Sticky Base Shooter

Generated by Doxygen 1.9.1

1 Todo List

Member ABaseShooter::CreateNewActionMapping (FName DesiredActionName, FKey DesiredActionKey)

Move to a controller class perhaps?

Member ABaseShooter::CreateNewAxisMapping (FName DesiredAxisName, FKey DesiredAxisKey)

Move to a controller class perhaps?

Class ABomberman

Move some of the functions of ABaseShooter into Bomberman, or rather rewrite some as virtual and implement them in child classes such as ABomberMan

Move some of the functions of ABaseShooter into Bomberman, or rather rewrite some as virtual and implement them in child classes such as ABomberMan

Currently this class is fairly bloated.

1. Move some of the functions of ABaseShooter into Bomberman, or rather rewrite some as virtual and implement them in child classes such as ABomberMan.
2. Move methods such as those regarding to input setup to the player controller class

Currently this class is fairly bloated.

1. Move some of the functions of ABaseShooter into Bomberman, or rather rewrite some as virtual and implement them in child classes such as ABomberMan.
2. Move methods such as those regarding to input setup to the player controller class

Class AStickyBaseActor

Pehaps the class needs to be extended to be be able to edit other editor uassets

Member AStickyGameMode::CheckAnyPlayerAlive ()

remove, superflous function

Member AStickyGameMode::PostLogin (APlayerController *NewPlayer) override

Register user-chosen names?

Member AStickyGameMode::RestartDeadPlayers ()

Reset player capsule orientation and reset stance?

Member AStickyGameMode::Tick (float DeltaSeconds) override

Have a timer based delay before resterting Dead players

Class AStickyGameState

Add gamestate logic

Class AStickyPlayerController

Move controller and input related code from the ABaseShooter class into here.

Move controller and input related code from ABaseShooter into the player Controller

Class AStickyPlayerState

Redesign and remove extraneous code

Class FMaterialGenerator

Extend with many variants of templates, both in regards to expression types but also package types, and opt for selective compilation with 'constexpr if's for efficiency.

Member FMaterialGenerator::CreateCelShadedExplosionMat (FString MaterialBaseName=FString("M_↩ CelExplosionMat"), FString PackageName=FString("/Game/GenMaterials/"))

Hook correct nodes toghether and then generate some banded textures to use. Also remember to set the material defaults to be able to use it as a particle material

Class FStickyFXManager

Write functions to handle creation and editing of new particle systems through c++

Write functions to handle creation and editing of new particle systems through c++

Class UAmmoComp

This component could be expaned into something more like an inventory. What would be needed is making a dedicated item_data class, hold the item_data in the inventory. Then spawn the items on server-side when brought out of inventory.

2 Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

3 Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

4 File Index

4.1 File List

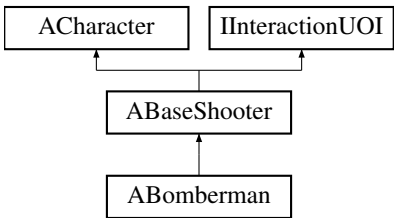
Here is a list of all files with brief descriptions:

5 Class Documentation

5.1 ABaseShooter Class Reference

```
#include <BaseShooter.h>
```

Inheritance diagram for ABaseShooter:



Public Member Functions

- ABaseShooter ()

  Construct a new ABaseShooter object.
- virtual void TryInteractItem () override

  Interact Item, Start.
- virtual void EndInteractItem () override

  Interact Item, End.
- UStickyGunSkeletalComp * GetStickyGun ()

  Get the Sticky Gun object.
- USkeletalMeshComponent * GetCharMesh ()

  Get the Char Mesh object.
- UHealthComp * GetHealthComp ()

  Get the Health Comp object.
- UAmmoComp * GetAmmoComp ()

  Get the Ammo Comp object.
- UCameraComponent * GetFirstPersonCameraComponent ()

  Get the First Person Camera Component object.
- void TryStartFire ()

  Try triggering weapon, checks ammo with server in subsequent calls.
- void ServerTriggerRagdoll ()

  Trigger ragdoll, can only be run by Server.
- void ServerUndoRagdoll ()

  Reset player/Undo Ragdoll, Can only be run by server.
- void TriggerPlayerStateAmmo (int LocalAmmoUpdate)

  Triggers an update to the players UAmmoWidget.

Public Attributes

- float BaseTurnRate
- float BaseLookUpRate

  In deg/sec. Other scaling may affect final turn rate.

Protected Member Functions

- virtual void BeginPlay ()

  Component BeginPlay.
- virtual void SetupPlayerInputComponent (UInputComponent *InputComponent) override

  Override in dervied classes to set up class-specific inputs.
- virtual float TakeDamage (float DamageAmount, FDamageEvent const &DamageEvent, AController *EventInstigator, AActor *DamageCauser) final

  TakeDamage event.
- void MulticastUndoRagdoll ()

  In deg/sec. Other scaling may affect final rate.
- void MulticastTriggerRagdoll ()

  Trigger Ragdoll function, will run multicasted.
- void ServerTryInteractItem ()

  Try Interaction function, will only run on server.
- void ServerEndInteractItem ()

  End Interaction function, will only run on server.

- void ServerTakeDamage (AActor *ThisActor, float DamageAmount, const UDamageType *DamageType, AController *EventInstigator, AActor *DamageCauser)

  TakeDamage function, will only run on server.
- void InitSkeletalBody ()

  Initializes the Skeletal mesh component and it's mesh.
- void InitCamera ()

  Initializes the Camera component.
- void InitActorComponents ()

  Initialize various actor components which aren't initialized in a function of their own.
- void SetupStickyGun ()

  Construct a AStickyGunSkeletalComp and call some intializing functions on it.
- void SetupCollision ()

  Sets up collision channels and collision settings.
- void MoveRight (float Val)
- void MoveForward (float Val)
- void TurnAtRate (float Rate)

  Turn Rate.
- void LookUpAtRate (float Rate)

  LookUp Rate.
- void CreateNewAxisMapping (FName DesiredAxisName, FKey DesiredAxisKey)

  Create and store New Axis Mapping.
- void CreateNewActionMapping (FName DesiredActionName, FKey DesiredActionKey)

  Create and store New Action Mapping.

Protected Attributes

- USkeletalMeshComponent * MeshPtr = nullptr
- UStickyGunSkeletalComp * StickyGun = nullptr

  SkelMesh: 1st person view arms.
- UCameraComponent * FirstPersonCameraComponent = nullptr

  SkelMesh: Skeletal Gun mesh.
- UHealthComp * HealthComponent = nullptr

  First person camera.
- UAmmoComp * AmmoComp = nullptr

  Networked health component.
- UStickyLinetraceComp * LinetraceComp = nullptr

  Networked ammo component.
- bool bIsRagdoll = false

  Interaction detection component, uses linetrace.

5.1.1 Constructor & Destructor Documentation

5.1.1.1 ABaseShooter() `ABaseShooter::ABaseShooter ( )`

Construct a new ABaseShooter object.

5.1.2 Member Function Documentation

5.1.2.1 BeginPlay() `void ABaseShooter::BeginPlay ( ) [protected], [virtual]`

Component BeginPlay.

Runs when world at BeginPlay, or when constructed in an already running world.

Inherited Methods: Overrides

Reimplemented in ABomberman.

5.1.2.2 CreateNewActionMapping() `void ABaseShooter::CreateNewActionMapping ( FName DesiredActionName, FKey DesiredActionKey ) [protected]`

Create and store New Action Mapping.

Parameters

DesiredActionName	
DesiredActionKey	

Todo Move to a controller class perhaps?

5.1.2.3 CreateNewAxisMapping() `void ABaseShooter::CreateNewAxisMapping ( FName DesiredAxisName, FKey DesiredAxisKey ) [protected]`

Create and store New Axis Mapping.

Parameters

DesiredAxisName	
DesiredAxisKey	

Todo Move to a controller class perhaps?

5.1.2.4 EndInteractItem() `void ABaseShooter::EndInteractItem ( ) [override], [virtual]`

Interact Item, End.

Implements IInteractionUOI.

5.1.2.5 GetAmmoComp() `UAmmoComp * ABaseShooter::GetAmmoComp ( )`

Get the Ammo Comp object.

Returns

UAmmoComp*

5.1.2.6 GetCharMesh() `USkeletalMeshComponent * ABaseShooter::GetCharMesh ( )`

Get the Char Mesh object.

Returns

USkeletalMeshComponent*

5.1.2.7 GetFirstPersonCameraComponent() `UCameraComponent * ABaseShooter::GetFirstPerson↩ CameraComponent ( )`

Get the First Person Camera Component object.

Returns

UCameraComponent*

5.1.2.8 GetHealthComp() `UHealthComp * ABaseShooter::GetHealthComp ( )`

Get the Health Comp object.

Returns

UHealthComp*

5.1.2.9 GetStickyGun() `UStickyGunSkeletalComp * ABaseShooter::GetStickyGun ( )`

Get the Sticky Gun object.

Returns

UStickyGunSkeletalComp*

5.1.2.10 InitActorComponents() `void ABaseShooter::InitActorComponents ( ) [protected]`

Initialize various actor components which aren't initialized in a function of their own.

5.1.2.11 InitCamera() `void ABaseShooter::InitCamera ( ) [protected]`

Initializes the Camera component.

5.1.2.12 InitSkeletalBody() `void ABaseShooter::InitSkeletalBody ( ) [protected]`

Initializes the Skeletal mesh component and it's mesh.

5.1.2.13 LookUpAtRate() `void ABaseShooter::LookUpAtRate ( float Rate ) [protected]`

LookUp Rate.

Called via input to turn look up/down at a given rate.

Parameters

Rate	This is a normalized rate, i.e. 1.0 means 100% of desired turn rate

5.1.2.14 MoveForward() `void ABaseShooter::MoveForward ( float Val ) [protected]`

5.1.2.15 MoveRight() `void ABaseShooter::MoveRight ( float Val ) [protected]`

5.1.2.16 MulticastTriggerRagdoll() void ABaseShooter::MulticastTriggerRagdoll ( ) [protected]

Trigger Ragdoll function, will run multicasted.

5.1.2.17 MulticastUndoRagdoll() void ABaseShooter::MulticastUndoRagdoll ( ) [protected]

In deg/sec. Other scaling may affect final rate.

Undo Ragdoll function, will run multicasted

5.1.2.18 ServerEndInteractItem() void ABaseShooter::ServerEndInteractItem ( ) [protected]

End Interaction function, will only run on server.

5.1.2.19 ServerTakeDamage()

```
void ABaseShooter::ServerTakeDamage (
        AActor * ThisActor,
        float DamageAmount,
        const UDamageType * DamageType,
        AController * EventInstigator,
        AActor * DamageCauser ) [protected]
```

TakeDamage function, will only run on server.

Parameters

ThisActor	
DamageAmount	
DamageType	
EventInstigator	
DamageCauser	

5.1.2.20 ServerTriggerRagdoll() void ABaseShooter::ServerTriggerRagdoll ( )

Trigger ragdoll, can only be run by Server.

5.1.2.21 ServerTryInteractItem() void ABaseShooter::ServerTryInteractItem ( ) [protected]

Try Interaction function, will only run on server.

5.1.2.22 ServerUndoRagdoll() `void ABaseShooter::ServerUndoRagdoll ( )`

Reset player/Undo Ragdoll, Can only be run by server.

5.1.2.23 SetupCollision() `void ABaseShooter::SetupCollision ( ) [protected]`

Sets up collision channels and collision settings.

5.1.2.24 SetupPlayerInputComponent() `void ABaseShooter::SetupPlayerInputComponent ( UInputComponent * InputComponent ) [override], [protected], [virtual]`

Override in dervied classes to set up class-specific inputs.

Parameters

InputComponent	

Reimplemented in ABomberman.

5.1.2.25 SetupStickyGun() `void ABaseShooter::SetupStickyGun ( ) [protected]`

Construct a AStickyGunSkeletalComp and call some intializing functions on it.

5.1.2.26 TakeDamage() `float ABaseShooter::TakeDamage ( float DamageAmount, FDamageEvent const & DamageEvent, AController * EventInstigator, AActor * DamageCauser ) [final], [protected], [virtual]`

TakeDamage event.

Parameters

DamageAmount	
DamageEvent	
EventInstigator	
DamageCauser	

Returns

float

5.1.2.27 TriggerPlayerStateAmmo() `void ABaseShooter::TriggerPlayerStateAmmo ( int LocalAmmoUpdate )`

Triggers an update to the players UAmmoWidget.

Parameters

LocalAmmoUpdate	

5.1.2.28 TryInteractItem() `void ABaseShooter::TryInteractItem ( ) [override], [virtual]`

Interact Item, Start.

Implements IInteractionUOI.

5.1.2.29 TryStartFire() `void ABaseShooter::TryStartFire ( )`

Try triggering weapon, checks ammo with server in subsequent calls.

5.1.2.30 TurnAtRate() `void ABaseShooter::TurnAtRate ( float Rate ) [protected]`

Turn Rate.

Called via input to turn at a given rate.

Parameters

Rate	This is a normalized rate, i.e. 1.0 means 100% of desired turn rate

5.1.3 Member Data Documentation

5.1.3.1 AmmoComp `UAmmoComp* ABaseShooter::AmmoComp = nullptr [protected]`

Networked health component.

5.1.3.2 BaseLookUpRate `float ABaseShooter::BaseLookUpRate`

In deg/sec. Other scaling may affect final turn rate.

5.1.3.3 BaseTurnRate `float ABaseShooter::BaseTurnRate`

5.1.3.4 bIsRagdoll `bool ABaseShooter::bIsRagdoll = false [protected]`

Interaction detection component, uses linetrace.

5.1.3.5 FirstPersonCameraComponent `UCameraComponent* ABaseShooter::FirstPersonCameraComponent = nullptr [protected]`

SkelMesh: Skeletal Gun mesh.

5.1.3.6 HealthComponent `UHealthComp* ABaseShooter::HealthComponent = nullptr [protected]`

First person camera.

5.1.3.7 LinetraceComp `UStickyLinetraceComp* ABaseShooter::LinetraceComp = nullptr [protected]`

Networked ammo component.

5.1.3.8 MeshPtr `USkeletalMeshComponent* ABaseShooter::MeshPtr = nullptr [protected]`

5.1.3.9 StickyGun `UStickyGunSkeletalComp* ABaseShooter::StickyGun = nullptr [protected]`

SkelMesh: 1st person view arms.

The documentation for this class was generated from the following files:

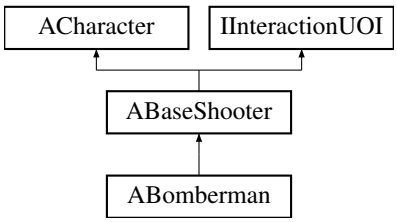
- Source/ario_stickyBomb_UE4/Public/Characters/BaseShooter.h
- Source/ario_stickyBomb_UE4/Private/Characters/BaseShooter.cpp

5.2 ABomberman Class Reference

Inherits from ACharacter & IInteractionUOI.

`#include <Bomberman.h>`

Inheritance diagram for ABomberman:



Public Member Functions

- ABomberman ()

  Construct a new ABomberman object.
- virtual void TryInteractItem () override

  Interact Item, Start.
- virtual void EndInteractItem () override

  Interact Item, End.
- UStickyGunSkeletalComp * GetStickyGun ()

  Get the Sticky Gun object.
- USkeletalMeshComponent * GetCharMesh ()

  Get the Char Mesh object.
- UHealthComp * GetHealthComp ()

  Get the Health Comp object.
- UAmmoComp * GetAmmoComp ()

  Get the Ammo Comp object.
- UCameraComponent * GetFirstPersonCameraComponent ()

  Get the First Person Camera Component object.
- void TryStartFire ()

  Try triggering weapon, checks ammo with server in subsequent calls.
- void ServerTriggerRagdoll ()

  Trigger ragdoll, can only be run by Server.
- void ServerUndoRagdoll ()

  Reset player/Undo Ragdoll, Can only be run by server.
- void TriggerPlayerStateAmmo (int LocalAmmoUpdate)

  Triggers an update to the players UAmmoWidget.

Public Attributes

- float BaseTurnRate
- float BaseLookUpRate

  In deg/sec. Other scaling may affect final turn rate.

Protected Member Functions

- virtual void BeginPlay ()

  Actor BeginPlay.
- virtual void SetupPlayerInputComponent (UInputComponent *InputComponent) override

  Override in dervied classes to set up class-specific inputs.
- virtual float TakeDamage (float DamageAmount, FDamageEvent const &DamageEvent, AController *EventInstigator, AActor *DamageCauser) final

  TakeDamage event.
- void MulticastUndoRagdoll ()

  In deg/sec. Other scaling may affect final rate.
- void MulticastTriggerRagdoll ()

  Trigger Ragdoll function, will run multicasted.
- void ServerTryInteractItem ()

  Try Interaction function, will only run on server.
- void ServerEndInteractItem ()

  End Interaction function, will only run on server.
- void ServerTakeDamage (AActor *ThisActor, float DamageAmount, const UDamageType *DamageType, AController *EventInstigator, AActor *DamageCauser)

  TakeDamage function, will only run on server.
- void InitSkeletalBody ()

  Initializes the Skeletal mesh component and it's mesh.
- void InitCamera ()

  Initializes the Camera component.
- void InitActorComponents ()

  Initialize various actor components which aren't initialized in a function of their own.
- void SetupStickyGun ()

  Construct a AStickyGunSkeletalComp and call some intializing functions on it.
- void SetupCollision ()

  Sets up collision channels and collision settings.
- void MoveRight (float Val)
- void MoveForward (float Val)
- void TurnAtRate (float Rate)

  Turn Rate.
- void LookUpAtRate (float Rate)

  LookUp Rate.
- void CreateNewAxisMapping (FName DesiredAxisName, FKey DesiredAxisKey)

  Create and store New Axis Mapping.
- void CreateNewActionMapping (FName DesiredActionName, FKey DesiredActionKey)

  Create and store New Action Mapping.

Protected Attributes

- USkeletalMeshComponent * MeshPtr = nullptr
- UStickyGunSkeletalComp * StickyGun = nullptr

  SkelMesh: 1st person view arms.
- UCameraComponent * FirstPersonCameraComponent = nullptr

  SkelMesh: Skeletal Gun mesh.
- UHealthComp * HealthComponent = nullptr

  First person camera.
- UAmmoComp * AmmoComp = nullptr

Networked health component.

- UStickyLinetraceComp * LinetraceComp = nullptr

  Networked ammo component.

- bool bIsRagdoll = false

  Interaction detection component, uses linetrace.

5.2.1 Detailed Description

Inherits from ACharacter & IInteractionUOI.

Inherits from ABaseShooter.

Derived from ACharacter and implements a StickyGunSkeletalComponent, HealthComponent and a Ammo↩ Component

Todo Currently this class is fairly bloated.

1. Move some of the functions of ABaseShooter into Bomberman, or rather rewrite some as virtual and implement them in child classes such as ABomberMan.
2. Move methods such as those regarding to input setup to the player controller class

Currently it only sets up the key_bindings

Todo Move some of the functions of ABaseShooter into Bomberman, or rather rewrite some as virtual and implement them in child classes such as ABomberMan

Derived from ACharacter and implements a StickyGunSkeletalComponent, HealthComponent and a Ammo↩ Component

Todo Currently this class is fairly bloated.

1. Move some of the functions of ABaseShooter into Bomberman, or rather rewrite some as virtual and implement them in child classes such as ABomberMan.
2. Move methods such as those regarding to input setup to the player controller class

Todo Move some of the functions of ABaseShooter into Bomberman, or rather rewrite some as virtual and implement them in child classes such as ABomberMan

5.2.2 Constructor & Destructor Documentation

5.2.2.1 ABomberman() `ABomberman::ABomberman ( )`

Construct a new ABomberman object.

5.2.3 Member Function Documentation

5.2.3.1 BeginPlay() `void ABomberman::BeginPlay ( ) [protected], [virtual]`

Actor BeginPlay.

Runs when world at BeginPlay, or when constructed in an already running world.

Reimplemented from ABaseShooter.

5.2.3.2 CreateNewActionMapping() `void ABaseShooter::CreateNewActionMapping ( FName DesiredActionName, FKey DesiredActionKey ) [protected], [inherited]`

Create and store New Action Mapping.

Parameters

DesiredActionName	
DesiredActionKey	

Todo Move to a controller class perhaps?

5.2.3.3 CreateNewAxisMapping() `void ABaseShooter::CreateNewAxisMapping ( FName DesiredAxisName, FKey DesiredAxisKey ) [protected], [inherited]`

Create and store New Axis Mapping.

Parameters

DesiredAxisName	
DesiredAxisKey	

Todo Move to a controller class perhaps?

5.2.3.4 EndInteractItem() `void ABaseShooter::EndInteractItem ( ) [override], [virtual], [inherited]`

Interact Item, End.

Implements IInteractionUOI.

5.2.3.5 GetAmmoComp() UAmmoComp * ABaseShooter::GetAmmoComp ( ) [inherited]

Get the Ammo Comp object.

Returns

UAmmoComp*

5.2.3.6 GetCharMesh() USkeletalMeshComponent * ABaseShooter::GetCharMesh ( ) [inherited]

Get the Char Mesh object.

Returns

USkeletalMeshComponent*

5.2.3.7 GetFirstPersonCameraComponent() UCameraComponent * ABaseShooter::GetFirstPerson↩ CameraComponent ( ) [inherited]

Get the First Person Camera Component object.

Returns

UCameraComponent*

5.2.3.8 GetHealthComp() UHealthComp * ABaseShooter::GetHealthComp ( ) [inherited]

Get the Health Comp object.

Returns

UHealthComp*

5.2.3.9 GetStickyGun() UStickyGunSkeletalComp * ABaseShooter::GetStickyGun ( ) [inherited]

Get the Sticky Gun object.

Returns

UStickyGunSkeletalComp*

5.2.3.10 InitActorComponents() `void ABaseShooter::InitActorComponents ( ) [protected], [inherited]`

Initialize various actor components which aren't initialized in a function of their own.

5.2.3.11 InitCamera() `void ABaseShooter::InitCamera ( ) [protected], [inherited]`

Initializes the Camera component.

5.2.3.12 InitSkeletalBody() `void ABaseShooter::InitSkeletalBody ( ) [protected], [inherited]`

Initializes the Skeletal mesh component and it's mesh.

5.2.3.13 LookUpAtRate() `void ABaseShooter::LookUpAtRate ( float Rate ) [protected], [inherited]`

LookUp Rate.

Called via input to turn look up/down at a given rate.

Parameters

Rate	This is a normalized rate, i.e. 1.0 means 100% of desired turn rate

5.2.3.14 MoveForward() `void ABaseShooter::MoveForward ( float Val ) [protected], [inherited]`

5.2.3.15 MoveRight() `void ABaseShooter::MoveRight ( float Val ) [protected], [inherited]`

5.2.3.16 MulticastTriggerRagdoll() `void ABaseShooter::MulticastTriggerRagdoll ( ) [protected], [inherited]`

Trigger Ragdoll function, will run multicasted.

5.2.3.17 MulticastUndoRagdoll() `void ABaseShooter::MulticastUndoRagdoll ( ) [protected], [inherited]`

In deg/sec. Other scaling may affect final rate.

Undo Ragdoll function, will run multicasted

5.2.3.18 ServerEndInteractItem() `void ABaseShooter::ServerEndInteractItem ( ) [protected], [inherited]`

End Interaction function, will only run on server.

5.2.3.19 ServerTakeDamage() `void ABaseShooter::ServerTakeDamage (`

```
AActor * ThisActor,
float DamageAmount,
const UDamageType * DamageType,
AController * EventInstigator,
AActor * DamageCauser ) [protected], [inherited]
```

TakeDamage function, will only run on server.

Parameters

ThisActor	
DamageAmount	
DamageType	
EventInstigator	
DamageCauser	

5.2.3.20 ServerTriggerRagdoll() `void ABaseShooter::ServerTriggerRagdoll ( ) [inherited]`

Trigger ragdoll, can only be run by Server.

5.2.3.21 ServerTryInteractItem() `void ABaseShooter::ServerTryInteractItem ( ) [protected], [inherited]`

Try Interaction function, will only run on server.

5.2.3.22 ServerUndoRagdoll() `void ABaseShooter::ServerUndoRagdoll ( ) [inherited]`

Reset player/Undo Ragdoll, Can only be run by server.

5.2.3.23 SetupCollision() `void ABaseShooter::SetupCollision ( ) [protected], [inherited]`

Sets up collision channels and collision settings.

5.2.3.24 SetupPlayerInputComponent() `void ABomberman::SetupPlayerInputComponent ( UInputComponent * InputComponent ) [override], [protected], [virtual]`

Override in dervied classes to set up class-specific inputs.

Parameters

InputComponent	

Reimplemented from ABaseShooter.

5.2.3.25 SetupStickyGun() `void ABaseShooter::SetupStickyGun ( ) [protected], [inherited]`

Construct a AStickyGunSkeletalComp and call some intializing functions on it.

5.2.3.26 TakeDamage() `float ABaseShooter::TakeDamage ( float DamageAmount, FDamageEvent const & DamageEvent, AController * EventInstigator, AActor * DamageCauser ) [final], [protected], [virtual], [inherited]`

TakeDamage event.

Parameters

DamageAmount	
DamageEvent	
EventInstigator	
DamageCauser	

Returns

float

5.2.3.27 TriggerPlayerStateAmmo() `void ABaseShooter::TriggerPlayerStateAmmo ( int LocalAmmoUpdate ) [inherited]`

Triggers an update to the players UAmmoWidget.

Parameters

LocalAmmoUpdate	

5.2.3.28 TryInteractItem() `void ABaseShooter::TryInteractItem ( ) [override], [virtual], [inherited]`

Interact Item, Start.

Implements IInteractionUOI.

5.2.3.29 TryStartFire() `void ABaseShooter::TryStartFire ( ) [inherited]`

Try triggering weapon, checks ammo with server in subsequent calls.

5.2.3.30 TurnAtRate() `void ABaseShooter::TurnAtRate ( float Rate ) [protected], [inherited]`

Turn Rate.

Called via input to turn at a given rate.

Parameters

Rate	This is a normalized rate, i.e. 1.0 means 100% of desired turn rate

5.2.4 Member Data Documentation

5.2.4.1 AmmoComp `UAmmoComp* ABaseShooter::AmmoComp = nullptr [protected], [inherited]`

Networked health component.

5.2.4.2 BaseLookUpRate `float ABaseShooter::BaseLookUpRate [inherited]`

In deg/sec. Other scaling may affect final turn rate.

5.2.4.3 BaseTurnRate `float ABaseShooter::BaseTurnRate [inherited]`

5.2.4.4 bIsRagdoll `bool ABaseShooter::bIsRagdoll = false [protected], [inherited]`

Interaction detection component, uses linetrace.

5.2.4.5 FirstPersonCameraComponent `UCameraComponent* ABaseShooter::FirstPersonCameraComponent = nullptr [protected], [inherited]`

SkelMesh: Skeletal Gun mesh.

5.2.4.6 HealthComponent `UHealthComp* ABaseShooter::HealthComponent = nullptr [protected], [inherited]`

First person camera.

5.2.4.7 LinetraceComp `UStickyLinetraceComp* ABaseShooter::LinetraceComp = nullptr [protected], [inherited]`

Networked ammo component.

5.2.4.8 MeshPtr `USkeletalMeshComponent* ABaseShooter::MeshPtr = nullptr [protected], [inherited]`

5.2.4.9 StickyGun `UStickyGunSkeletalComp* ABaseShooter::StickyGun = nullptr [protected], [inherited]`

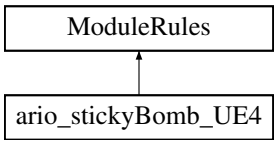
SkelMesh: 1st person view arms.

The documentation for this class was generated from the following files:

- Source/ario_stickyBomb_UE4/Public/Characters/Bomberman.h
- Source/ario_stickyBomb_UE4/Private/Characters/Bomberman.cpp

5.3 ario_stickyBomb_UE4 Class Reference

Inheritance diagram for ario_stickyBomb_UE4:



Public Member Functions

- ario_stickyBomb_UE4 (ReadOnlyTargetRules Target)

5.3.1 Constructor & Destructor Documentation

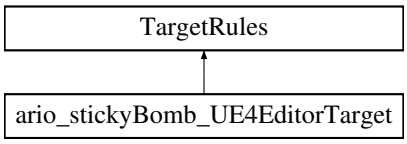
5.3.1.1 ario_stickyBomb_UE4() `ario_stickyBomb_UE4.ario_stickyBomb_UE4 ( ReadOnlyTargetRules Target ) [inline]`

The documentation for this class was generated from the following file:

- Source/ario_stickyBomb_UE4/ario_stickyBomb_UE4.Build.cs

5.4 ario_stickyBomb_UE4EditorTarget Class Reference

Inheritance diagram for ario_stickyBomb_UE4EditorTarget:



Public Member Functions

- ario_stickyBomb_UE4EditorTarget (TargetInfo Target)

5.4.1 Constructor & Destructor Documentation

5.4.1.1 ario_stickyBomb_UE4EditorTarget() `ario_stickyBomb_UE4EditorTarget.ario_stickyBomb_UE4EditorTarget (`

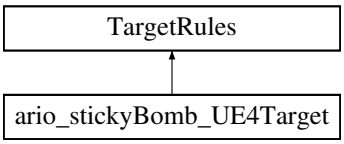
`TargetInfo *Target* ) [inline]`

The documentation for this class was generated from the following file:

- Source/ario_stickyBomb_UE4Editor.Target.cs

5.5 ario_stickyBomb_UE4Target Class Reference

Inheritance diagram for ario_stickyBomb_UE4Target:



Public Member Functions

- ario_stickyBomb_UE4Target (TargetInfo Target)

5.5.1 Constructor & Destructor Documentation

5.5.1.1 ario_stickyBomb_UE4Target() `ario_stickyBomb_UE4Target.ario_stickyBomb_UE4Target (`

`TargetInfo *Target* ) [inline]`

The documentation for this class was generated from the following file:

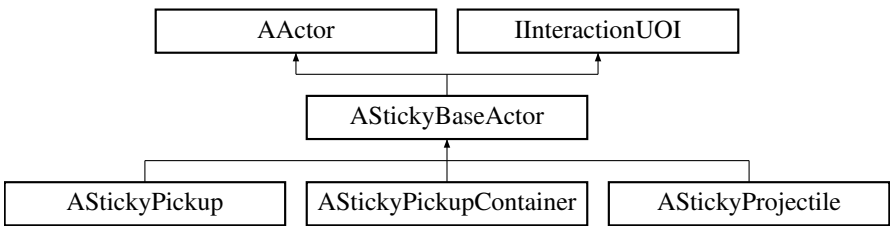
- Source/ario_stickyBomb_UE4.Target.cs

5.6 AStickyBaseActor Class Reference

A simple shell of a class to be expanded upon.

`#include <StickyBaseActor.h>`

Inheritance diagram for AStickyBaseActor:



Public Member Functions

- AStickyBaseActor ()

  Construct a new AStickyBaseActor object.
- virtual void TryInteractItem () override

  Interact Item, End.
- virtual void EndInteractItem () override

  Interact Item, Start.

5.6.1 Detailed Description

A simple shell of a class to be expanded upon.

A simple shell of a class implementation to be expanded upon.

A simple utility class for generating and editing materialgraphs in code.

Todo Pehaps the class needs to be extended to be be able to edit other editor uassets

5.6.2 Constructor & Destructor Documentation

5.6.2.1 AStickyBaseActor() `AStickyBaseActor::AStickyBaseActor ( )`

Construct a new AStickyBaseActor object.

5.6.3 Member Function Documentation

5.6.3.1 EndInteractItem() `void AStickyBaseActor::EndInteractItem ( ) [override], [virtual]`

Interact Item, Start.

Implements IInteractionUOI.

Reimplemented in AStickyPickupContainer, AStickyPickup, and AStickyProjectile.

5.6.3.2 TryInteractItem() `void AStickyBaseActor::TryInteractItem ( ) [override], [virtual]`

Interact Item, End.

Implements IInteractionUOI.

Reimplemented in AStickyPickupContainer, AStickyPickup, and AStickyProjectile.

The documentation for this class was generated from the following files:

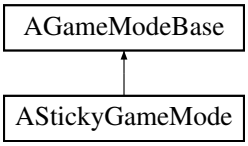
- Source/ario_stickyBomb_UE4/Public/Actors/StickyBaseActor.h
- Source/ario_stickyBomb_UE4/Private/Actors/StickyBaseActor.cpp

5.7 AStickyGameMode Class Reference

Naive implementation of Gamemode for networked play.

`#include <StickyGameMode.h>`

Inheritance diagram for AStickyGameMode:



Public Member Functions

- AStickyGameMode ()
  Construct a new AStickyGameMode object.
- virtual void StartPlay () override
  Plays when gamemode starts.
- virtual void BeginPlay () override
  Unchanged, only calls Super::BeginPlay.
- virtual void Tick (float DeltaSeconds) override
  This gamemodes tick. uses it to keep track of the players and the game.
- virtual void PostLogin (APlayerController ∗NewPlayer) override
  Called when user logs-in.
- virtual void Logout (AController ∗ExitingPlayer) override
  Called when user is exiting the game.
- ABaseShooter ∗ FindPlayer (int32 LocalPlayerId)
  Find player on server given player ID sent by client.

Public Attributes

- FOnActorKilled OnActorKilled
  OnActorKilled delegate, multicasts when a player has been killed.
- FOnTargetHitEvent OnHitEvent
  OnHitEvent Delegate, multicasts when a hit has occured.

Protected Member Functions

- void RegisterNewPlayer (ABaseShooter ∗NewPlayer)
  Register newly logged ing player.
- void DeregisterExitingPlayer (ABaseShooter ∗ExitingPlayer)
  Unregister player who is logging out.
- void CheckAnyPlayerAlive ()
  Is any player alive?
- bool HasPlayerWon () const
  Checks whether match is still going of if has been won.
- void RestartDeadPlayers ()
  Restart any dead players who are still connected to the server.
- void PlayerKilled (AActor ∗DamageCauser, AActor ∗DamagedActor, AController ∗InstigatorController)
  Registers a killed player.
- void GameOver ()
  Ends the current game.

Protected Attributes

- bool bIsGameOver
- bool bHasGameStarted
- bool bTriggerOnce = false
- FMaterialGenerator * MatGen

5.7.1 Detailed Description

Naive implementation of Gamemode for networked play.

Simple GameMode which has been designed to work with ABaseShooter derived classes.

Implements barebone player checks which could be expanded on.

5.7.2 Constructor & Destructor Documentation

5.7.2.1 AStickyGameMode() `AStickyGameMode::AStickyGameMode ( )`

Construct a new AStickyGameMode object.

5.7.3 Member Function Documentation

5.7.3.1 BeginPlay() `void AStickyGameMode::BeginPlay ( ) [override], [virtual]`

Unchanged, only calls Super::BeginPlay.

5.7.3.2 CheckAnyPlayerAlive() `void AStickyGameMode::CheckAnyPlayerAlive ( ) [protected]`

Is any player alive?

Todo remove, superflous function

5.7.3.3 DeregisterExitingPlayer() `void AStickyGameMode::DeregisterExitingPlayer ( ABaseShooter * ExitingPlayer ) [protected]`

Unregister player who is logging out.

Parameters

ExitingPlayer	

5.7.3.4 FindPlayer() `ABaseShooter * AStickyGameMode::FindPlayer ( int32 LocalPlayerId )`

Find player on server given player ID sent by client.

Parameters

Local↩ PlayerId	

Returns

ABaseShooter∗

5.7.3.5 GameOver() `void AStickyGameMode::GameOver ( ) [protected]`

Ends the current game.

5.7.3.6 HasPlayerWon() `bool AStickyGameMode::HasPlayerWon ( ) const [protected]`

Checks whether match is still going of if has been won.

Returns

true | false

5.7.3.7 Logout() `void AStickyGameMode::Logout ( AController * ExitingPlayer ) [override], [virtual]`

Called when user is exiting the game.

Parameters

ExitingPlayer	

5.7.3.8 PlayerKilled()

```
void AStickyGameMode::PlayerKilled (
        AActor * DamageCauser,
        AActor * DamagedActor,
        AController * InstigatorController )  [protected]
```

Registers a killed player.

Does not do much right now, but could be expanded upon to trigger OnKill audio cues or something similar

Parameters

DamageCauser	
DamagedActor	
InstigatorController	

5.7.3.9 PostLogin()

```
void AStickyGameMode::PostLogin (
        APlayerController * NewPlayer )  [override], [virtual]
```

Called when user logs-in.

Todo Register user-chosen names?

Parameters

NewPlayer	

5.7.3.10 RegisterNewPlayer()

```
void AStickyGameMode::RegisterNewPlayer (
        ABaseShooter * NewPlayer )  [protected]
```

Register newly logged ing player.

Parameters

NewPlayer	

5.7.3.11 RestartDeadPlayers()

```
void AStickyGameMode::RestartDeadPlayers ( )  [protected]
```

Restart any dead players who are still connected to the server.

Todo Reset player capsule orientation and reset stance?

5.7.3.12 StartPlay() `void AStickyGameMode::StartPlay ( ) [override], [virtual]`

Plays when gamemode starts.

Inherited Methods: Overrides

Generates a material using FMaterialGenerator and turns off timeline logs

Inherited Methods: Overrides

5.7.3.13 Tick() `void AStickyGameMode::Tick ( float DeltaSeconds ) [override], [virtual]`

This gamemodes tick. uses it to keep track of the players and the game.

Parameters

DeltaSeconds	

Todo Have a timer based delay before resterting Dead players

5.7.4 Member Data Documentation

5.7.4.1 bHasGameStarted `bool AStickyGameMode::bHasGameStarted [protected]`

5.7.4.2 bIsGameOver `bool AStickyGameMode::bIsGameOver [protected]`

5.7.4.3 bTriggerOnce `bool AStickyGameMode::bTriggerOnce = false [protected]`

5.7.4.4 MatGen `FMaterialGenerator* AStickyGameMode::MatGen [protected]`

5.7.4.5 OnActorKilled `FOnActorKilled AStickyGameMode::OnActorKilled`

OnActorKilled delegate, multicasts when a player has been killed.

5.7.4.6 OnHitEvent `FOnTargetHitEvent AStickyGameMode::OnHitEvent`

OnHitEvent Delegate, multicasts when a hit has occured.

The documentation for this class was generated from the following files:

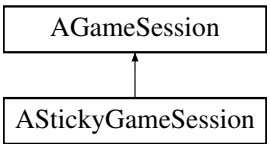
- Source/ario_stickyBomb_UE4/Public/StickyGameMode.h
- Source/ario_stickyBomb_UE4/Private/StickyGameMode.cpp

5.8 AStickyGameSession Class Reference

Dummy gamesession class.

`#include <StickyGameSession.h>`

Inheritance diagram for AStickyGameSession:



5.8.1 Detailed Description

Dummy gamesession class.

The documentation for this class was generated from the following file:

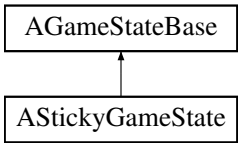
- Source/ario_stickyBomb_UE4/Public/StickyGameSession.h

5.9 AStickyGameState Class Reference

A simple shell of a class implementation to be expanded upon.

`#include <StickyGameState.h>`

Inheritance diagram for AStickyGameState:



5.9.1 Detailed Description

A simple shell of a class implementation to be expanded upon.

Dummy Gamestate, derives from AGameStateBase.

Todo Add gamestate logic

The documentation for this class was generated from the following file:

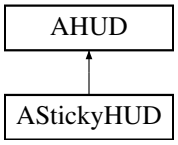
- Source/ario_stickyBomb_UE4/Public/StickyGameState.h

5.10 AStickyHUD Class Reference

A simple HUD class to handle widgets.

`#include <StickyHUD.h>`

Inheritance diagram for AStickyHUD:



Public Member Functions

- AStickyHUD ()
  Construct a new AStickyHUD object.
- virtual void Tick (float DeltaTime) final
  Inherited and mostly unused - Super::Tick.
- virtual void DrawHUD () final
  Inherited and mostly unused - Super::DrawHUD.
- void UpdateTotalKills ()
  Update Killbox/container with total kills.
- void UpdateAmmo ()
  Reflect value to textbox.
- void ToggleGameMenu ()
  Toggle (unimplemented) game menu.
- void AddKillToWidget (FString Kill)
  Update widget with Kill (unfinished/untested)

Protected Member Functions

- virtual void PostInitializeComponents () override

  Inherited and unused PostInit.
- virtual void BeginPlay () final

  Initialize some components on BeginPlay.
- void InitializeKillOverlayWidget ()

  Init Kill Overlay widget.
- void InitializeTotalKillsWidget ()

  Init Killbox/container widget.
- void InitializeAmmoWidget ()

  Init AmmoWidget.

Protected Attributes

- TSharedPtr< SKillWidget > TotalKillsWidget
- TSharedPtr< SAmmoWidget > AmmoWidget
- TSharedPtr< SSlideInText > OverlayMenu
- TSharedPtr< SKillContentContainer > KillList
- TSharedPtr< AStickyHUD > StickyHUD
- bool bIsTitleVisible
- bool bIsKillWidgetInitialized
- bool bIsAmmoWidgetInitialized
- bool bIsOverlayMenuVisible

5.10.1 Detailed Description

A simple HUD class to handle widgets.

Main HUD class for ABaseShooter, derives from AHUD.

Simple HUD class to load the widgets in UI/Widgets/ and set up display

5.10.2 Constructor & Destructor Documentation

5.10.2.1 AStickyHUD() `AStickyHUD::AStickyHUD ( )`

Construct a new AStickyHUD object.

5.10.3 Member Function Documentation

5.10.3.1 AddKillToWidget() `void AStickyHUD::AddKillToWidget ( FString Kill )`

Update widget with Kill (unfinished/untested)

Parameters

Kill	

5.10.3.2 BeginPlay() `void AStickyHUD::BeginPlay ( ) [final], [protected], [virtual]`

Initialize some components on BeginPlay.

5.10.3.3 DrawHUD() `void AStickyHUD::DrawHUD ( ) [final], [virtual]`

Inherited and mostly unused - Super::DrawHUD.

5.10.3.4 InitializeAmmoWidget() `void AStickyHUD::InitializeAmmoWidget ( ) [protected]`

Init AmmoWidget.

5.10.3.5 InitializeKillOverlayWidget() `void AStickyHUD::InitializeKillOverlayWidget ( ) [protected]`

Init Kill Overlay widget.

5.10.3.6 InitializeTotalKillsWidget() `void AStickyHUD::InitializeTotalKillsWidget ( ) [protected]`

Init Killbox/container widget.

5.10.3.7 PostInitializeComponents() `void AStickyHUD::PostInitializeComponents ( ) [override], [protected], [virtual]`

Inherited and unused PostInit.

Inherited Methods: Overrides

5.10.3.8 Tick() `void AStickyHUD::Tick ( float DeltaTime ) [final], [virtual]`

Inherited and mostly unused - Super::Tick.

Parameters

DeltaTime	

5.10.3.9 ToggleGameMenu() `void AStickyHUD::ToggleGameMenu ( )`

Toggle (unimplemented) game menu.

5.10.3.10 UpdateAmmo() `void AStickyHUD::UpdateAmmo ( )`

Reflect value to textbox.

Reflects value from ABaseShooter->StickyGun->AmmoComp->GetAmmoCount(),e

5.10.3.11 UpdateTotalKills() `void AStickyHUD::UpdateTotalKills ( )`

Update Killbox/container with total kills.

5.10.4 Member Data Documentation

5.10.4.1 AmmoWidget `TSharedPtr<SAmmoWidget> AStickyHUD::AmmoWidget [protected]`

5.10.4.2 bIsAmmoWidgetInitialized `bool AStickyHUD::bIsAmmoWidgetInitialized [protected]`

5.10.4.3 bIsKillWidgetInitialized `bool AStickyHUD::bIsKillWidgetInitialized [protected]`

5.10.4.4 bIsOverlayMenuVisible `bool AStickyHUD::bIsOverlayMenuVisible [protected]`

5.10.4.5 bIsTitleVisible `bool AStickyHUD::bIsTitleVisible [protected]`

5.10.4.6 KillList TSharedPtr<SKillContentContainer> AStickyHUD::KillList [protected]

5.10.4.7 OverlayMenu TSharedPtr<SSlideInText> AStickyHUD::OverlayMenu [protected]

5.10.4.8 StickyHUD TSharedPtr<AStickyHUD> AStickyHUD::StickyHUD [protected]

5.10.4.9 TotalKillsWidget TSharedPtr<SKillWidget> AStickyHUD::TotalKillsWidget [protected]

The documentation for this class was generated from the following files:

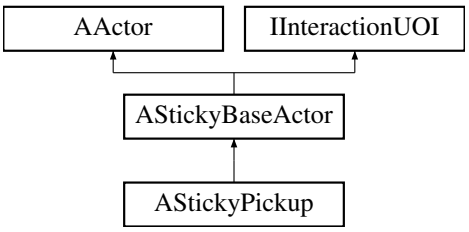
- Source/ario_stickyBomb_UE4/Public/UI/StickyHUD.h
- Source/ario_stickyBomb_UE4/Private/UI/StickyHUD.cpp

5.11 AStickyPickup Class Reference

A simple pickup actor.

`#include <StickyPickup.h>`

Inheritance diagram for AStickyPickup:



Public Member Functions

- AStickyPickup ()

  Construct a new AStickyPickup object.
- virtual void EndInteractItem () override

  Interact Item, End.
- virtual void TryInteractItem () override

  Interact Item, Start.
- bool DidPickup (AActor *OtherActor)

  Tries picking up actor, returns to reflect if it was successful.
- USphereComponent * GetCollisionComp () const

  Get the Collision Comp object.

Protected Member Functions

- virtual void Tick (float DeltaTime) final
- virtual void BeginPlay () final

Protected Attributes

- USphereComponent * CollisionComp = nullptr
- UStaticMeshComponent * MeshComponentPtr = nullptr

5.11.1 Detailed Description

A simple pickup actor.

Has rudimentary pickup logic which sends it to the players ammo component

5.11.2 Constructor & Destructor Documentation

5.11.2.1 AStickyPickup() `AStickyPickup::AStickyPickup ( )`

Construct a new AStickyPickup object.

5.11.3 Member Function Documentation

5.11.3.1 BeginPlay() `void AStickyPickup::BeginPlay ( ) [final], [protected], [virtual]`

5.11.3.2 DidPickup() `bool AStickyPickup::DidPickup ( AActor * OtherActor )`

Tries picking up actor, returns to reflect if it was successful.

Parameters

OtherActor	

Returns

true | false

5.11.3.3 EndInteractItem() `void AStickyPickup::EndInteractItem ( ) [override], [virtual]`

Interact Item, End.

Reimplemented from AStickyBaseActor.

5.11.3.4 GetCollisionComp() `USphereComponent * AStickyPickup::GetCollisionComp ( ) const`

Get the Collision Comp object.

Returns

USphereComponent*

5.11.3.5 Tick() `void AStickyPickup::Tick ( float DeltaTime ) [final], [protected], [virtual]`

Inherited Methods: Overrides

5.11.3.6 TryInteractItem() `void AStickyPickup::TryInteractItem ( ) [override], [virtual]`

Interact Item, Start.

Reimplemented from AStickyBaseActor.

5.11.4 Member Data Documentation

5.11.4.1 CollisionComp `USphereComponent* AStickyPickup::CollisionComp = nullptr [protected]`

5.11.4.2 MeshComponentPtr `UStaticMeshComponent* AStickyPickup::MeshComponentPtr = nullptr [protected]`

The documentation for this class was generated from the following files:

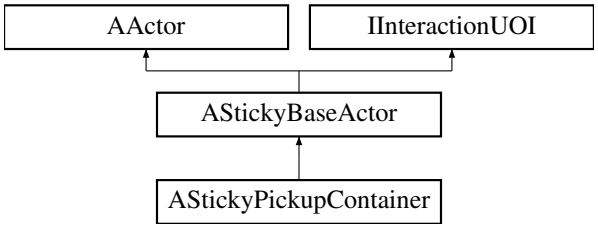
- Source/ario_stickyBomb_UE4/Public/Actors/StickyPickup.h
- Source/ario_stickyBomb_UE4/Private/Actors/StickyPickup.cpp

5.12 AStickyPickupContainer Class Reference

A simple actor which holds a static mesh container.

`#include <StickyPickupContainer.h>`

Inheritance diagram for AStickyPickupContainer:



Public Member Functions

- AStickyPickupContainer ()

  Construct a new AStickyPickupContainer object.
- virtual void EndInteractItem () override

  Interact Item, End.
- virtual void TryInteractItem () override

  Interact Item, Start.

Protected Member Functions

- virtual void Tick (float DeltaTime) final
- virtual void BeginPlay () final

Protected Attributes

- TArray< UStaticMeshComponent * > MeshComponentArray

5.12.1 Detailed Description

A simple actor which holds a static mesh container.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 AStickyPickupContainer() `AStickyPickupContainer::AStickyPickupContainer ( )`

Construct a new AStickyPickupContainer object.

5.12.3 Member Function Documentation

5.12.3.1 BeginPlay() `void AStickyPickupContainer::BeginPlay ( ) [final], [protected], [virtual]`

5.12.3.2 EndInteractItem() `void AStickyPickupContainer::EndInteractItem ( ) [override], [virtual]`

Interact Item, End.

Reimplemented from AStickyBaseActor.

5.12.3.3 Tick() `void AStickyPickupContainer::Tick ( float DeltaTime ) [final], [protected], [virtual]`

Inherited Methods: Overrides

5.12.3.4 TryInteractItem() `void AStickyPickupContainer::TryInteractItem ( ) [override], [virtual]`

Interact Item, Start.

Reimplemented from AStickyBaseActor.

5.12.4 Member Data Documentation

5.12.4.1 MeshComponentArray `TArray<UStaticMeshComponent*> AStickyPickupContainer::MeshComponentArray [protected]`

The documentation for this class was generated from the following files:

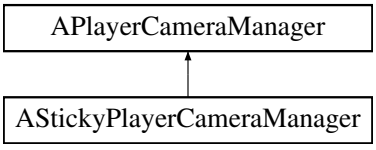
- Source/ario_stickyBomb_UE4/Public/Actors/StickyPickupContainer.h
- Source/ario_stickyBomb_UE4/Private/Actors/StickyPickupContainer.cpp

5.13 AStickyPlayerCameraManager Class Reference

A simple shell of a class implementation to be expanded upon.

`#include <StickyPlayerCameraManager.h>`

Inheritance diagram for AStickyPlayerCameraManager:



Public Member Functions

- AStickyPlayerCameraManager ()

  Construct a new AStickyPlayerCameraManager object.

5.13.1 Detailed Description

A simple shell of a class implementation to be expanded upon.

Dummy Camera Manager, derives from APlayerCameraManager.

Only used to force 'bShouldSendClientSideCameraUpdate' to be true, as the default camera in the FPS example is not controlled by a Camera Manager, and thus doesn't have builtin support for replicating camera pitch.

Only used to set pitch to replicate thought the camera manager, instead of forcing replication of pitch on the tick of the player

5.13.2 Constructor & Destructor Documentation

5.13.2.1 AStickyPlayerCameraManager() `AStickyPlayerCameraManager::AStickyPlayerCameraManager ( )`

Construct a new AStickyPlayerCameraManager object.

The documentation for this class was generated from the following files:

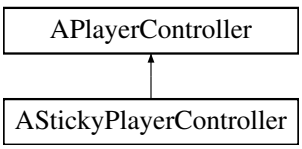
- Source/ario_stickyBomb_UE4/Public/StickyPlayerCameraManager.h
- Source/ario_stickyBomb_UE4/Private/StickyPlayerCameraManager.cpp

5.14 AStickyPlayerController Class Reference

A simple shell of a class implementation to be expanded upon.

`#include <StickyPlayerController.h>`

Inheritance diagram for AStickyPlayerController:



Public Member Functions

- AStickyPlayerController ()

  Construct a new AStickyPlayerController object.

5.14.1 Detailed Description

A simple shell of a class implementation to be expanded upon.

Dummy Player Controller, derives from APlayerController.

Todo Move controller and input related code from the ABaseShooter class into here.

This class is currently only used to explicitly set some default values

Todo Move controller and input related code from ABaseShooter into the player Controller

5.14.2 Constructor & Destructor Documentation

5.14.2.1 AStickyPlayerController() `AStickyPlayerController::AStickyPlayerController ( )`

Construct a new AStickyPlayerController object.

The documentation for this class was generated from the following files:

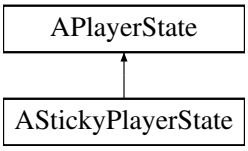
- Source/ario_stickyBomb_UE4/Public/StickyPlayerController.h
- Source/ario_stickyBomb_UE4/Private/StickyPlayerController.cpp

5.15 AStickyPlayerState Class Reference

Networked Player-state implementation.

`#include <StickyPlayerState.h>`

Inheritance diagram for AStickyPlayerState:



Public Member Functions

- void AddUserScore (float DeltaScore)
  Add new Score Delta.
- float GetUserScore () const
  Get the User Score value.
- void AddKill (int32 KillCount)
  Add new kill count Delta.
- int32 GetKills () const
  Get the Kills value.
- void SetAmmo (int32 AmmoCount)
  Set the Ammo count.
- int32 GetAmmo () const
  Get the Ammo count.

Protected Member Functions

- void OnRep_Kills ()

  On replication, Kills.
- void OnRep_Ammo ()

  OnReplication, Ammo.

Protected Attributes

- int32 Kills
- int32 Ammo

5.15.1 Detailed Description

Networked Player-state implementation.

Networked Player-state definition, derives from AStickyPlayerState.

Keeps track of the state of the player, in regards to kills and ammo.

Todo Redesign and remove extraneous code

Keeps track of the state of the player, in regards to kills and ammo. Shares data to the server through the gamemode->gamestate

5.15.2 Member Function Documentation

5.15.2.1 AddKill() `void AStickyPlayerState::AddKill ( int32 KillCount )`

Add new kill count Delta.

Parameters

KillCount	

5.15.2.2 AddUserScore() `void AStickyPlayerState::AddUserScore ( float DeltaScore )`

Add new Score Delta.

Parameters

DeltaScore	

5.15.2.3 GetAmmo() `int32 AStickyPlayerState::GetAmmo ( ) const`

Get the Ammo count.

Returns

int32

5.15.2.4 GetKills() `int32 AStickyPlayerState::GetKills ( ) const`

Get the Kills value.

Returns

int32

5.15.2.5 GetUserScore() `float AStickyPlayerState::GetUserScore ( ) const`

Get the User Score value.

Returns

float

5.15.2.6 OnRep_Ammo() `void AStickyPlayerState::OnRep_Ammo ( ) [protected]`

OnReplication, Ammo.

5.15.2.7 OnRep_Kills() `void AStickyPlayerState::OnRep_Kills ( ) [protected]`

On replication, Kills.

5.15.2.8 SetAmmo() `void AStickyPlayerState::SetAmmo ( int32 AmmoCount )`

Set the Ammo count.

Parameters

AmmoCount	

5.15.3 Member Data Documentation

5.15.3.1 Ammo `int32 AStickyPlayerState::Ammo [protected]`

5.15.3.2 Kills `int32 AStickyPlayerState::Kills [protected]`

The documentation for this class was generated from the following files:

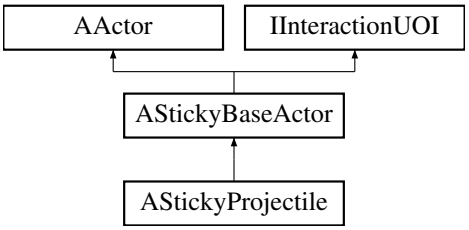
- Source/ario_stickyBomb_UE4/Public/StickyPlayerState.h
- Source/ario_stickyBomb_UE4/Private/StickyPlayerState.cpp

5.16 AStickyProjectile Class Reference

A projectile moving actor class.

`#include <StickyProjectile.h>`

Inheritance diagram for AStickyProjectile:



Public Member Functions

- AStickyProjectile ()

  Construct a new AStickyProjectile object.
- virtual void TryInteractItem () final

  Interact Item, End.
- virtual void EndInteractItem () final

  Interact Item, Start.
- float GetMaxLifetime () const

  Get the Max Lifetime.
- USphereComponent * GetCollisionComp () const

  Get the Collision Comp object.

- UProjectileMovementComponent * GetProjectileMovement () const

  Get the Projectile Movement object.
- UTimelineComponent * GetTimelineComp () const

  Get the Timeline Comp object.
- TArray< UActorComponent * > & GetReplicatedComponents ()

  Get the Replicated Components object.
- float GetDamageRadius () const

  Get the Damage Radius.
- float GetDamageAmount () const

  Get the Damage Amount.
- void SetDamageRadius (float InRadius)

  Set the Damage Radius.
- void SetDamageAmount (float InDamage)

  Set the Damage Amount.
- void SetMaxPossibleLifetime (float MaxLifetime)

  Set the Max Possible Lifetime length.
- void SetCurve (UCurveFloat *InCurve)

  Set the Curve object.
- UCurveFloat * GetCurve ()

  Get the Curve object.
- bool DidPickup (AActor *OtherActor)

  Tries picking up actor, returns to reflect if it was successful.

Protected Member Functions

- virtual void Tick (float DeltaTime) final
- virtual void BeginPlay () final
- virtual void LifeSpanExpired () final
- void OnHit (UPrimitiveComponent *HitComp, AActor *OtherActor, UPrimitiveComponent *OtherComp, FVector NormalImpulse, const FHitResult &Hit)

  Called when a hit triggers the collision of the actor.
- void OnExplode ()

  Decides what happens when the projectile explodes.
- void ModulateColor (const float InterpValue)

  Multicasted timeline callback.
- void TriggerExplosionFX ()

  Triggers explosion effects from the ue4 starter content.
- void ServerTriggerExplosionFX ()

  Server check.
- void MultiCastTriggerExplosionFX ()

  Multicast TriggerExplosionFX call to all clients.

Protected Attributes

- USphereComponent * CollisionComp = nullptr

  Used as a primitive collision component.
- UProjectileMovementComponent * ProjectileMovement = nullptr

  Handles projectile movement with an easy to use interface.
- UStaticMeshComponent * MeshComponentPtr = nullptr

  Mesh component of the projectile.

- UTimelineComponent * StickyTimelineComp = nullptr

  Timeline component to be used to modulate material color.
- UParticleSystemComponent * ParticleSystemComp = nullptr

  Particle Systems Component to spawn particle effects.
- UParticleSystem * ParticleFX = nullptr

  Particle System Type to actually hold the particle effect.
- UCurveFloat * StickyTimelineCurve = nullptr

  UCurve object to store the UCurve that is generated by StickyGunSkeletalComp.
- FOnTimelineEvent TimelineFinishedEvent

  Delegate signature for the function which will handle our Finished event.
- FOnTimelineFloat InterpTimelineEvent

  Delegate signature for the function which will handle our timeline loop.
- TEnumAsByte< ETimelineDirection::Type > TimelineDirection = ETimelineDirection::Type::Forward

5.16.1 Detailed Description

A projectile moving actor class.

A projectile actor which acts as a sticky bomb. Networked, but only naively implemented.

5.16.2 Constructor & Destructor Documentation

5.16.2.1 AStickyProjectile() `AStickyProjectile::AStickyProjectile ( )`

Construct a new AStickyProjectile object.

5.16.3 Member Function Documentation

5.16.3.1 BeginPlay() `void AStickyProjectile::BeginPlay ( ) [final], [protected], [virtual]`

5.16.3.2 DidPickup() `bool AStickyProjectile::DidPickup ( AActor * OtherActor )`

Tries picking up actor, returns to reflect if it was successful.

Parameters

OtherActor	

Returns

true | false

5.16.3.3 EndInteractItem() `void AStickyProjectile::EndInteractItem ( ) [final], [virtual]`

Interact Item, Start.

Reimplemented from AStickyBaseActor.

5.16.3.4 GetCollisionComp() `USphereComponent * AStickyProjectile::GetCollisionComp ( ) const`

Get the Collision Comp object.

Returns

USphereComponent*

5.16.3.5 GetCurve() `UCurveFloat * AStickyProjectile::GetCurve ( )`

Get the Curve object.

Returns

UCurveFloat*

5.16.3.6 GetDamageAmount() `float AStickyProjectile::GetDamageAmount ( ) const`

Get the Damage Amount.

Returns

float

5.16.3.7 GetDamageRadius() `float AStickyProjectile::GetDamageRadius ( ) const`

Get the Damage Radius.

Returns

float

5.16.3.8 GetMaxLifetime() `float AStickyProjectile::GetMaxLifetime ( ) const`

Get the Max Lifetime.

Returns

float

5.16.3.9 GetProjectileMovement() `UProjectileMovementComponent * AStickyProjectile::GetProjectile↩ Movement ( ) const`

Get the Projectile Movement object.

Returns

UProjectileMovementComponent*

5.16.3.10 GetReplicatedComponents() `TArray< UActorComponent * > & AStickyProjectile::Get↩ ReplicatedComponents ( )`

Get the Replicated Components object.

Returns

TArray<UActorComponent*>&

5.16.3.11 GetTimelineComp() `UTimelineComponent * AStickyProjectile::GetTimelineComp ( ) const`

Get the Timeline Comp object.

Returns

UTimelineComponent*

5.16.3.12 LifeSpanExpired() `void AStickyProjectile::LifeSpanExpired ( ) [final], [protected], [virtual]`

5.16.3.13 ModulateColor() `void AStickyProjectile::ModulateColor ( const float InterpValue ) [protected]`

Multicasted timeline callback.

Parameters

InterpValue	

5.16.3.14 MultiCastTriggerExplosionFX() `void AStickyProjectile::MultiCastTriggerExplosionFX ( ) [protected]`

Multicast TriggerExplosionFX call to all clients.

5.16.3.15 OnExplode() `void AStickyProjectile::OnExplode ( ) [protected]`

Decides what happens when the projectile explodes.

If server accepts, send radial damage and replicate it and then multicast VFX/SFX to clients

5.16.3.16 OnHit()

```
void AStickyProjectile::OnHit (
    UPrimitiveComponent * HitComp,
    AActor * OtherActor,
    UPrimitiveComponent * OtherComp,
    FVector NormalImpulse,
    const FHitResult & Hit ) [protected]
```

Called when a hit triggers the collision of the actor.

Parameters

HitComp	
OtherActor	
OtherComp	
NormalImpulse	
Hit	

5.16.3.17 ServerTriggerExplosionFX() `void AStickyProjectile::ServerTriggerExplosionFX ( ) [protected]`

Server check.

5.16.3.18 SetCurve()

```
void AStickyProjectile::SetCurve (
    UCurveFloat * InCurve )
```

Set the Curve object.

Parameters

InCurve	

5.16.3.19 SetDamageAmount() `void AStickyProjectile::SetDamageAmount ( float InDamage )`

Set the Damage Amount.

Parameters

InDamage	

5.16.3.20 SetDamageRadius() `void AStickyProjectile::SetDamageRadius ( float InRadius )`

Set the Damage Radius.

Parameters

InRadius	

5.16.3.21 SetMaxPossibleLifetime() `void AStickyProjectile::SetMaxPossibleLifetime ( float MaxLifetime )`

Set the Max Possible Lifetime length.

Parameters

MaxLifetime	

5.16.3.22 Tick() `void AStickyProjectile::Tick ( float DeltaTime ) [final], [protected], [virtual]`

Inherited Methods: Overrides

5.16.3.23 TriggerExplosionFX() `void AStickyProjectile::TriggerExplosionFX ( ) [protected]`

Triggers explosion effects from the ue4 starter content.

5.16.3.24 TryInteractItem() `void AStickyProjectile::TryInteractItem ( ) [final], [virtual]`

Interact Item, End.

Reimplemented from AStickyBaseActor.

5.16.4 Member Data Documentation

5.16.4.1 CollisionComp `USphereComponent* AStickyProjectile::CollisionComp = nullptr [protected]`

Used as a primitive collision component.

5.16.4.2 InterpTimelineEvent `FOnTimelineFloat AStickyProjectile::InterpTimelineEvent [protected]`

Delegate signature for the function which will handle our timeline loop.

5.16.4.3 MeshComponentPtr `UStaticMeshComponent* AStickyProjectile::MeshComponentPtr = nullptr [protected]`

Mesh component of the projectile.

5.16.4.4 ParticleFX `UParticleSystem* AStickyProjectile::ParticleFX = nullptr [protected]`

Particle System Type to actually hold the particle effect.

5.16.4.5 ParticleSystemComp `UParticleSystemComponent* AStickyProjectile::ParticleSystemComp = nullptr [protected]`

Particle Systems Component to spawn particle effects.

5.16.4.6 ProjectileMovement `UProjectileMovementComponent* AStickyProjectile::ProjectileMovement = nullptr [protected]`

Handles projectile movement with an easy to use interface.

5.16.4.7 StickyTimelineComp `UTimelineComponent* AStickyProjectile::StickyTimelineComp = nullptr [protected]`

Timeline component to be used to modulate material color.

5.16.4.8 StickyTimelineCurve `UCurveFloat* AStickyProjectile::StickyTimelineCurve = nullptr [protected]`

UCurve object to store the UCurve that is generated by StickyGunSkeletalComp.

5.16.4.9 TimelineDirection `TEnumAsByte<ETimelineDirection::Type> AStickyProjectile::TimelineDirection = ETimelineDirection::Type::Forward [protected]`

5.16.4.10 TimelineFinishedEvent `FOnTimelineEvent AStickyProjectile::TimelineFinishedEvent [protected]`

Delegate signature for the function which will handle our Finished event.

The documentation for this class was generated from the following files:

- Source/ario_stickyBomb_UE4/Public/Actors/StickyProjectile.h
- Source/ario_stickyBomb_UE4/Private/Actors/StickyProjectile.cpp

5.17 FMaterialGenerator Class Reference

A WIP utility class for Material creation/editing.

`#include <MaterialGenerator.h>`

Public Member Functions

- void CreateBasicMaterial (FString MaterialBaseName="M_Material", FString PackageName="/Game/GenMaterials/")

  Create a Basic Material object.
- void CreateCelShadedExplosionMat (FString MaterialBaseName=FString("M_CelExplosionMat"), FString PackageName=FString("/Game/GenMaterials/"))

  Create a Cel Shaded Explosion Mat object.

Static Public Member Functions

- static FMaterialGenerator * CreateObject ()

  Create a FMaterialGenerator object.

5.17.1 Detailed Description

A WIP utility class for Material creation/editing.

This class lets you create new packages, design them through code and save them to your content_browser (Currently only Materials). End result is a uasset primed for used in the editor, it can therefor be used to create templates.

The Actual idea with this class was to be able to create an explosion material during BeginPlay which would have been used in the actual explosion effects of the StickyProjectile.

Todo Extend with many variants of templates, both in regards to expression types but also package types, and opt for selective compilation with 'constexpr if's for efficiency.

5.17.2 Member Function Documentation

5.17.2.1 CreateBasicMaterial() `void FMaterialGenerator::CreateBasicMaterial (`

```
FString MaterialBaseName = "M_Material",
FString PackageName = "/Game/GenMaterials/" )
```

Create a Basic Material object.

Parameters

MaterialBaseName	
PackageName	

5.17.2.2 CreateCelShadedExplosionMat() `void FMaterialGenerator::CreateCelShadedExplosionMat (`

```
FString MaterialBaseName = FString("M_CelExplosionMat"),
FString PackageName = FString("/Game/GenMaterials/") )
```

Create a Cel Shaded Explosion Mat object.

Parameters

MaterialBaseName	
PackageName	

Todo Hook correct nodes toghether and then generate some banded textures to use. Also remember to set the material defaults to be able to use it as a particle material

5.17.2.3 CreateObject() FMaterialGenerator * FMaterialGenerator::CreateObject ( ) [static]

Create a FMaterialGenerator object.

Returns

FMaterialGenerator*

The documentation for this class was generated from the following files:

- Source/ario_stickyBomb_UE4/Public/Effects/MaterialGenerator.h
- Source/ario_stickyBomb_UE4/Private/Effects/MaterialGenerator.cpp

5.18 FPlayerData Struct Reference

Simple utility struct to pass data into widgets.

`#include <FPlayerData.h>`

Public Member Functions

- FPlayerData (float score=0.0f, FString name=FString("No Player Name"))
  Construct a new FPlayerData object.

Public Attributes

- float Score
- FString Name

5.18.1 Detailed Description

Simple utility struct to pass data into widgets.

5.18.2 Constructor & Destructor Documentation

5.18.2.1 FPlayerData() FPlayerData::FPlayerData (float score = 0.0f, FString name = FString("No Player Name")) [inline]

Construct a new FPlayerData object.

Parameters

score	
name	

5.18.3 Member Data Documentation

5.18.3.1 Name `FString FPlayerData::Name`

5.18.3.2 Score `float FPlayerData::Score`

The documentation for this struct was generated from the following file:

- Source/ario_stickyBomb_UE4/Public/UI/FPlayerData.h

5.19 FStickyFXManager Class Reference

A particle system primer class.

`#include <StickyFXManager.h>`

5.19.1 Detailed Description

A particle system primer class.

Todo Write functions to handle creation and editing of new particle systems through c++

Todo Write functions to handle creation and editing of new particle systems through c++

The documentation for this class was generated from the following file:

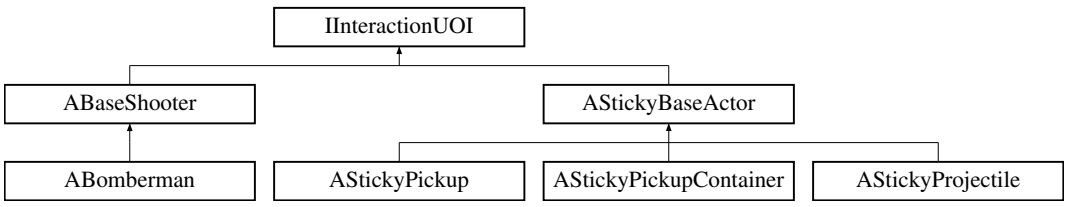
- Source/ario_stickyBomb_UE4/Private/Effects/StickyFXManager.cpp

5.20 IInteractionUOI Class Reference

A simple shell of a class to be expanded upon.

`#include <InteractionUOI.h>`

Inheritance diagram for IInteractionUOI:



Public Member Functions

- virtual void TryInteractItem ()=0

  Pure virtual, implement in child-classes.
- virtual void EndInteractItem ()=0

  Pure virtual, implement in child-classes.

5.20.1 Detailed Description

A simple shell of a class to be expanded upon.

Interaction Interface.

Simple Interaction Interface for managing pickups, could be more fleshed out if needed.

5.20.2 Member Function Documentation

5.20.2.1 EndInteractItem() `virtual void IInteractionUOI::EndInteractItem ( ) [pure virtual]`

Pure virtual, implement in child-classes.

Implemented in ABaseShooter, AStickyPickupContainer, AStickyPickup, AStickyBaseActor, and AStickyProjectile.

5.20.2.2 TryInteractItem() `virtual void IInteractionUOI::TryInteractItem ( ) [pure virtual]`

Pure virtual, implement in child-classes.

Implemented in ABaseShooter, AStickyPickupContainer, AStickyPickup, AStickyBaseActor, and AStickyProjectile.

The documentation for this class was generated from the following file:

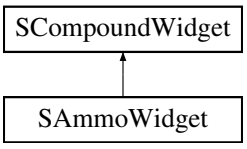
- Source/ario_stickyBomb_UE4/Public/Interfaces/InteractionUOI.h

5.21 SAmmoWidget Class Reference

Ammo Widget which derives from SCompoundWidget.

`#include <SAmmoWidget.h>`

Inheritance diagram for SAmmoWidget:



Public Member Functions

- SLATE_BEGIN_ARGS (SAmmoWidget)
- SLATE_ARGUMENT (TWeakObjectPtr< AStickyHUD >, OwnerHud)
- void Construct (const FArguments &InArgs)

  Construct widget.
- void SetAmmoText (FText Ammo)

  Set the Ammo Text object.
- FText GetAmmoText () const

  Get the Ammo Text object.

5.21.1 Detailed Description

Ammo Widget which derives from SCompoundWidget.

Simple widget to display the ammo count of the ammo component

5.21.2 Member Function Documentation

5.21.2.1 Construct() `void SAmmoWidget::Construct ( const FArguments & InArgs )`

Construct widget.

Parameters

InArgs	

5.21.2.2 GetAmmoText() `FText SAmmoWidget::GetAmmoText ( ) const`

Get the Ammo Text object.

Returns

FText

5.21.2.3 SetAmmoText() `void SAmmoWidget::SetAmmoText ( FText Ammo )`

Set the Ammo Text object.

Parameters

Ammo	

5.21.2.4 SLATE_ARGUMENT() `SAmmoWidget::SLATE_ARGUMENT ( TWeakObjectPtr< AStickyHUD > , OwnerHud )`

5.21.2.5 SLATE_BEGIN_ARGS() `SAmmoWidget::SLATE_BEGIN_ARGS ( SAmmoWidget ) [inline]`

The documentation for this class was generated from the following files:

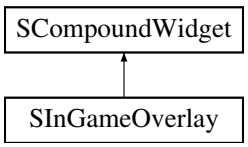
- Source/ario_stickyBomb_UE4/Public/UI/Widgets/SAmmoWidget.h
- Source/ario_stickyBomb_UE4/Private/UI/Widgets/SAmmoWidget.cpp

5.22 SInGameOverlay Class Reference

A simple shell of a class to be expanded upon. Right now it only funcitons as a blank overlay.

`#include <SInGameOverlay.h>`

Inheritance diagram for SInGameOverlay:



Public Member Functions

- SLATE_BEGIN_ARGS (SInGameOverlay)
- void Construct (const FArguments &InArgs)

  Construct widget.

5.22.1 Detailed Description

A simple shell of a class to be expanded upon. Right now it only funcitons as a blank overlay.

Game Overlay widget, derives from SCompoundWidget.

Simple widget to use a in game overlay.

5.22.2 Member Function Documentation

5.22.2.1 Construct() `BEGIN_SLATE_FUNCTION_BUILD_OPTIMIZATION void SInGameOverlay::Construct ( const FArguments & InArgs )`

Construct widget.

Parameters

InArgs	

5.22.2.2 SLATE_BEGIN_ARGS() `SInGameOverlay::SLATE_BEGIN_ARGS ( SInGameOverlay ) [inline]`

The documentation for this class was generated from the following files:

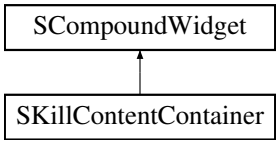
- Source/ario_stickyBomb_UE4/Public/UI/SInGameOverlay.h
- Source/ario_stickyBomb_UE4/Private/UI/SInGameOverlay.cpp

5.23 SKillContentContainer Class Reference

A simple shell of a class to be expanded upon.

`#include <SKillContentContainer.h>`

Inheritance diagram for SKillContentContainer:



Public Member Functions

- SLATE_BEGIN_ARGS (SKillContentContainer)
- void Construct (const FArguments &InArgs)

  Construct widget.
- void AddSlot (FString Value)

  Add Slot to widget.

Protected Member Functions

- TSharedRef< ITableRow > OnGenerateRowForList (TSharedPtr< FString > NewItem, const TSharedRef< STableViewBase > &OwnerTable)

  Generate row from table.

Protected Attributes

- AStickyHUD * OwnerHud
- TArray< TSharedPtr< FString > > Items
- TSharedPtr< SListView< TSharedPtr< FString > > > ListViewWidget
- FSlateFontInfo FontForKills

5.23.1 Detailed Description

A simple shell of a class to be expanded upon.

Kill Container widget, derives from SCompoundWidget.

Simple widget to display the kills in a cornered box.

5.23.2 Member Function Documentation

5.23.2.1 AddSlot() `void SKillContentContainer::AddSlot ( FString Value )`

Add Slot to widget.

Parameters

Value	

5.23.2.2 Construct() `BEGIN_SLATE_FUNCTION_BUILD_OPTIMIZATION void SKillContentContainer::Construct ( const FArguments & InArgs )`

Construct widget.

Parameters

InArgs	

5.23.2.3 OnGenerateRowForList() `TSharedRef< ITableRow > SKillContentContainer::OnGenerateRowForList ( TSharedPtr< FString > NewItem, const TSharedRef< STableViewBase > & OwnerTable ) [protected]`

Generate row from table.

Parameters

NewItem	
OwnerTable	

Returns

TSharedRef<ITableRow>

5.23.2.4 SLATE_BEGIN_ARGS() `SKillContentContainer::SLATE_BEGIN_ARGS ( SKillContentContainer ) [inline]`

5.23.3 Member Data Documentation

5.23.3.1 FontForKills `FSlateFontInfo SKillContentContainer::FontForKills [protected]`

5.23.3.2 Items `TArray<TSharedPtr<FString> > SKillContentContainer::Items [protected]`

5.23.3.3 ListViewWidget `TSharedPtr<SListView<TSharedPtr<FString> > > SKillContentContainer::ListViewWidget [protected]`

5.23.3.4 OwnerHud `AStickyHUD* SKillContentContainer::OwnerHud [protected]`

The documentation for this class was generated from the following files:

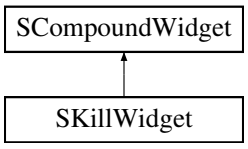
- Source/ario_stickyBomb_UE4/Public/UI/Widgets/SKillContentContainer.h
- Source/ario_stickyBomb_UE4/Private/UI/Widgets/SKillContentContainer.cpp

5.24 SKillWidget Class Reference

A simple shell of a class to be expanded upon to display current kills.

`#include <SKillWidget.h>`

Inheritance diagram for SKillWidget:



Public Member Functions

- SLATE_BEGIN_ARGS (SKillWidget)
- SLATE_ARGUMENT (TWeakObjectPtr< class AStickyHUD >, OwnerHud)
- void Construct (const FArguments &InArgs)

  Construct Widget.
- void SetScoreText (FText Score)

  Set the Score Text.
- FText GetScoreText () const

  Get the Score Text.

5.24.1 Detailed Description

A simple shell of a class to be expanded upon to display current kills.

Kill Count widget, derives from SCompoundWidget.

Simple widget to display the kill count.

5.24.2 Member Function Documentation

5.24.2.1 Construct() `void SKillWidget::Construct ( const FArguments & InArgs )`

Construct Widget.

Parameters

InArgs	

5.24.2.2 GetScoreText() `FText SKillWidget::GetScoreText ( ) const`

Get the Score Text.

Returns

FText

5.24.2.3 SetScoreText() `void SKillWidget::SetScoreText ( FText Score )`

Set the Score Text.

Parameters

Score	

5.24.2.4 SLATE_ARGUMENT() `SKillWidget::SLATE_ARGUMENT ( TWeakObjectPtr< class AStickyHUD > , OwnerHud )`

5.24.2.5 SLATE_BEGIN_ARGS() `SKillWidget::SLATE_BEGIN_ARGS ( SKillWidget ) [inline]`

The documentation for this class was generated from the following files:

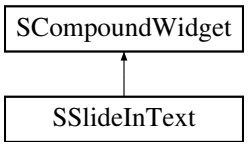
- Source/ario_stickyBomb_UE4/Public/UI/Widgets/SKillWidget.h
- Source/ario_stickyBomb_UE4/Private/UI/Widgets/SKillWidget.cpp

5.25 SSlideInText Class Reference

A simple widget that kan be used to display a recent player kill.

`#include <SSlideInText.h>`

Inheritance diagram for SSlideInText:



Public Member Functions

- SLATE_BEGIN_ARGS (SSlideInText)
- void Construct (const FArguments &InArgs)

  Construct widget.
- void TransitionIn ()

  Transitioning widget | in.
- void TransitionOut ()

  Transitioning widget | out.

Public Attributes

- FText TextToShow

Protected Member Functions

- FLinearColor GetColor () const

  Get the Color object.
- FVector2D GetItemScale () const

  Get the Item Scale object.

Protected Attributes

- EVisibleState CurrentState
- FCurveSequence VisibleAnimation
- FCurveHandle ScaleCurveX
- FCurveHandle ScaleCurveY
- FCurveSequence FadeAnimation
- FCurveHandle FadeValue

5.25.1 Detailed Description

A simple widget that kan be used to display a recent player kill.

Slide In Text widget, derives from SCompoundWidget.

Simple widget to slide in text upon a player kill.

5.25.2 Member Function Documentation

5.25.2.1 Construct() `BEGIN_SLATE_FUNCTION_BUILD_OPTIMIZATION void SSlideInText::Construct ( const FArguments & InArgs )`

Construct widget.

Parameters

InArgs	

5.25.2.2 GetColor() `FLinearColor SSlideInText::GetColor ( ) const [protected]`

Get the Color object.

Returns

FLinearColor

5.25.2.3 GetItemScale() FVector2D SSlideInText::GetItemScale ( ) const [protected]

Get the Item Scale object.

Returns

FVector2D

5.25.2.4 SLATE_BEGIN_ARGS() SSlideInText::SLATE_BEGIN_ARGS (SSlideInText) [inline]

5.25.2.5 TransitionIn() END_SLATE_FUNCTION_BUILD_OPTIMIZATION void SSlideInText::TransitionIn ( )

Transitioning widget | in.

5.25.2.6 TransitionOut() void SSlideInText::TransitionOut ( )

Transitioning widget | out.

5.25.3 Member Data Documentation

5.25.3.1 CurrentState EVisibleState SSlideInText::CurrentState [protected]

5.25.3.2 FadeAnimation FCurveSequence SSlideInText::FadeAnimation [protected]

5.25.3.3 FadeValue FCurveHandle SSlideInText::FadeValue [protected]

5.25.3.4 ScaleCurveX FCurveHandle SSlideInText::ScaleCurveX [protected]

5.25.3.5 ScaleCurveY `FCurveHandle SSlideInText::ScaleCurveY [protected]`

5.25.3.6 TextToShow `FText SSlideInText::TextToShow`

5.25.3.7 VisibleAnimation `FCurveSequence SSlideInText::VisibleAnimation [protected]`

The documentation for this class was generated from the following files:

- Source/ario_stickyBomb_UE4/Public/UI/Widgets/SSlideInText.h
- Source/ario_stickyBomb_UE4/Private/UI/Widgets/SSlideInText.cpp

5.26 UAmmoComp Class Reference

Simple ammo component.

`#include <AmmoComp.h>`

5.26.1 Detailed Description

Simple ammo component.

Networked ammo component.

A simple but networked ammo component.

Todo This component could be expaned into something more like an inventory. What would be needed is making a dedicated item_data class, hold the item_data in the inventory. Then spawn the items on server-side when brought out of inventory.

Simple networked Ammo Component to be used for ABaseShooter derived actors

The documentation for this class was generated from the following file:

- Source/ario_stickyBomb_UE4/Private/Components/AmmoComp.cpp

5.27 UHealthComp Class Reference

Networked health component.

`#include <HealthComp.h>`

5.27.1 Detailed Description

Networked health component.

Simple networked health component implementation, to be used with ABaseShooter derived characters, but could be retrofitted for other character types if it is rewritten as a templated class

Simple networked health component, to be used with ABaseShooter derived characters, but could be retrofitted for other character types if it is rewritten as a templated class

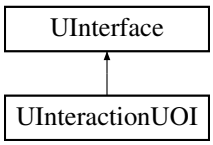
The documentation for this class was generated from the following file:

- Source/ario_stickyBomb_UE4/Private/Components/HealthComp.cpp

5.28 UInteractionUOI Class Reference

`#include <InteractionUOI.h>`

Inheritance diagram for UInteractionUOI:



The documentation for this class was generated from the following file:

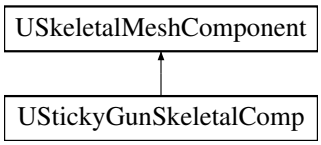
- Source/ario_stickyBomb_UE4/Public/Interfaces/InteractionUOI.h

5.29 UStickyGunSkeletalComp Class Reference

Networked weapon component.

`#include <StickyGunSkeletalComp.h>`

Inheritance diagram for UStickyGunSkeletalComp:



Public Member Functions

- UStickyGunSkeletalComp ()

  Construct a new UStickyGunSkeletalComp object.
- void InitStickyGun (ABaseShooter *Caller, FVector GunOffset, USceneComponent *MuzzlePlacementComp)

  Called after constructing StickyGunSkeletalComp as a subobject.
- void TryStartFire ()

  Request to fire weapon, does not guarantee a spawned projectile.
- UAmmoComp * GetAmmoComp ()

  Get the Ammo Comp object.
- ABaseShooter * GetOwningCharacter ()

  Get the Owning Character object.
- USoundBase * GetFireSound ()

  Get the Fire Sound object.
- UAnimMontage * GetFireAnimMontage ()

  Get the Fire Anim Montage object.
- void MulticastFireGunEffects (AStickyProjectile *LocalProjectileActorPtr)

  Multicasted VFX/SFX trigger.

Protected Member Functions

- void SuccessFireEffects ()

  Succeeded with fire effcts.
- void FailFireEffects ()

  Failed with fire effcts.
- void ServerOnFire ()

  Server Only OnFire event.
- void OnFire ()

  Tries spawning projectile, if successful then use a round of ammo.
- void PrepDeferredSpawnProjectile (AStickyProjectile *LocalProjectileActorPtr)

  Prepare Deferred Spawn of AStickyProjectile object.
- bool FinishSpawnProjectile (AStickyProjectile *LocalProjectileActorPtr, FTransform const &SpawnTransform)

  Finishes spawning the projectile.
- void GenerateCurve ()

  Generates a UCurve asset.

Protected Attributes

- USkeletalMesh * MeshPtr = nullptr
- FVector GunOffset

  Actual Skeletal Mesh asset.
- TSubclassOf< AStickyProjectile > ProjectileClass

  Gun muzzle's offset from the characters location.
- USoundBase * FireSound

  Projectile class to spawn.
- UAnimMontage * FireAnimation

  Sound to play each time we fire.
- USceneComponent * PlacementComp

  AnimMontage to play each time we fire.
- UAmmoComp * AmmoComp

  Spawn-point for projectiles.
- ABaseShooter * OwningCharacter
- UCurveFloat * FloatCurve = nullptr
- FRichCurve * GeneratedRichCurve

5.29.1 Detailed Description

Networked weapon component.

Derived from USkeletalMeshComponent, Designed to be used with ABaseShooter derived classes

5.29.2 Constructor & Destructor Documentation

5.29.2.1 UStickyGunSkeletalComp() `UStickyGunSkeletalComp::UStickyGunSkeletalComp ( )`

Construct a new UStickyGunSkeletalComp object.

5.29.3 Member Function Documentation

5.29.3.1 FailFireEffects() `void UStickyGunSkeletalComp::FailFireEffects ( ) [protected]`

Failed with fire effcts.

5.29.3.2 FinishSpawnProjectile() `bool UStickyGunSkeletalComp::FinishSpawnProjectile ( AStickyProjectile * LocalProjectileActorPtr, FTransform const & SpawnTransform ) [protected]`

Finishes spawning the projectile.

Parameters

LocalProjectileActorPtr	
SpawnTransform	

Returns

true | false (true if successful spawn)

5.29.3.3 GenerateCurve() `void UStickyGunSkeletalComp::GenerateCurve ( ) [protected]`

Generates a UCurve asset.

Generates a FRichCruve, point by point, then stores the FRichCurve into UCurve->CurveFloat

5.29.3.4 GetAmmoComp() UAmmoComp * UStickyGunSkeletalComp::GetAmmoComp ( )

Get the Ammo Comp object.

Returns

UAmmoComp*

5.29.3.5 GetFireAnimMontage() UAnimMontage * UStickyGunSkeletalComp::GetFireAnimMontage ( )

Get the Fire Anim Montage object.

Returns

UAnimMontage*

5.29.3.6 GetFireSound() USoundBase * UStickyGunSkeletalComp::GetFireSound ( )

Get the Fire Sound object.

Returns

USoundBase*

5.29.3.7 GetOwningCharacter() ABaseShooter * UStickyGunSkeletalComp::GetOwningCharacter ( )

Get the Owning Character object.

Returns

ABaseShooter*

5.29.3.8 InitStickyGun() void UStickyGunSkeletalComp::InitStickyGun (ABaseShooter * *Caller*, FVector *GunOffset*, USceneComponent * *MuzzlePlacementComp*)

Called after constructing StickyGunSkeletalComp as a subobject.

Parameters

Caller	
GunOffset	
MuzzlePlacementComp	

5.29.3.9 MulticastFireGunEffects() `void UStickyGunSkeletalComp::MulticastFireGunEffects ( AStickyProjectile * LocalProjectileActorPtr )`

Multicasted VFX/SFX trigger.

Parameters

LocalProjectileActorPtr	

5.29.3.10 OnFire() `void UStickyGunSkeletalComp::OnFire ( ) [protected]`

Tries spawning projectile, if successful then use a round of ammo.

5.29.3.11 PrepDeferredSpawnProjectile() `void UStickyGunSkeletalComp::PrepDeferredSpawnProjectile ( AStickyProjectile * LocalProjectileActorPtr ) [protected]`

Prepare Deferred Spawn of AStickyProjectile object.

Takes the deferred spawn and sets some member of the object before actually executing the spawn

Parameters

LocalProjectileActorPtr	

5.29.3.12 ServerOnFire() `void UStickyGunSkeletalComp::ServerOnFire ( ) [protected]`

Server Only OnFire event.

When called only server can handle it, from there server calls protected OnFire method

5.29.3.13 SuccessFireEffects() `void UStickyGunSkeletalComp::SuccessFireEffects ( ) [protected]`

Succeeded with fire effcts.

5.29.3.14 TryStartFire() `void UStickyGunSkeletalComp::TryStartFire ( )`

Request to fire weapon, does not guarantee a spawned projectile.

5.29.4 Member Data Documentation

5.29.4.1 AmmoComp `UAmmoComp* UStickyGunSkeletalComp::AmmoComp [protected]`

Spawn-point for projectiles.

5.29.4.2 FireAnimation `UAnimMontage* UStickyGunSkeletalComp::FireAnimation [protected]`

Sound to play each time we fire.

5.29.4.3 FireSound `USoundBase* UStickyGunSkeletalComp::FireSound [protected]`

Projectile class to spawn.

5.29.4.4 FloatCurve `UCurveFloat* UStickyGunSkeletalComp::FloatCurve = nullptr [protected]`

5.29.4.5 GeneratedRichCurve `FRichCurve* UStickyGunSkeletalComp::GeneratedRichCurve [protected]`

5.29.4.6 GunOffset `FVector UStickyGunSkeletalComp::GunOffset [protected]`

Actual Skeletal Mesh asset.

5.29.4.7 MeshPtr `USkeletalMesh* UStickyGunSkeletalComp::MeshPtr = nullptr [protected]`

5.29.4.8 OwningCharacter `ABaseShooter* UStickyGunSkeletalComp::OwningCharacter [protected]`

5.29.4.9 PlacementComp `USceneComponent* UStickyGunSkeletalComp::PlacementComp [protected]`

AnimMontage to play each time we fire.

5.29.4.10 ProjectileClass `TSubclassOf<AStickyProjectile> UStickyGunSkeletalComp::ProjectileClass [protected]`

Gun muzzle's offset from the characters location.

The documentation for this class was generated from the following files:

- Source/ario_stickyBomb_UE4/Public/Components/StickyGunSkeletalComp.h
- Source/ario_stickyBomb_UE4/Private/Components/StickyGunSkeletalComp.cpp

5.30 UStickyLinetraceComp Class Reference

Not much more than a linetracer. Almost a functor,.

`#include <StickyLinetraceComp.h>`

5.30.1 Detailed Description

Not much more than a linetracer. Almost a functor,.

It is not quite a functor, but it is very small and serves only one function, the tick-component function. It runs a tick on 0.2 second period, in which it shoots a linetrace about 2.5 meters adjusted from Unreal Units. It is designed in a way that it is attached to the camera of ABaseShooter (and derived) actors and will activate the component by pressing the designated key for it,

The documentation for this class was generated from the following file:

- Source/ario_stickyBomb_UE4/Private/Components/StickyLinetraceComp.cpp

6 File Documentation

6.1 Source/ario_stickyBomb_UE4.Target.cs File Reference

Classes

- class ario_stickyBomb_UE4Target

6.2 Source/ario_stickyBomb_UE4/ario_stickyBomb_UE4.Build.cs File Reference

Classes

- class ario_stickyBomb_UE4

6.3 Source/ario_stickyBomb_UE4/ario_stickyBomb_UE4.cpp File Reference

```
#include "ario_stickyBomb_UE4.h"
#include "Modules/ModuleManager.h"
```

Functions

- IMPLEMENT_PRIMARY_GAME_MODULE (FDefaultGameModuleImpl, ario_stickyBomb_UE4, "ario_↩ stickyBomb_UE4")

6.3.1 Function Documentation

6.3.1.1 IMPLEMENT_PRIMARY_GAME_MODULE()

```
IMPLEMENT_PRIMARY_GAME_MODULE (
        FDefaultGameModuleImpl ,
        ario_stickyBomb_UE4 ,
        "ario_stickyBomb_UE4"  )
```

6.4 Source/ario_stickyBomb_UE4/ario_stickyBomb_UE4.h File Reference

```
#include "CoreMinimal.h"
```

6.5 Source/ario_stickyBomb_UE4/Private/Actors/StickyBaseActor.cpp File Reference

```
#include "Actors/StickyBaseActor.h"
```

6.6 Source/ario_stickyBomb_UE4/Private/Actors/StickyPickup.cpp File Reference

```
#include "Actors/StickyPickup.h"
#include "Characters/BaseShooter.h"
#include "Interfaces/InteractionUOI.h"
#include "Components/AmmoComp.h"
#include "Components/StickyGunSkeletalComp.h"
#include "Helpers/CollisionChannels.h"
#include "Helpers/Macros.h"
#include <Kismet/KismetSystemLibrary.h>
#include <Net/UnrealNetwork.h>
#include <UObject/ConstructorHelpers.h>
#include <Components/SphereComponent.h>
#include <Materials/MaterialInstanceDynamic.h>
```

6.7 Source/ario_stickyBomb_UE4/Private/Actors/StickyPickupContainer.cpp File Reference

```
#include "Actors/StickyPickupContainer.h"
#include "Helpers/CollisionChannels.h"
#include "Helpers/Macros.h"
#include <Kismet/KismetSystemLibrary.h>
#include <Net/UnrealNetwork.h>
#include <UObject/ConstructorHelpers.h>
#include <Materials/MaterialInstanceDynamic.h>
```

6.7.1 Detailed Description

Author

Ario Amin

6.8 Source/ario_stickyBomb_UE4/Private/Actors/StickyProjectile.cpp File Reference

```
#include "Actors/StickyProjectile.h"
#include "Characters/BaseShooter.h"
#include "Interfaces/InteractionUOI.h"
#include "StickyGameMode.h"
#include "StickyPlayerState.h"
#include "Components/AmmoComp.h"
#include "Components/HealthComp.h"
#include "Components/StickyGunSkeletalComp.h"
#include "Helpers/CollisionChannels.h"
#include "Helpers/Macros.h"
#include <Kismet/GameplayStatics.h>
#include <Kismet/KismetSystemLibrary.h>
#include <Net/UnrealNetwork.h>
#include <UObject/ConstructorHelpers.h>
#include <Components/SphereComponent.h>
#include <GameFramework/ProjectileMovementComponent.h>
#include <Particles/ParticleSystemComponent.h>
#include <GameFramework/DamageType.h>
#include <Materials/MaterialInstanceDynamic.h>
#include <Sound/SoundBase.h>
```

6.8.1 Detailed Description

Author

Ario Amin

6.9 Source/ario_stickyBomb_UE4/Private/Characters/BaseShooter.cpp File Reference

```
#include "Characters/BaseShooter.h"
#include "Actors/StickyProjectile.h"
#include "Helpers/CollisionChannels.h"
#include "StickyPlayerState.h"
#include "Components/AmmoComp.h"
#include "Components/HealthComp.h"
#include "Components/StickyGunSkeletalComp.h"
#include "Components/StickyLinetraceComp.h"
#include <Camera/CameraComponent.h>
#include <Components/CapsuleComponent.h>
#include <Components/InputComponent.h>
#include <GameFramework/CharacterMovementComponent.h>
#include <MotionControllerComponent.h>
#include <Animation/AnimInstance.h>
#include <GameFramework/DamageType.h>
#include <GameFramework/InputSettings.h>
#include <Net/UnrealNetwork.h>
#include <Animation/AnimBlueprint.h>
#include <Kismet/GameplayStatics.h>
#include <UObject/ConstructorHelpers.h>
#include <HeadMountedDisplayFunctionLibrary.h>
#include <XRMotionControllerBase.h>
```

Functions

- DEFINE_LOG_CATEGORY_STATIC (LogFPChar, Warning, All)

6.9.1 Function Documentation

6.9.1.1 DEFINE_LOG_CATEGORY_STATIC()

```
DEFINE_LOG_CATEGORY_STATIC (
        LogFPChar ,
        Warning ,
        All  )
```

6.10 Source/ario_stickyBomb_UE4/Private/Characters/Bomberman.cpp File Reference

```
#include "Characters/Bomberman.h"
#include "Components/StickyGunSkeletalComp.h"
#include "Interfaces/InteractionUOI.h"
#include <Camera/CameraComponent.h>
#include <Components/CapsuleComponent.h>
#include <Components/InputComponent.h>
#include <MotionControllerComponent.h>
#include <Animation/AnimInstance.h>
#include <GameFramework/InputSettings.h>
#include <GameFramework/PlayerInput.h>
#include <HeadMountedDisplayFunctionLibrary.h>
#include <XRMotionControllerBase.h>
#include <Kismet/GameplayStatics.h>
#include <UObject/UObjectGlobals.h>
```

Functions

- DEFINE_LOG_CATEGORY_STATIC (LogFPChar, Warning, All)

6.10.1 Function Documentation

6.10.1.1 DEFINE_LOG_CATEGORY_STATIC()

```
DEFINE_LOG_CATEGORY_STATIC (
        LogFPChar ,
        Warning ,
        All  )
```

6.11 Source/ario_stickyBomb_UE4/Private/Components/AmmoComp.cpp File Reference

```
#include "Components/AmmoComp.h"
#include "Characters/BaseShooter.h"
#include "StickyGameMode.h"
#include "Helpers/Macros.h"
#include <Net/UnrealNetwork.h>
```

6.11.1 Detailed Description

Author

Ario Amin

6.12 Source/ario_stickyBomb_UE4/Private/Components/HealthComp.cpp File Reference

```
#include "Components/HealthComp.h"
#include "StickyGameMode.h"
#include "Helpers/Macros.h"
#include <GameFramework/Actor.h>
#include <Math/UnrealMathUtility.h>
#include <Net/UnrealNetwork.h>
```

6.12.1 Detailed Description

Author

Ario Amin

6.13 Source/ario_stickyBomb_UE4/Private/Components/StickyGunSkeletalComp.cpp File Reference

```
#include "Components/StickyGunSkeletalComp.h"
#include "Actors/StickyProjectile.h"
#include "Components/AmmoComp.h"
#include "Helpers/CollisionChannels.h"
#include <Components/SkeletalMeshComponent.h>
#include <Kismet/GameplayStatics.h>
#include <Net/UnrealNetwork.h>
#include <UObject/ConstructorHelpers.h>
#include <Animation/AnimInstance.h>
#include <Sound/SoundBase.h>
```

6.13.1 Detailed Description

Author

Ario Amin

6.14 Source/ario_stickyBomb_UE4/Private/Components/StickyLinetraceComp.cpp File Reference

```
#include "Components/StickyLinetraceComp.h"
#include "Actors/StickyBaseActor.h"
#include "Characters/BaseShooter.h"
#include "Components/AmmoComp.h"
#include "Interfaces/InteractionUOI.h"
#include <Kismet/KismetSystemLibrary.h>
```

6.15 Source/ario_stickyBomb_UE4/Private/Effects/MaterialGenerator.cpp File Reference

```
#include "Effects/MaterialGenerator.h"
#include <AssetRegistry/AssetRegistryModule.h>
#include <Factories/MaterialFactoryNew.h>
#include <Materials/MaterialExpressionAppendVector.h>
#include <MaterialExpressionIO.h>
#include <Materials/MaterialExpressionAdd.h>
#include <Materials/MaterialExpressionCeil.h>
#include <Materials/MaterialExpressionComponentMask.h>
#include <Materials/MaterialExpressionConstant.h>
#include <Materials/MaterialExpressionConstant3Vector.h>
#include <Materials/MaterialExpressionDotProduct.h>
#include <Materials/MaterialExpressionDynamicParameter.h>
#include <Materials/MaterialExpressionLinearInterpolate.h>
#include <Materials/MaterialExpressionMultiply.h>
#include <Materials/MaterialExpressionOneMinus.h>
#include <Materials/MaterialExpressionPanner.h>
#include <Materials/MaterialExpressionSaturate.h>
```

```
#include <Materials/MaterialExpressionScalarParameter.h>
#include <Materials/MaterialExpressionSubtract.h>
#include <Materials/MaterialExpressionTextureCoordinate.h>
#include <Materials/MaterialExpressionTextureSample.h>
#include <Materials/MaterialExpressionVectorParameter.h>
#include <Materials/MaterialExpressionVertexNormalWS.h>
```

6.15.1 Detailed Description

Author

Ario Amin

6.16 Source/ario_stickyBomb_UE4/Private/Effects/StickyFXManager.cpp File Reference

6.16.1 Detailed Description

Author

Ario Amin

6.17 Source/ario_stickyBomb_UE4/Private/Interfaces/InteractionUOI.cpp File Reference

```
#include "Interfaces/InteractionUOI.h"
```

6.17.1 Detailed Description

Author

Ario Amin

6.18 Source/ario_stickyBomb_UE4/Private/StickyGameMode.cpp File Reference

```
#include "StickyGameMode.h"
#include "Characters/Bomberman.h"
#include "StickyGameState.h"
#include "StickyPlayerController.h"
#include "StickyPlayerState.h"
#include "Components/AmmoComp.h"
#include "Components/HealthComp.h"
#include "UI/StickyHUD.h"
#include "Effects/MaterialGenerator.h"
#include <Engine/World.h>
#include <EngineUtils.h>
#include <GameFramework/Actor.h>
#include <TimerManager.h>
#include <UObject/ConstructorHelpers.h>
#include <Widgets/DeclarativeSyntaxSupport.h>
```

6.18.1 Detailed Description

Author

Ario Amin

6.19 Source/ario_stickyBomb_UE4/Private/StickyGameSession.cpp File Reference

```
#include "StickyGameSession.h"
```

6.19.1 Detailed Description

Author

Ario Amin

6.20 Source/ario_stickyBomb_UE4/Private/StickyGameState.cpp File Reference

```
#include "StickyGameState.h"
```

6.20.1 Detailed Description

Author

Ario Amin

6.21 Source/ario_stickyBomb_UE4/Private/StickyPlayerCameraManager.cpp File Reference

```
#include "StickyPlayerCameraManager.h"
```

6.21.1 Detailed Description

Author

Ario Amin

6.22 Source/ario_stickyBomb_UE4/Private/StickyPlayerController.cpp File Reference

```
#include "StickyPlayerController.h"
#include "StickyPlayerCameraManager.h"
```

6.22.1 Detailed Description

Author

Ario Amin

6.23 Source/ario_stickyBomb_UE4/Private/StickyPlayerState.cpp File Reference

```
#include "StickyPlayerState.h"
#include "StickyGameMode.h"
#include "UI/StickyHUD.h"
#include <Engine/Engine.h>
#include <Kismet/GameplayStatics.h>
#include <Net/UnrealNetwork.h>
```

6.23.1 Detailed Description

Author

Ario Amin

6.24 Source/ario_stickyBomb_UE4/Private/UI/SInGameOverlay.cpp File Reference

```
#include "UI/SInGameOverlay.h"
#include "SlateOptMacros.h"
```

6.24.1 Detailed Description

Author

Ario Amin

6.25 Source/ario_stickyBomb_UE4/Private/UI/StickyHUD.cpp File Reference

```
#include "UI/StickyHUD.h"
#include "StickyGameMode.h"
#include "StickyPlayerState.h"
#include "Helpers/Macros.h"
#include "UI/Widgets/SAmmoWidget.h"
#include "UI/Widgets/SKillContentContainer.h"
#include "UI/Widgets/SKillWidget.h"
#include "UI/Widgets/SSlideInText.h"
#include <Engine/Engine.h>
#include <Widgets/DeclarativeSyntaxSupport.h>
#include <Widgets/SWeakWidget.h>
```

6.25.1 Detailed Description

Author

Ario Amin

6.26 Source/ario_stickyBomb_UE4/Private/UI/Widgets/SAmmoWidget.cpp File Reference

```
#include "UI/Widgets/SAmmoWidget.h"
#include "Helpers/Macros.h"
#include "SlateOptMacros.h"
#include <Internationalization/Internationalization.h>
#include <Widgets/DeclarativeSyntaxSupport.h>
#include <Widgets/SOverlay.h>
#include <Widgets/Text/STextBlock.h>
```

Macros

- #define LOCTEXT_NAMESPACE "SAmmoWidget"

6.26.1 Macro Definition Documentation

6.26.1.1 LOCTEXT_NAMESPACE `#define LOCTEXT_NAMESPACE "SAmmoWidget"`

6.27 Source/ario_stickyBomb_UE4/Private/UI/Widgets/SKillContentContainer.cpp File Reference

```
#include "UI/Widgets/SKillContentContainer.h"
#include "SlateOptMacros.h"
#include "UI/Widgets/SSlideInText.h"
#include <Engine/World.h>
#include <Containers/UnrealString.h>
#include <Templates/SharedPointer.h>
#include <Widgets/Input/SButton.h>
#include <Widgets/Layout/SWrapBox.h>
#include <Widgets/SCompoundWidget.h>
#include <Widgets/Views/SListView.h>
#include <Widgets/Views/STableRow.h>
```

6.28 Source/ario_stickyBomb_UE4/Private/UI/Widgets/SKillWidget.cpp File Reference

```
#include "UI/Widgets/SKillWidget.h"
#include "SlateOptMacros.h"
#include <Internationalization/Internationalization.h>
#include <Widgets/DeclarativeSyntaxSupport.h>
#include <Widgets/SOverlay.h>
#include <Widgets/Text/STextBlock.h>
```

Macros

- #define LOCTEXT_NAMESPACE "SScoreWidget"

6.28.1 Detailed Description

Author

Ario Amin

6.28.2 Macro Definition Documentation

6.28.2.1 LOCTEXT_NAMESPACE `#define LOCTEXT_NAMESPACE "SScoreWidget"`

6.29 Source/ario_stickyBomb_UE4/Private/UI/Widgets/SSlideInText.cpp File Reference

```
#include "UI/Widgets/SSlideInText.h"
#include "SlateOptMacros.h"
#include <Animation/CurveHandle.h>
#include <Animation/CurveSequence.h>
#include <Engine/World.h>
#include <Engine/EngineTypes.h>
#include <Fonts/SlateFontInfo.h>
#include <Misc/Paths.h>
#include <Widgets/DeclarativeSyntaxSupport.h>
#include <Widgets/Layout/SBorder.h>
#include <Widgets/Layout/SConstraintCanvas.h>
#include <Widgets/SCanvas.h>
#include <Widgets/SCompoundWidget.h>
#include <Widgets/SOverlay.h>
```

6.29.1 Detailed Description

Author

Ario Amin

6.30 Source/ario_stickyBomb_UE4/Public/Actors/StickyBaseActor.h File Reference

```
#include "CoreMinimal.h"
#include "Interfaces/InteractionUOI.h"
#include <GameFramework/Actor.h>
#include "StickyBaseActor.generated.h"
```

Classes

- class AStickyBaseActor

  A simple shell of a class to be expanded upon.

6.30.1 Detailed Description

Author

Ario Amin

6.31 Source/ario_stickyBomb_UE4/Public/Actors/StickyPickup.h File Reference

```
#include "Actors/StickyBaseActor.h"
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
#include "Interfaces/InteractionUOI.h"
#include <Templates/SharedPointer.h>
#include "StickyPickup.generated.h"
```

Classes

- class AStickyPickup

  A simple pickup actor.

6.31.1 Detailed Description

Author

Ario Amin

6.32 Source/ario_stickyBomb_UE4/Public/Actors/StickyPickupContainer.h File Reference

```
#include "Actors/StickyBaseActor.h"
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
#include "Interfaces/InteractionUOI.h"
#include <Templates/SharedPointer.h>
#include "StickyPickupContainer.generated.h"
```

Classes

- class AStickyPickupContainer

  A simple actor which holds a static mesh container.

6.32.1 Detailed Description

Author

Ario Amin

6.33 Source/ario_stickyBomb_UE4/Public/Actors/StickyProjectile.h File Reference

```
#include "Actors/StickyBaseActor.h"
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
#include <Components/TimelineComponent.h>
#include "StickyProjectile.generated.h"
```

Classes

- class AStickyProjectile

  A projectile moving actor class.

6.33.1 Detailed Description

Author

Ario Amin

6.34 Source/ario_stickyBomb_UE4/Public/Characters/BaseShooter.h File Reference

```
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
#include "Interfaces/InteractionUOI.h"
#include <Delegates/Delegate.h>
#include <GameFramework/Character.h>
#include "BaseShooter.generated.h"
```

Classes

- class ABaseShooter

6.35 Source/ario_stickyBomb_UE4/Public/Characters/Bomberman.h File Reference

```
#include "Characters/BaseShooter.h"
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
#include "Bomberman.generated.h"
```

Classes

- class ABomberman

  Inherits from ACharacter & IInteractionUOI.

6.36 Source/ario_stickyBomb_UE4/Public/Components/AmmoComp.h File Reference

```
#include "Components/ActorComponent.h"
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
#include "Helpers/Macros.h"
#include <Engine/EngineTypes.h>
#include "AmmoComp.generated.h"
```

Functions

- DECLARE_DYNAMIC_MULTICAST_DELEGATE_FourParams (FOnAmmoChangedSignature, UAmmoComp ∗, OwningAmmoComp, int, Ammo, int, AmmoDelta, AController ∗, InstigatedBy)

6.36.1 Detailed Description

Author

Ario Amin

6.36.2 Function Documentation

6.36.2.1 DECLARE_DYNAMIC_MULTICAST_DELEGATE_FourParams()

```
DECLARE_DYNAMIC_MULTICAST_DELEGATE_FourParams (
            FOnAmmoChangedSignature ,
            UAmmoComp * ,
            OwningAmmoComp ,
            int ,
            Ammo ,
            int ,
            AmmoDelta ,
            AController * ,
            InstigatedBy  )
```

6.37 Source/ario_stickyBomb_UE4/Public/Components/HealthComp.h File Reference

```
#include "Components/ActorComponent.h"
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
#include <Engine/EngineTypes.h>
#include "HealthComp.generated.h"
```

Functions

- DECLARE_DYNAMIC_MULTICAST_DELEGATE_SixParams (FOnHealthChangedSignature, UHealthComp ∗, OwningHealthComp, float, Health, float, HealthDelta, const UDamageType ∗, DamageType, AController ∗, InstigatedBy, AActor ∗, DamageCauser)

6.37.1 Detailed Description

Author

Ario Amin

6.37.2 Function Documentation

6.37.2.1 DECLARE_DYNAMIC_MULTICAST_DELEGATE_SixParams() DECLARE_DYNAMIC_MULTICAST_↩ DELEGATE_SixParams (

```
            FOnHealthChangedSignature ,
            UHealthComp * ,
            OwningHealthComp ,
            float ,
            Health ,
            float ,
            HealthDelta ,
            const UDamageType * ,
            DamageType ,
            AController * ,
            InstigatedBy ,
            AActor * ,
            DamageCauser  )
```

6.38 Source/ario_stickyBomb_UE4/Public/Components/StickyGunSkeletalComp.h File Reference

```
#include "Characters/BaseShooter.h"
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
#include <Components/SkeletalMeshComponent.h>
#include "StickyGunSkeletalComp.generated.h"
```

Classes

- class UStickyGunSkeletalComp

  Networked weapon component.

6.38.1 Detailed Description

Author

Ario Amin

6.39 Source/ario_stickyBomb_UE4/Public/Components/StickyLinetraceComp.h File Reference

```
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
#include "Helpers/Macros.h"
#include <Components/SceneComponent.h>
#include "StickyLinetraceComp.generated.h"
```

6.40 Source/ario_stickyBomb_UE4/Public/Effects/MaterialGenerator.h File Reference

```
#include "CoreMinimal.h"
```

Classes

- class FMaterialGenerator

  A WIP utility class for Material creation/editing.

6.40.1 Detailed Description

Author

Ario Amin

6.41 Source/ario_stickyBomb_UE4/Public/Effects/StickyFXManager.h File Reference

6.41.1 Detailed Description

Author

Ario Amin

6.42 Source/ario_stickyBomb_UE4/Public/Helpers/CollisionChannels.h File Reference

Collision Channels defintions.

Macros

- #define ECC_StickyProjectile ECollisionChannel::ECC_GameTraceChannel3
- #define ECC_StickyGun ECollisionChannel::ECC_GameTraceChannel4
- #define ECC_CharacterMesh ECollisionChannel::ECC_GameTraceChannel5

6.42.1 Detailed Description

Collision Channels defintions.

Author

Ario Amin

6.42.2 Macro Definition Documentation

6.42.2.1 ECC_CharacterMesh `#define ECC_CharacterMesh ECollisionChannel::ECC_GameTrace↩ Channel5`

6.42.2.2 ECC_StickyGun `#define ECC_StickyGun ECollisionChannel::ECC_GameTraceChannel4`

6.42.2.3 ECC_StickyProjectile `#define ECC_StickyProjectile ECollisionChannel::ECC_GameTrace↩ Channel3`

6.43 Source/ario_stickyBomb_UE4/Public/Helpers/ForwardDecls.h File Reference

Forward Declarations.

6.43.1 Detailed Description

Forward Declarations.

Author

Ario Amin

Not pretty, but makes the actual headers look a bit cleaner

6.44 Source/ario_stickyBomb_UE4/Public/Helpers/Macros.h File Reference

Various macros.

Macros

- #define STICKY_DEBUG
- #define LL_IF(A, B, CondA) (A ∗ (CondA)) + (B ∗ !(CondA))

  Branchless Numeric If.
- #define LL_IFELSE(A, B, CondA, CondB) (A ∗ (CondA)) + (B ∗ (CondB))

  Branchless Numeric 'If-Else'.
- #define LL_CLAMP(Value, Min, Max) (0 ∗ (Value < Min)) + (Max ∗ (Value > Max)) + (Value ∗ (Value >= Min && Value <= Max))

  Branchless Arithmetic Clamp.
- #define DEBUG_PRINT_LOC(Position) UE_LOG(LogTemp, Warning, TEXT("git commit -S -m \"LOCATION: {%f,%f,%f}!\""), Position.X, Position.Y, Position.Z);

  Print Location.
- #define DEFAULT_STICKY_GUN_MAX 3
- #define MAKETEXT(InString) Text(FText::FromString(InString))

  Regarding Slate SWidget based classes prefix with . when using these during create using SNew()
- #define MAKETEXTBOUND(Function) Text(this, &Function)
- #define MAKEROBOTO(FontSize) Font(FSlateFontInfo(FPaths::EngineContentDir() / TEXT("Slate/Fonts/Roboto-Bold.ttf"), FontSize))
- #define MAKECOLOR(FLinearColorA, FLinearColorB) ShadowColorAndOpacity(FLinearColorA).Color↩ AndOpacity(FLinearColorB)
- #define MAKESHADOW_OFFSET(OffsetMin, OffsetMax) ShadowOffset(FIntPoint(OffsetMin, OffsetMax))

6.44.1 Detailed Description

Various macros.

Author

Ario Amin

Some arithmetic if statements for efficieny and some shorthands for nasty Slate function calls

6.44.2 Macro Definition Documentation

6.44.2.1 DEBUG_PRINT_LOC `#define DEBUG_PRINT_LOC(`

```
Position ) UE_LOG(LogTemp, Warning, TEXT("git commit -S -m \"LOCATION: {%f,%f,%f}!\""),
Position.X, Position.Y, Position.Z);
```

Print Location.

6.44.2.2 DEFAULT_STICKY_GUN_MAX `#define DEFAULT_STICKY_GUN_MAX 3`

6.44.2.3 LL_CLAMP #define LL_CLAMP(

```
Value,
Min,
Max ) (0 * (Value < Min)) + (Max * (Value > Max)) + (Value * (Value >= Min &&
Value <= Max))
```

Branchless Arithmetic Clamp.

6.44.2.4 LL_IF #define LL_IF(

```
A,
B,
CondA ) (A * (CondA)) + (B * !(CondA))
```

Branchless Numeric If.

6.44.2.5 LL_IFELSE #define LL_IFELSE(

```
A,
B,
CondA,
CondB ) (A * (CondA)) + (B * (CondB))
```

Branchless Numeric 'If-Else'.

6.44.2.6 MAKECOLOR #define MAKECOLOR(

```
FLinearColorA,
FLinearColorB ) ShadowColorAndOpacity(FLinearColorA).ColorAndOpacity(FLinear↩
ColorB)
```

6.44.2.7 MAKEROBOTO #define MAKEROBOTO(

```
FontSize ) Font(FSlateFontInfo(FPaths::EngineContentDir() / TEXT("Slate/Fonts/Roboto-Bold.↩
ttf"), FontSize))
```

6.44.2.8 MAKESHADOW_OFFSET #define MAKESHADOW_OFFSET(

```
OffsetMin,
OffsetMax ) ShadowOffset(FIntPoint(OffsetMin, OffsetMax))
```

6.44.2.9 MAKETEXT #define MAKETEXT(

```
InString ) Text(FText::FromString(InString))
```

Regarding Slate SWidget based classes prefix with . when using these during create using SNew()

6.44.2.10 MAKETEXTBOUND

```
#define MAKETEXTBOUND(
    Function ) Text(this, &Function)
```

6.44.2.11 STICKY_DEBUG

```
#define STICKY_DEBUG
```

6.45 Source/ario_stickyBomb_UE4/Public/Interfaces/InteractionUOI.h File Reference

```
#include "CoreMinimal.h"
#include <UObject/Interface.h>
#include "InteractionUOI.generated.h"
```

Classes

- class UInteractionUOI
- class IInteractionUOI

  A simple shell of a class to be expanded upon.

6.45.1 Detailed Description

Author

Ario Amin

6.46 Source/ario_stickyBomb_UE4/Public/StickyGameMode.h File Reference

```
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
#include <Delegates/Delegate.h>
#include <GameFramework/GameModeBase.h>
#include "StickyGameMode.generated.h"
```

Classes

- class AStickyGameMode

  Naive implementation of Gamemode for networked play.

Functions

- DECLARE_DYNAMIC_MULTICAST_DELEGATE_ThreeParams (FOnActorKilled, AActor ∗, VictimActor, AActor ∗, KillerActor, AController ∗, KillerController)

  Declare a new dynamic multicast delegate with three parameters.
- DECLARE_DYNAMIC_MULTICAST_DELEGATE_FourParams (FOnTargetHitEvent, FVector, HitLocation, FVector, ShotDirection, float, HitValue, AActor ∗, HitOwner)

  Declare a new dynamic multicast delegate with four parameters.

6.46.1 Detailed Description

Author

Ario Amin

6.46.2 Function Documentation

6.46.2.1 DECLARE_DYNAMIC_MULTICAST_DELEGATE_FourParams() DECLARE_DYNAMIC_MULTICAST_↩ DELEGATE_FourParams (

```
            FOnTargetHitEvent ,
            FVector ,
            HitLocation ,
            FVector ,
            ShotDirection ,
            float ,
            HitValue ,
            AActor * ,
            HitOwner  )
```

Declare a new dynamic multicast delegate with four parameters.

6.46.2.2 DECLARE_DYNAMIC_MULTICAST_DELEGATE_ThreeParams() DECLARE_DYNAMIC_MULTICAST_↩ DELEGATE_ThreeParams (

```
            FOnActorKilled ,
            AActor * ,
            VictimActor ,
            AActor * ,
            KillerActor ,
            AController * ,
            KillerController  )
```

Declare a new dynamic multicast delegate with three parameters.

6.47 Source/ario_stickyBomb_UE4/Public/StickyGameSession.h File Reference

```
#include "CoreMinimal.h"
#include "GameFramework/GameSession.h"
#include "StickyGameSession.generated.h"
```

Classes

- class AStickyGameSession

  Dummy gamesession class.

6.47.1 Detailed Description

Author

Ario Amin

6.48 Source/ario_stickyBomb_UE4/Public/StickyGameState.h File Reference

```
#include "CoreMinimal.h"
#include "GameFramework/GameStateBase.h"
#include "StickyGameState.generated.h"
```

Classes

- class AStickyGameState

  A simple shell of a class implementation to be expanded upon.

6.48.1 Detailed Description

Author

Ario Amin

6.49 Source/ario_stickyBomb_UE4/Public/StickyPlayerCameraManager.h File Reference

```
#include "Camera/PlayerCameraManager.h"
#include "CoreMinimal.h"
#include "StickyPlayerCameraManager.generated.h"
```

Classes

- class AStickyPlayerCameraManager

  A simple shell of a class implementation to be expanded upon.

6.49.1 Detailed Description

Author

Ario Amin

6.50 Source/ario_stickyBomb_UE4/Public/StickyPlayerController.h File Reference

```
#include "CoreMinimal.h"
#include "GameFramework/PlayerController.h"
#include "StickyPlayerController.generated.h"
```

Classes

- class AStickyPlayerController

  A simple shell of a class implementation to be expanded upon.

6.50.1 Detailed Description

Author

Ario Amin

6.51 Source/ario_stickyBomb_UE4/Public/StickyPlayerState.h File Reference

```
#include "CoreMinimal.h"
#include "StickyGameMode.h"
#include <GameFramework/PlayerState.h>
#include "StickyPlayerState.generated.h"
```

Classes

- class AStickyPlayerState

  Networked Player-state implementation.

6.51.1 Detailed Description

Author

Ario Amin

6.52 Source/ario_stickyBomb_UE4/Public/UI/FPlayerData.h File Reference

```
#include "CoreMinimal.h"
#include "FPlayerData.generated.h"
```

Classes

- struct FPlayerData

  Simple utility struct to pass data into widgets.

6.52.1 Detailed Description

Author

Ario Amin

6.53 Source/ario_stickyBomb_UE4/Public/UI/SInGameOverlay.h File Reference

```
#include "CoreMinimal.h"
#include <Widgets/SCompoundWidget.h>
```

Classes

- class SInGameOverlay

  A simple shell of a class to be expanded upon. Right now it only funcitons as a blank overlay.

6.53.1 Detailed Description

Author

Ario Amin

6.54 Source/ario_stickyBomb_UE4/Public/UI/StickyHUD.h File Reference

```
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
#include "Helpers/Macros.h"
#include <Components/WidgetComponent.h>
#include <GameFramework/HUD.h>
#include "StickyHUD.generated.h"
```

Classes

- class AStickyHUD

  A simple HUD class to handle widgets.

6.54.1 Detailed Description

Author

Ario Amin

6.55 Source/ario_stickyBomb_UE4/Public/UI/Widgets/SAmmoWidget.h File Reference

```
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
#include <Widgets/DeclarativeSyntaxSupport.h>
#include <Widgets/SCompoundWidget.h>
```

Classes

- class SAmmoWidget

  Ammo Widget which derives from SCompoundWidget.

6.55.1 Detailed Description

Author

Ario Amin

6.56 Source/ario_stickyBomb_UE4/Public/UI/Widgets/SKillContentContainer.h File Reference

```
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
```

Classes

- class SKillContentContainer

  A simple shell of a class to be expanded upon.

6.56.1 Detailed Description

Author

Ario Amin

6.57 Source/ario_stickyBomb_UE4/Public/UI/Widgets/SKillWidget.h File Reference

```
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
#include <Widgets/DeclarativeSyntaxSupport.h>
#include <Widgets/SCompoundWidget.h>
```

Classes

- class SKillWidget

  A simple shell of a class to be expanded upon to display current kills.

6.57.1 Detailed Description

Author

Ario Amin

6.58 Source/ario_stickyBomb_UE4/Public/UI/Widgets/SSlideInText.h File Reference

```
#include "CoreMinimal.h"
#include "Helpers/ForwardDecls.h"
```

Classes

- class SSlideInText

  A simple widget that kan be used to display a recent player kill.

Enumerations

- enum class EVisibleState : uint8 { VS_Animating_To_Show , VS_Animating_To_Hide , VS_Visible , VS_Hidden }

6.58.1 Detailed Description

Author

Ario Amin

6.58.2 Enumeration Type Documentation

6.58.2.1 EVisibleState `enum EVisibleState : uint8 [strong]`

Enumerator

VS_Animating_To_Show	
VS_Animating_To_Hide	
VS_Visible	
VS_Hidden	

6.59 Source/ario_stickyBomb_UE4Editor.Target.cs File Reference

Classes

- class ario_stickyBomb_UE4EditorTarget

Index