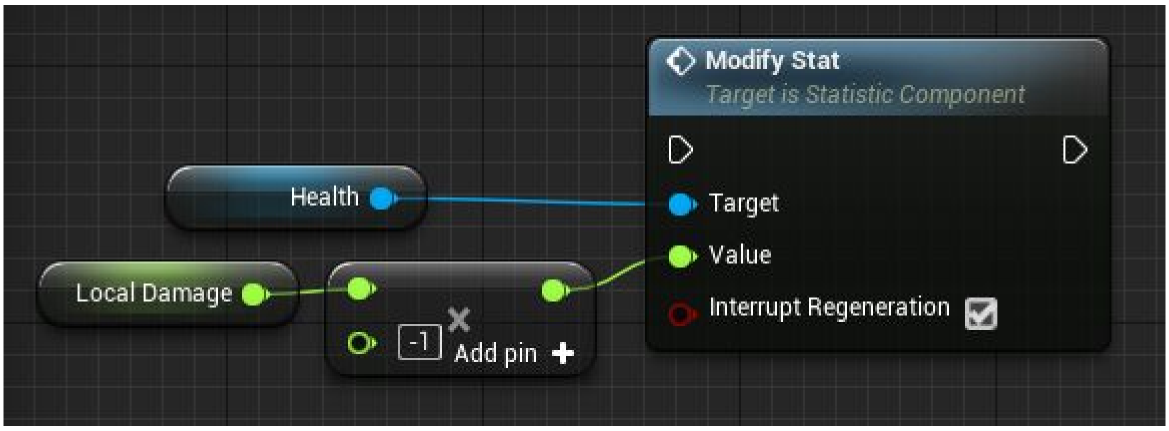


Event OnComponentBeginPlay is called with little delay, to avoid calling it before owner’s BeginPlay.



You could also initialize it on your own with above function InitializeStats which is public.

Example of modyfing stat.



Animation Notifies

AN_IgnoreRootMotion-Player

This system is described in Player’s IgnoreRootMotion section.

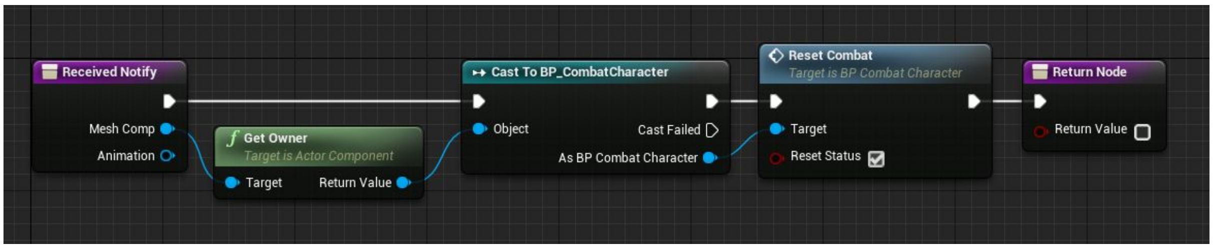
ANS_IgnoreRootMotion-Player

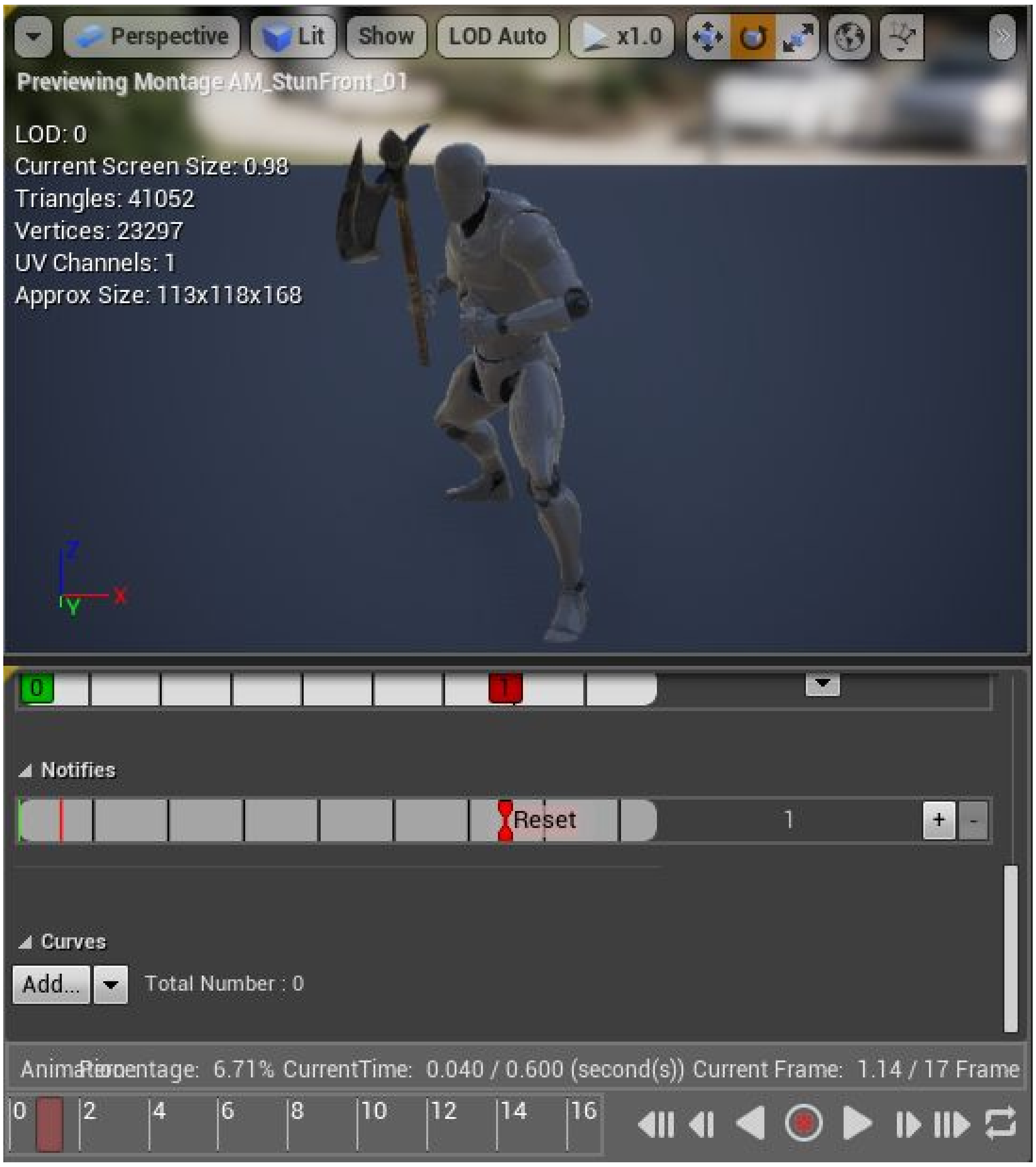
This system is described in Player’s IgnoreRootMotion section.

AN_ResetCombat-Player

Notify calling ResetCombat function from player character.

This notify should be used on Impact/Stun player's montage, instead of ANS_InputBuffer-Player.




Perspective
Lit
Show
LOD Auto
x1.0
Previewing Montage AM_StunFront_01
LOD: 0
Current Screen Size: 0.98
Triangles: 41052
Vertices: 23297
UV Channels: 1
Approx Size: 113x118x168
Notifies
Reset
Curves
Add...
Total Number : 0
Percentage: 6.71% CurrentTime: 0.040 / 0.600 (second(s)) Current Frame: 1.14 / 17 Frame

More info in InputBuffer.

ANS_ParrySusceptible

Described in Parry

ANS_HitBox

Described in CollisionHandlerComponent.

ANS_InputBuffer-Player

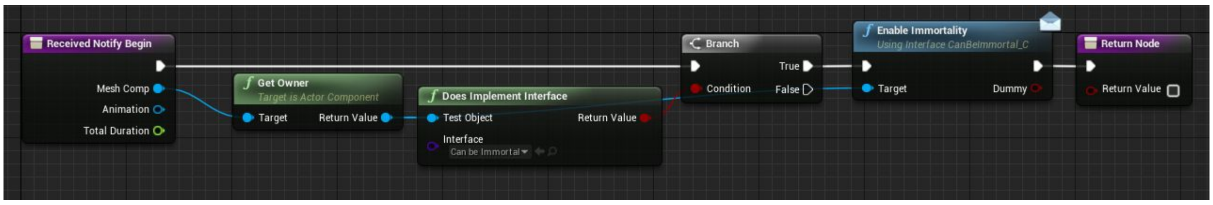
Described in InputBuffer section.

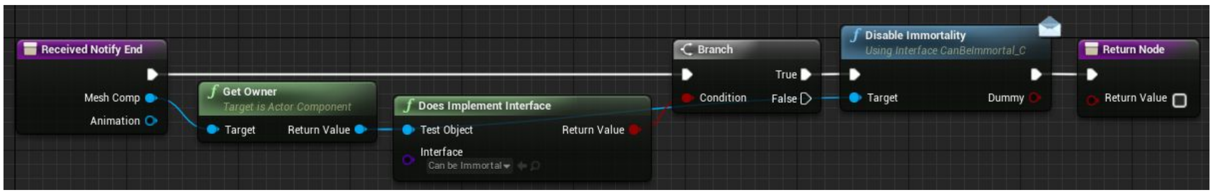
ANS_RotateTowardsTarget-AI

Described in RotateTowardsTarget.

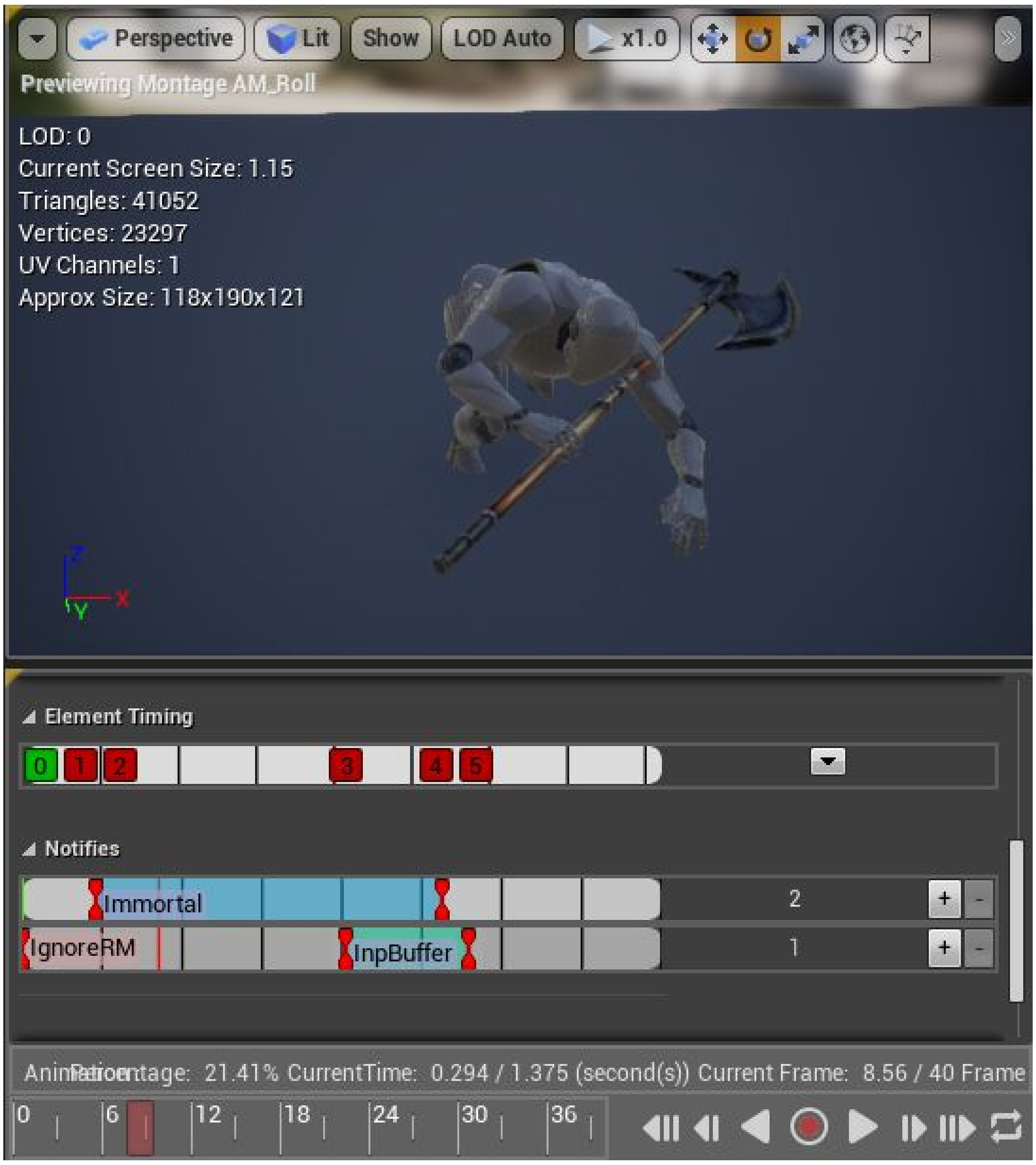
ANS_ToggleImmortality

State which calls CanBeImmortal interface functions.
On Begin - EnableImmortality
On End - DisableImmortality



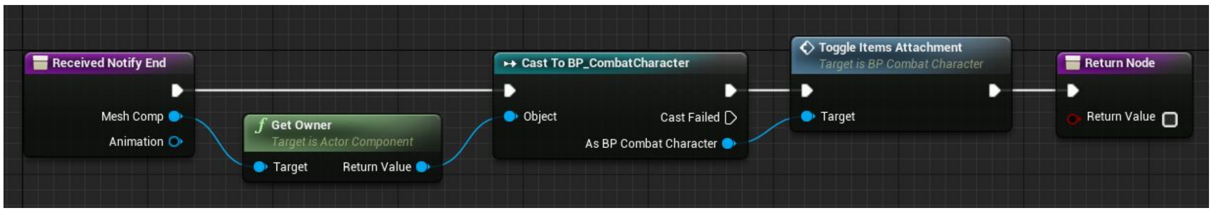


Used on roll/dash montages to make character immortal.

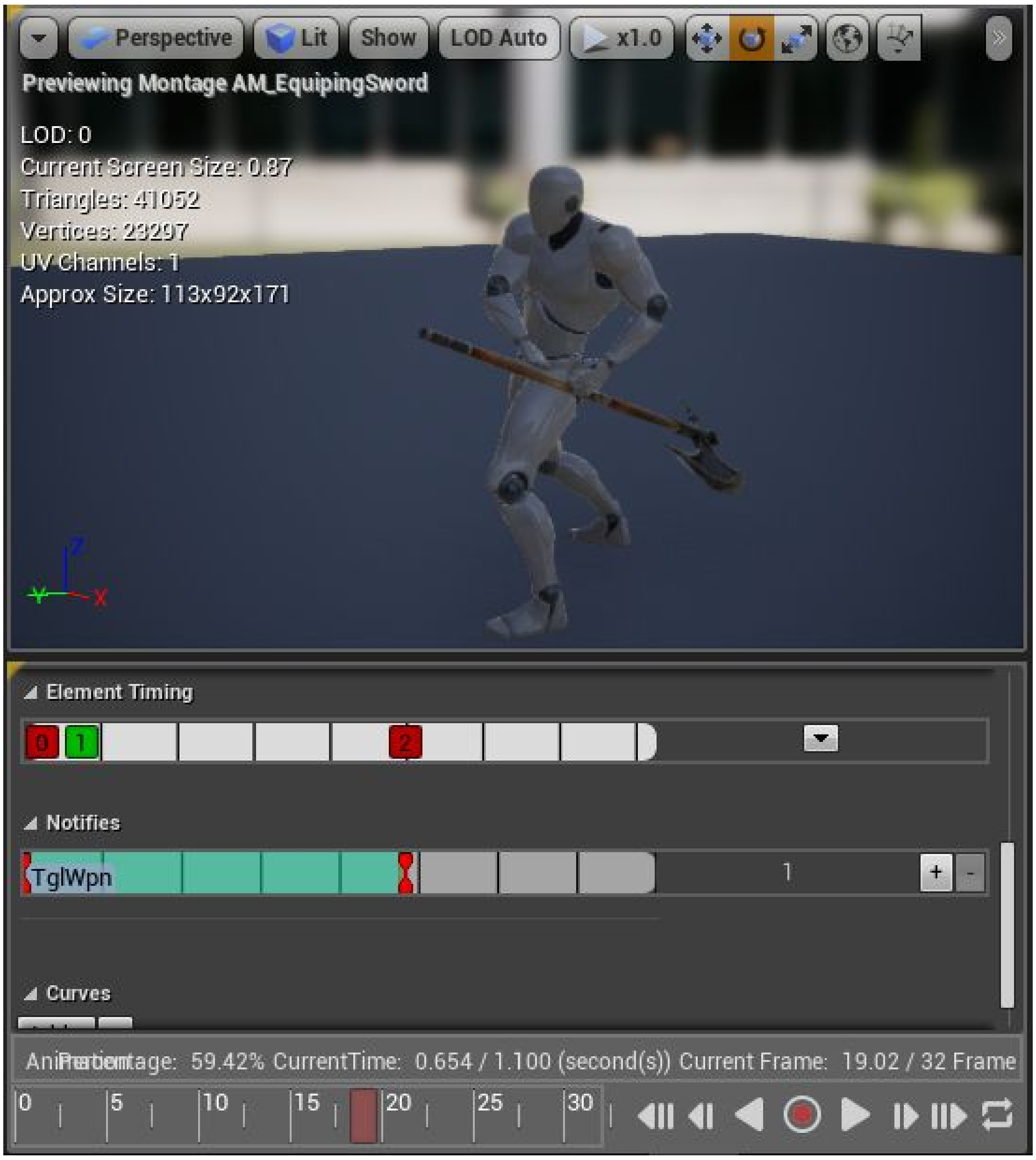


ANS_ToggleWeaponAttachment-Player

Called on equipping/sheathing montages used by player.



It is called from beginning of the montage to the moment when sockets should be changed, because even if montage will be interrupted, end event will be called and weapon will be attached to proper slot.


Perspective
Lit
Show
LOD Auto
x1.0
Previewing Montage AM_EquipingSword
LOD: 0
Current Screen Size: 0.87
Triangles: 41052
Vertices: 23297
UV Channels: 1
Approx Size: 113x92x171
Element Timing
Notifies
TglWpn
Curves
Percentage: 59.42% CurrentTime: 0.654 / 1.100 (second(s)) Current Frame: 19.02 / 32 Frame

Enums

EAICombatBehavior

Behaviors of AI while it is in combat state, e.g. approach/strafe/attack etc.
Can be used in behavior tree.

EAIHealthState

Health state of AI, low/medium/high.
Can be used in behavior tree.

EAIPatrolBehavior

Behaviors of AI while it is in combat state, e.g. look around/wait/move to waypoint etc.
Can be used in behavior tree.

EAIStatus

Main status of AI e.g. combat/patrol/Investigate/flee etc.
Can be used in behavior tree.

EAttackType

Used to perform correct attack, modify damage based on type, define which types can stun opponent or be blocked etc.

EInputBufferKey

Used in InputBufferComponent to choose action based on stored key.

EMontageAction

Used by MontageManagerComponent to choose proper montage from DataTable based on given value.
More info in MontageManagerComponent.

ERotationMode

Used by DynamicTargetingComponent.

EState

Used by StateMachineComponent to define current state of the owner. Mainly used in player blueprint.