Deck 15

Generated by Doxygen 1.8.18

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

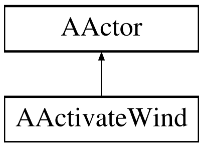
Class Documentation

3.1 AActivateWind Class Reference

This trigger is used to turn on and off the boolean in the wind trigger box that turns onand off the wind mechanic.

```
#include <ActivateWind.h>
```

Inheritance diagram for AActivateWind:



Public Member Functions

- virtual void **Tick** (float DeltaTime) override
- void OnOverlapBegin (class UPrimitiveComponent ∗OverlappedComp, class AActor ∗OtherActor, class U↩ PrimitiveComponent ∗OtherComp, int32 OtherBodyIndex, bool bFromSweep, const FHitResult &Sweep↩ Result)

  Toggles properties when ball overlaps the actor. Sets CanBeHit to false. Toggles associated light's visibility, whether an associated platform is moving and whether an associated elevator can move.

Public Attributes

- bool ToggleTwoWindTriggerBoxes

  boolean to specify how many trigger boxes.
- class UStaticMeshComponent ∗ MyMesh

  Visible mesh representing the actor. Set in UE4 editor.
- class UMaterial ∗ OnMaterial

  Green material should be selected, represents the switch in the on state. Set in UE4 editor.
- class UMaterial ∗ OffMaterial

  Red material should be selected, represents the switch in the off state. Set in UE4 editor.
- class UBoxComponent ∗ MyBoxComponent

Box component for the overlapping section of the switch, where the ball will make contact and toggle switch properties.

- bool OnAtStart = false
- class AWindTriggerBox ∗ AssociatedWindBox

  The WindTriggerBox that needs to be turned on or off.
- class ABall ∗ Ball

  The ball that impact the switch to turn it on or off. Specific ball set in UE4 editor.
- class ARectLight ∗ AssociatedRectlight

  Selected light source to toggle visibility. Set in UE4 editor.
- class AAmbientSound ∗ WindSoundLeft

  left side wind sound
- class AAmbientSound ∗ WindSoundRight

  Right side wind sound.
- FTimerHandle MemberTimerHandle

  timer for the wind force.

Protected Member Functions

- virtual void **BeginPlay** () override

Private Types

- enum SwitchState { **On**, **Off** }

  enum for switches to internally represent whether they are in an on or off state.

Private Member Functions

- void ToggleWind ()

  Called to turn the wind off if it is on.
- void ToggleWindOff ()

  Called to turn the wind on if it is off.
- void ToggleSpotlight ()

  Called to turn the the spotlight on or off.
- void SetCanBeHit ()

  sets CanBeHit variable to true.
- void ToggleWindSound ()

  Called to turn the wind sound on or off.

Private Attributes

- bool CanBeHit

  Variable to decide whether or not this actor can be hit.
- SwitchState CurrentSwitchState = Off

  Current state of the the switch, on of off.
- SwitchState CurrentSpotlightState = Off

  Current state of the light, on or off.
- SwitchState CurrentWindSoundRight = Off

  Current state of the fan in the left room, on or off.
- SwitchState CurrentWindSoundLeft = On

  Current state of the fan in the right room, on or off.

3.1.1 Detailed Description

This trigger is used to turn on and off the boolean in the wind trigger box that turns onand off the wind mechanic.

3.1.2 Member Function Documentation

3.1.2.1 ToggleSpotlight()

```
void AActivateWind::ToggleSpotlight ( )  [private]
```

Called to turn the the spotlight on or off.

3.1.2.2 ToggleWind()

```
void AActivateWind::ToggleWind ( )  [private]
```

Called to turn the wind off if it is on.

3.1.2.3 ToggleWindOff()

```
void AActivateWind::ToggleWindOff ( )  [private]
```

Called to turn the wind on if it is off.

3.1.2.4 ToggleWindSound()

```
void AActivateWind::ToggleWindSound ( )  [private]
```

Called to turn the wind sound on or off.

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

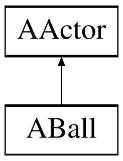
- ActivateWind.h
- ActivateWind.cpp

3.2 ABall Class Reference

The main ball in the game that the player can pick up, throw, summon, and teleport to.

```
#include <Ball.h>
```

Inheritance diagram for ABall:



Public Member Functions

- ABall ()

  Adds static mesh component for the visual aspect, audio component for the crackling sound, and a particle system component to the ball actor.
- virtual void Tick (float DeltaTime) override

  Called every frame.
- bool GetCanBeTeleportedTo ()

  Returns if the Ball can currently be teleported to \reutrns bool the value of CanBeTeleportedTo.
- void SetCanBeTeleportedTo (bool CanTeleportTo)

  Sets if the Ball can currently be teleported to based on the value of CanTeleportTo.
- bool GetHasBeenSummonedOnce ()

  Returns if the Ball has been summoned at least once during the current level.
- void SetHasBeenSummonedOnce (bool HasSummoned)

  Sets the value of bool HasBeenSummoned to HasSummoned.
- bool GetCanBallBeSummoned ()

  Returns if the Ball can currently be summoned by the player.
- void SetCanBallBeSummoned (bool CanBeSummoned)

  sets the value of bool CanBallBeSummoned to CanBeSummoned.

Public Attributes

- UStaticMeshComponent $*$ SphereVisual

  Visual element of the ball actor, set in UE4 editor.
- UParticleSystemComponent $*$ SparksParticleSystem

  Visual sparks the ball emits.
- class UAudioComponent $*$ CrackleSoundPlayer

  Component for playing the crackling sound the ball makes.
- class USoundCue $*$ CrackleSound

  Crackling sound the ball makes. Set in UE4 editor.

Protected Member Functions

- virtual void BeginPlay () override

  Called when the game starts or when spawned, sets CanTeleportToInBeginning. to false.

Protected Attributes

- bool CanTeleportToInBeginning

  default the ball cannot be telported to in beginning.
- bool CanBeTeleportedTo

  Indicates if the ball can currently be teleported to.
- bool HasBeenSummoned

  Indicates if the ball has previously been summoned in this level.
- bool CanBallBeSummoned

  Indicates if the ball can be currently summoned.

3.2.1 Detailed Description

The main ball in the game that the player can pick up, throw, summon, and teleport to.

3.2.2 Member Function Documentation

3.2.2.1 BeginPlay()

```
void ABall::BeginPlay ( )  [override], [protected], [virtual]
```

Called when the game starts or when spawned, sets CanTeleportToInBeginning. to false.

Sets whether the ball can be teleported at the start of the a level.

3.2.2.2 GetCanBallBeSummoned()

```
bool ABall::GetCanBallBeSummoned ( )
```

Returns if the Ball can currently be summoned by the player.

Returns

∗ bool the value of CanBallBeSummoned.

3.2.2.3 GetHasBeenSummonedOnce()

```
bool ABall::GetHasBeenSummonedOnce ( )
```

Returns if the Ball has been summoned at least once during the current level.

Returns

bool the value of HasBeenSummoned

3.2.2.4 SetCanBallBeSummoned()

```
void ABall::SetCanBallBeSummoned (
             bool CanBeSummoned )
```

sets the value of bool CanBallBeSummoned to CanBeSummoned.

Parameters

CanBeSummoned	the new value of CanBallBeSummoned.

3.2.2.5 SetCanBeTeleportedTo()

```
void ABall::SetCanBeTeleportedTo (
            bool CanTeleportTo )
```

Sets if the Ball can currently be teleported to based on the value of CanTeleportTo.

Parameters

CanTeleportTo	the value of the ability to teleport to the ball at the current time, sets CanBeTeleportedTo to the value of CanTeleportTo.

3.2.2.6 SetHasBeenSummonedOnce()

```
void ABall::SetHasBeenSummonedOnce (
            bool HasSummoned )
```

Sets the value of bool HasBeenSummoned to HasSummoned.

Parameters

HasSummoned	the new value of HasBeenSummoned.

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

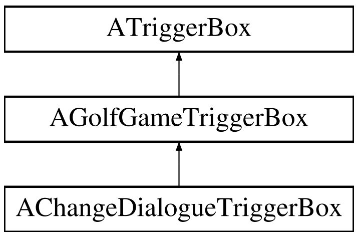
- Ball.h
- Ball.cpp

3.3 AChangeDialogueTriggerBox Class Reference

When the player begins overlapping the ChangeDialogueTriggerBox, a new Sound Base is loaded into the audio component in the GolfGameCharacter and played. A delay can be set from the UE4 editor to postpone the playing of the new dialogue for a set number of seconds. The Sound Base will only be played once, then the DialoguePlayed boolean is set to true.

```
#include <ChangeDialogueTriggerBox.h>
```

Inheritance diagram for AChangeDialogueTriggerBox:



Public Member Functions

- AChangeDialogueTriggerBox ()

  Sets DialoguePlayed to false.
- void DialoguePlay ()

  Play dialogue for player.

Public Attributes

- class USoundBase ∗ Dialogue

  New Sound Base to be loaded to player and player's audio component for dialogue. Sound Base is selected in the UE4 editor.
- class AGolfGameCharacter ∗ PlayerForAudio

  Player selected for which to play the Sound Base for. Player is selected in the UE4 editor.

Protected Member Functions

- virtual void BeginPlay () override

  Called when the game starts or when spawned.
- virtual void OverlapBeginAction () override

  Loads and plays dialogue when overlap begins.
- virtual void OverlapEndAction () override

  Overlap ends. Timer is cleared.

Protected Attributes

- FTimerHandle TimerHandle

  Timer for delaying the loading and playing of a new dialogue cue.
- float TimeToDelayDialogue

  Amount of time in seconds for the dialogue to be delayed before loading and playing. Amount of time is specified in the UE4 editor.

Private Attributes

- bool DialoguePlayed

  Set to true when the dialogue is played, only plays once.

3.3.1 Detailed Description

When the player begins overlapping the ChangeDialogueTriggerBox, a new Sound Base is loaded into the audio component in the GolfGameCharacter and played. A delay can be set from the UE4 editor to postpone the playing of the new dialogue for a set number of seconds. The Sound Base will only be played once, then the DialoguePlayed boolean is set to true.

3.3.2 Member Function Documentation

3.3.2.1 BeginPlay()

```
void AChangeDialogueTriggerBox::BeginPlay ( )  [override], [protected], [virtual]
```

Called when the game starts or when spawned.

\Macro that sets up the class to support the infrastructure required by the engine.

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

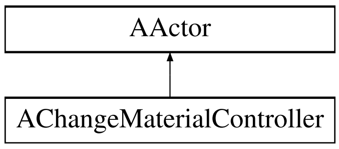
- ChangeDialogueTriggerBox.h
- ChangeDialogueTriggerBox.cpp

3.4 AChangeMaterialController Class Reference

Class controls the functionality behind the in-game switches. Toggles movement of associated elevator or moving platform, switches material of the controller to represent whether it is on or off, and toggles visiblity of an associated light source. The switch can only be hit once per second by the ball as to not have false positive hits.

```
#include <ChangeMaterialController.h>
```

Inheritance diagram for AChangeMaterialController:



Public Member Functions

- AChangeMaterialController ()

  Sets up static mesh component for the visual aspect of the actor. Adds materials for on and off states. Sets up box component for impact of the ball to toggle associated actor properties.
- virtual void Tick (float DeltaTime) override

  Called every frame.
- void OnOverlapBegin (class UPrimitiveComponent ∗OverlappedComp, class AActor ∗OtherActor, class U↩ PrimitiveComponent ∗OtherComp, int32 OtherBodyIndex, bool bFromSweep, const FHitResult &Sweep↩ Result)

  Toggles properties when ball overlaps the actor. Sets CanBeHit to false. Toggles associated light's visibility, whether an associated platform is moving and whether an associated elevator can move.

Public Attributes

- class UStaticMeshComponent ∗ MyMesh

  Visible mesh representing the actor. Set in UE4 editor.
- class UMaterial ∗ OnMaterial

  Green material should be selected, represents the switch in the on state. Set in UE4 editor.
- class UMaterial ∗ OffMaterial

  Red material should be selected, represents the switch in the off state. Set in UE4 editor.
- class UBoxComponent ∗ MyBoxComponent

  Box component for the overlapping section of the switch, where the ball will make contact and toggle switch properties.
- class APlatform_Moving ∗ AssociatedPlatform

  Represents the moving platform whose movement is toggled on or off. Associated platform set in UE4 editor.
- class ABall ∗ Ball

  The ball that impact the switch to turn it on or off. Specific ball set in UE4 editor.
- class ARectLight ∗ AssociatedRectlight

  Selected light source to toggle visibility. Set in UE4 editor.
- class AElevatorPlatform ∗ AssociatedElevator

  Selected elevator to toggle active or inactive. Set in UE4 editor.

Protected Member Functions

- virtual void BeginPlay () override

  Called when the game starts or when spawned. Sets material to OffMaterial. Sets CanBeHit to true so that is can toggle properties when hit. Sets timer for resetting CanBeHit.

Private Types

- enum SwitchState { **On**, **Off** }

  enum for switches to internally represent whether they are in an on or off state.

Private Member Functions

- void TogglePlatformMovement ()

  Toggles a moving platform's movement to on or off.
- void ToggleSpotlight ()

  Toggles visibility of selected light source.
- void ToggleElevator ()

  Toggles a selected elevator to active or inactive.
- void SetCanBeHit ()

  sets CanBeHit variable to true.
- void CyclePlatformMovingAudio ()

  Intended to toggle audio component sound on or off, unfinished.

Private Attributes

- SwitchState CurrentSwitchState = Off

    Current state of the the switch, on of off.
- SwitchState CurrentSpotlightState = Off

    Current state of the light, on or off.
- SwitchState CurrentMovingAudio = Off

    Current state of the audio component.
- bool CanBeHit

    Whether the switch can currently be hit by the ball to toggle associated actors.
- FTimerHandle MemberTimerHandle

    Timer for when the switch can be hit by the ball.

3.4.1 Detailed Description

Class controls the functionality behind the in-game switches. Toggles movement of associated elevator or moving platform, switches material of the controller to represent whether it is on or off, and toggles visiblity of an associated light source. The switch can only be hit once per second by the ball as to not have false positive hits.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 AChangeMaterialController()

```
AChangeMaterialController::AChangeMaterialController ( )
```

Sets up static mesh component for the visual aspect of the actor. Adds materials for on and off states. Sets up box component for impact of the ball to toggle associated actor properties.

\Macro that sets up the class to support the infrastructure required by the engine.

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

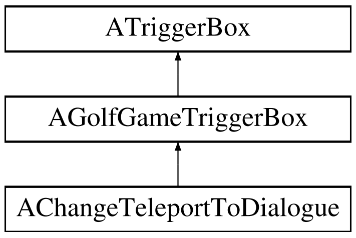
- ChangeMaterialController.h
- ChangeMaterialController.cpp

3.5 AChangeTeleportToDialogue Class Reference

Loads in new dialogue cue for when the player cannot teleport to the ball.

```
#include <ChangeTeleportToDialogue.h>
```

Inheritance diagram for AChangeTeleportToDialogue:



Public Member Functions

- virtual void OverlapBeginAction () override

    Loads new dialogue cue into player when overlap begins.
- virtual void OverlapEndAction () override

    Overlap ends.

Public Attributes

- class USoundBase ∗ Dialogue

    Macro that sets up the class to support the infrastructure required by the engine.
- class AGolfGameCharacter ∗ **Player**

Additional Inherited Members

3.5.1 Detailed Description

Loads in new dialogue cue for when the player cannot teleport to the ball.

3.5.2 Member Data Documentation

3.5.2.1 Dialogue

```
class USoundBase* AChangeTeleportToDialogue::Dialogue
```

Macro that sets up the class to support the infrastructure required by the engine.

New dialogue cue to be loaded into the player's audio component for dialogue.

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

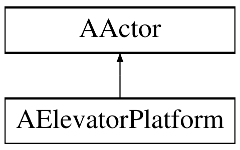
- ChangeTeleportToDialogue.h
- ChangeTeleportToDialogue.cpp

3.6 AElevatorPlatform Class Reference

Class communicates with the ElevatorTrigger trigger box that checks for the presence of the player character and moves up when the character enters the box. If the character exits the box before it reaches the target height, it moves back down to its original position. The associated static mesh, trigger box, and desired speed of movement and height of path can all be set inside the Unreal editor.

```
#include <ElevatorPlatform.h>
```

Inheritance diagram for AElevatorPlatform:



Public Member Functions

- AElevatorPlatform ()

  The class's constructor, which activates the actor's tick function,
  sets up the static mesh component, and initializes variables.
- virtual void Tick (float DeltaTime) override

Public Attributes

- UStaticMeshComponent ∗ Mesh

  Visible mesh representing the actor. Set in UE4 editor.
- class AElevatorTrigger ∗ Trigger

  The trigger box responsible for detecting the presence of the player character and activating the platform's movement accordingly.
- float Speed

  Determines the speed at which the platform can move.
- float TargetZ

  The target height that the platform can rise to.
- bool ElevatorActive

  Boolean values determining whether the elevator is active, whether it should move up or down, and whether it has reached its target height.
- bool **MoveUp**
- bool **MoveDown**
- bool **ReachedTarget**
- float OriginalZ

  The z value obtained from the original location vector of the platform at the start of the game.

Protected Member Functions

- virtual void BeginPlay () override

  Called when the game starts or when spawned. Stores the Z value from the vector containing the initial location of the platform.

3.6.1 Detailed Description

Class communicates with the ElevatorTrigger trigger box that checks for the presence of the player character and moves up when the character enters the box. If the character exits the box before it reaches the target height, it moves back down to its original position. The associated static mesh, trigger box, and desired speed of movement and height of path can all be set inside the Unreal editor.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 AElevatorPlatform()

```
AElevatorPlatform::AElevatorPlatform ( )
```

The class's constructor, which activates the actor's tick function,
sets up the static mesh component, and initializes variables.

\Macro that sets up the class to support the infrastructure required by the engine.

3.6.3 Member Function Documentation

3.6.3.1 Tick()

```
void AElevatorPlatform::Tick (
             float DeltaTime )  [override], [virtual]
```

\ Called every frame. Checks whether the platform should be moving up or down, or if it has reached its target height.

Parameters

DeltaTime	The elapsed time since the last tick.

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

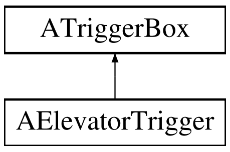
- ElevatorPlatform.h
- ElevatorPlatform.cpp

3.7 AElevatorTrigger Class Reference

Class responsible for communicating to the ElevatorPlatform whether the player character is present.

```
#include <ElevatorTrigger.h>
```

Inheritance diagram for AElevatorTrigger:



Public Member Functions

- AElevatorTrigger ()

  The class constructor. Uses helper macros to enable the trigger box to use the functions OnOverlapBegin and On←OverlapEnd, which make the box execute certain actions whenever an object overlaps or stops overlapping with the box.
- virtual void Tick (float DeltaTime) override
- void OnOverlapBegin (class AActor ∗OverlappedActor, class AActor ∗OtherActor)

  Checks whether the overlapped character is the player character, whether the ElevatorPlatform is active, and then tells the platform to move up.
- void OnOverlapEnd (class AActor ∗OverlappedActor, class AActor ∗OtherActor)

  Checks whether the overlapped character is the player character, whether the ElevatorPlatform is active, and then tells the platform to move down.

Public Attributes

- class AElevatorPlatform ∗ ElevatorPlatform

  The associated ElevatorPlatform with which the ElevatorTrigger communicates.

Protected Member Functions

- virtual void BeginPlay () override

  Called when the game starts or when spawned.

3.7.1 Detailed Description

Class responsible for communicating to the ElevatorPlatform whether the player character is present.

3.7.2 Member Function Documentation

3.7.2.1 BeginPlay()

```
void AElevatorTrigger::BeginPlay ( )  [override], [protected], [virtual]
```

Called when the game starts or when spawned.

\Macro that sets up the class to support the infrastructure required by the engine.

3.7.2.2 OnOverlapBegin()

```
void AElevatorTrigger::OnOverlapBegin (
             class AActor ∗ OverlappedActor,
             class AActor ∗ OtherActor )
```

Checks whether the overlapped character is the player character, whether the ElevatorPlatform is active, and then tells the platform to move up.

Parameters

OverlappedActor	A reference to this trigger box, which is being overlapped.
OtherActor	A reference to the actor that overlapped the trigger box.

3.7.2.3 OnOverlapEnd()

```
void AElevatorTrigger::OnOverlapEnd (
            class AActor * OverlappedActor,
            class AActor * OtherActor )
```

Checks whether the overlapped character is the player character, whether the ElevatorPlatform is active, and then tells the platform to move down.

Parameters

OverlappedActor	A reference to this trigger box, which is being overlapped.
OtherActor	A reference to the actor that overlapped the trigger box.

3.7.2.4 Tick()

```
void AElevatorTrigger::Tick (
            float DeltaTime )  [override], [virtual]
```

\ Called every frame.

Parameters

DeltaTime	The elapsed time since the last tick.

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

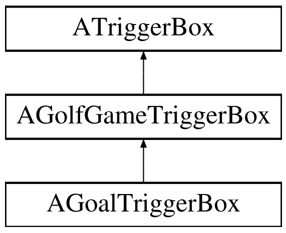
- ElevatorTrigger.h
- ElevatorTrigger.cpp

3.8 AGoalTriggerBox Class Reference

A trigger box that indicates if the Ball is in this trigger box or not.

`#include <GoalTriggerBox.h>`

Inheritance diagram for AGoalTriggerBox:



Public Member Functions

- virtual void OverlapBeginAction () override

  Disables the ability to teleport and summon the ball when the ball is no longer overlapping this trigger box.
- virtual void OverlapEndAction () override

  Enables the ability to teleport and summon the ball when the ball is no longer overlapping this trigger box.
- bool GetIsBallInGoal ()

  Returns true if the Ball is in the goal trigger box, returns false otherwise.
- void SetIsBallInGoal (bool BallInGoal)

  Sets the value of IsBallInGoal to the value in BallInGoal.

Protected Member Functions

- virtual void BeginPlay () override

  Calls super BeginPlay() and casts ActorToCheck (from super GolfGameTriggerBox) to Ball reference.

Protected Attributes

- AActor ∗ DoorTarget

  Reference to DoorTarget to open or close if necessary.
- AActor ∗ DoorLight

  Reference to a door light.
- AActor ∗ GoalLight

  Reference to a goal light.
- bool IsBallInGoal

  Indicates if the ball is in the goal or not.

Private Attributes

- ABall ∗ Ball

  Reference for the in game ball.

3.8.1 Detailed Description

A trigger box that indicates if the Ball is in this trigger box or not.

Goal trigger box that stops the ball from being summoned or teleported to when overlapped, and allows the ball to be summoned or teleported to when the ball stops overlapping this trigger box.

3.8.2 Member Function Documentation

3.8.2.1 GetIsBallInGoal()

```
bool AGoalTriggerBox::GetIsBallInGoal ( )
```

Returns true if the Ball is in the goal trigger box, returns false otherwise.

Returns

bool the value of whether or not the ball is overlapping this trigger box

3.8.2.2 SetIsBallInGoal()

```
void AGoalTriggerBox::SetIsBallInGoal (
             bool BallInGoal )
```

Sets the value of IsBallInGoal to the value in BallInGoal.

Parameters

BallInGoal	the new value to set IsBallInGoal to

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

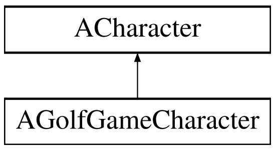
- GoalTriggerBox.h
- GoalTriggerBox.cpp

3.9 AGolfGameCharacter Class Reference

GolfGameCharacter houses all functionality for a user to operate a player in-game.

```
#include <GolfGameCharacter.h>
```

Inheritance diagram for AGolfGameCharacter:



Classes

- struct FTouchData

Public Member Functions

- AGolfGameCharacter ()

  Macro that sets up the class to support the infrastructure required by the engine.
- FORCEINLINE class UCameraComponent ∗ GetFirstPersonCameraComponent () const

  Returns FirstPersonCameraComponent subobject.
- void PlayDialogueCue ()

  current dialogue cue for the player in the audio component for dialogue
- void PlayMusicCue ()

  Plays the current music cue for the player in the audio component for music.
- void AdjustMusicVolumeUp ()

  Adjusts volume up when dialogue cue ends, not used.
- void AdjustMusicVolumeDown ()

  Adjusts volume down when dialogue cue starts, not used.
- void ChangeDialogueCue (USoundBase ∗NewDialogue)

  Swaps out old dialogue cue for new cue when overlapping a dialogue change trigger box.
- void ChangeMusicCue (USoundBase ∗NewMusic)

  Swaps out old music cue for new cue when overlapping music cue change trigger box.

Public Attributes

- class UCameraComponent ∗ FirstPersonCameraComponent

  First person camera.
- class UGrabThrowComponent ∗ GrabberClass

  Grabber class.
- class UPhysicsHandleComponent ∗ PhysicsHandle

  PhysicsHandle class.
- class USphereComponent ∗ BallSummonLocation

  The location to which the ball should be summoned.
- class ABall ∗ Ball

  Reference to the in-game ball for picking up, throwing, dropping, summoning, and teleporting to.
- class UAudioComponent ∗ DialoguePlayer

  Audio component attached to the player to play dialogue cues.
- class UAudioComponent ∗ MusicPlayer

  Audio component attached to the player to play music cues.
- class USoundBase ∗ CurrentMusicCue

  Current sound base for music.
- class USoundBase ∗ CannotSummonBallCue

  Current dialogue cue for when the ball cannot be summoned.
- float BaseTurnRate

  Base turn rate, in deg/sec. Other scaling may affect final turn rate.
- float BaseLookUpRate

  Base look up/down rate, in deg/sec. Other scaling may affect final rate.
- float BaseSpeed

  The regular walking speed of the character.
- float RunningSpeed

  The running speed of the character.
- FVector CameraPosition

  The position of the camera relative to the golf game character.
- class USoundBase ∗ NeedToTeleportBallCue

Current sound cue for when the player cannot be teleported to the ball. Selected in UE4 editor.

- class USoundBase ∗ TeleportSound

  Current sound cue for when the player is teleported to the ball. Selected in UE4 editor.
- class USoundBase ∗ SummonSound

  Current sound cue for when the player summons the ball. Selected in UE4 editor.

Protected Member Functions

- void BeginPlay () override

  Called when the game starts or when spawned.
- void MoveForward (float Value)
- void MoveRight (float Value)

  Moves the character to the left or the right.
- void GrabOrRelease ()

  Based on whether the character is currently holding an object, this method tries to either pick up an object in front of the character or drop the one currently being held.
- void MouseDown ()

  If the character is holding an object, this method tries to throw the object in the direction the character is facing.
- void Walk ()

  Toggles the character's movement speed to the base walking speed.
- void Sprint ()

  Toggles the character's moving speed to the running speed.
- void TurnAtRate (float Rate)

  Called via input to turn at a given rate.
- void LookUpAtRate (float Rate)

  Called via input to turn look up/down at a given rate.
- void Tick (float DeltaTime) override
- void SetupPlayerInputComponent (UInputComponent ∗PlayerInputComponent) override
- void Teleport ()

  Attempts to set the player's location to the ball's location.
- void SummonBall ()

  Attempts to set the ball's location to the player's location and has the player hold the ball if successful.

Protected Attributes

- FTouchData **TouchItem**

Private Attributes

- class USoundBase ∗ CurrentDialogueCue

  Current sound cue for dialogue.

3.9.1 Detailed Description

GolfGameCharacter houses all functionality for a user to operate a player in-game.

GolfGameCharacter encapsulates a camera component for the player to view the game from, a GrabThrow component to interact with the ball (pick up, drop, and throw), audio components for playing in-game music and dialogue cues, the ability to summon the ball and teleport to it, as well as necessary components for in-game operations.

3.9.2 Constructor & Destructor Documentation

3.9.2.1 AGolfGameCharacter()

```
AGolfGameCharacter::AGolfGameCharacter ( )
```

Macro that sets up the class to support the infrastructure required by the engine.

Sets default values for this character's properties

3.9.3 Member Function Documentation

3.9.3.1 LookUpAtRate()

```
void AGolfGameCharacter::LookUpAtRate (
             float Rate )  [protected]
```

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

3.9.3.2 MoveForward()

```
void AGolfGameCharacter::MoveForward (
             float Value )  [protected]
```

\ Moves the character forward or backward.

Parameters

Value	The value used for determining the rate at which the character moves.

3.9.3.3 MoveRight()

```
void AGolfGameCharacter::MoveRight (
             float Value )  [protected]
```

Moves the character to the left or the right.

Parameters

Value	The value used for determining the rate at which the character moves.

3.9.3.4 SetupPlayerInputComponent()

```
void AGolfGameCharacter::SetupPlayerInputComponent (
            UInputComponent * PlayerInputComponent )  [override], [protected]
```

\ Sets up the inputs usable by the character.

Parameters

PlayerInputComponent	The component belonging to the player character that allows the use of inputs.

3.9.3.5 Tick()

```
void AGolfGameCharacter::Tick (
            float DeltaTime )  [override], [protected]
```

\ Called every frame.

Parameters

DeltaTime	The elapsed time since the last tick.

3.9.3.6 TurnAtRate()

```
void AGolfGameCharacter::TurnAtRate (
            float Rate )  [protected]
```

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

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

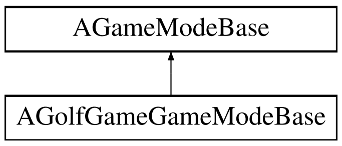
- GolfGameCharacter.h
- GolfGameCharacter.cpp

3.10 AGolfGameGameModeBase Class Reference

Class generated by Ue4 when the project was created. Not directly utilized.

```
#include <GolfGameGameModeBase.h>
```

Inheritance diagram for AGolfGameGameModeBase:



Public Member Functions

- void **EndGame** ()
- void **LevelComplete** ()
- void **LoadNextLevel** ()

Private Member Functions

- void **BeginPlay** () override
- void **CheckLevel** ()

Private Attributes

- TArray< FString > **Levels**
- APlayerController ∗ **Controller**
- int32 **CurrentLevel**
- FString **NextLevel**

3.10.1 Detailed Description

Class generated by Ue4 when the project was created. Not directly utilized.

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

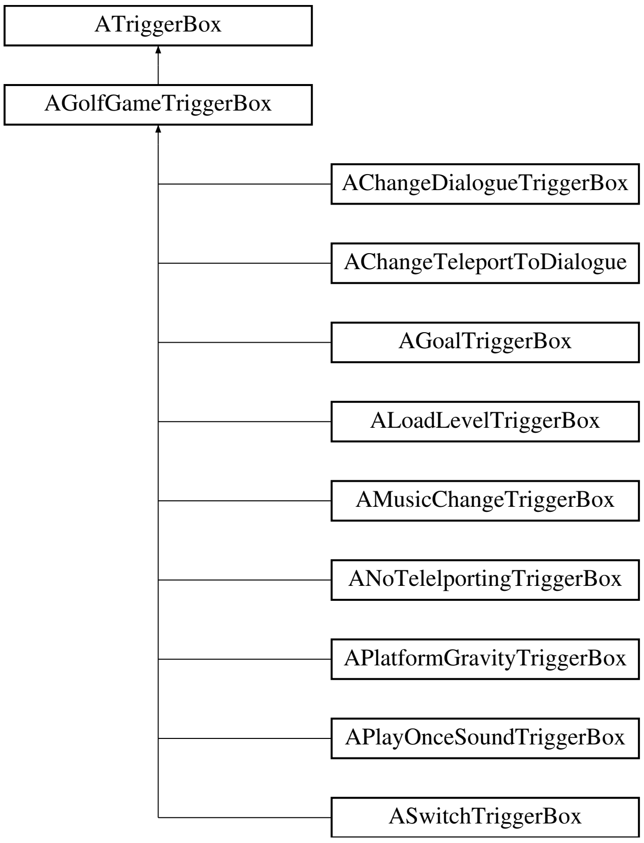
- GolfGameGameModeBase.h
- GolfGameGameModeBase.cpp

3.11 AGolfGameTriggerBox Class Reference

Abstract trigger box that calls a method when a specific actor begins or stops overlapping this trigger box.

```
#include <GolfGameTriggerBox.h>
```

Inheritance diagram for AGolfGameTriggerBox:



Public Member Functions

- AGolfGameTriggerBox ()

  Binds the function OnOverlapBegin to the delegate OnActorBeginOverlap and the function OnOverlapEnd to the delegate OnActorEndOverlap.
- void OnOverlapBegin (class AActor ∗OverlappedActor, class AActor ∗OtherActor)

  Checks to see if the Actor beginning to overlap is the Actor referenced in ActorToCheck the function to call is handled in the subclass.
- void OnOverlapEnd (class AActor ∗OverlappedActor, class AActor ∗OtherActor)

  Checks to see if the Actor ending the overlap is the Actor referenced in ActorToCheck the function to call is handled in the subclass.

Protected Attributes

- AActor ∗ ActorToCheck

  Reference to the Actor that must be set to determine if said Actor is overlapping this class or not.

3.11.1 Detailed Description

Abstract trigger box that calls a method when a specific actor begins or stops overlapping this trigger box.

3.11.2 Member Function Documentation

3.11.2.1 OnOverlapBegin()

```
void AGolfGameTriggerBox::OnOverlapBegin (
             class AActor * OverlappedActor,
             class AActor * OtherActor )
```

Checks to see if the Actor beginning to overlap is the Actor referenced in ActorToCheck the function to call is handled in the subclass.

Parameters

OverlappedActor	required param in FActorBeginOverlapSignature delegate.
OtherActor	the actor that overlaps this trigger box.

3.11.2.2 OnOverlapEnd()

```
void AGolfGameTriggerBox::OnOverlapEnd (
             class AActor * OverlappedActor,
             class AActor * OtherActor )
```

Checks to see if the Actor ending the overlap is the Actor referenced in ActorToCheck the function to call is handled in the subclass.

Parameters

OverlappedActor	required param in FActorBeginOverlapSignature delegate.
OtherActor	the actor that has stopped overlapping this trigger box.

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

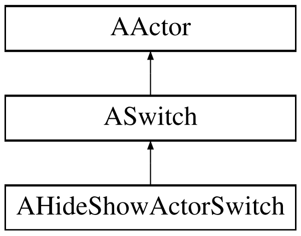
- GolfGameTriggerBox.h
- GolfGameTriggerBox.cpp

3.12 AHideShowActorSwitch Class Reference

Switches a list of actors on and off.

```
#include <HideShowActorSwitch.h>
```

Inheritance diagram for AHideShowActorSwitch:



Public Member Functions

- virtual void ActionOn () override

  The action that will be preformed when this switch is switched on, determined by the "On" variable.
- virtual void ActionOff () override

  The action that will be preformed when this switch is switched off, determined by the "Off" variable.
- virtual uint8 GetActionOff () override

  Returns the "Off" variable.
- virtual uint8 GetActionOn () override

  Returns the "On" variable.
- virtual void SetActionOn (uint8 Status) override

  Sets the "On" variable.
- virtual void SetActionOff (uint8 Status) override

  Sets the "Off" variable.

Protected Member Functions

- void HideActors (bool Show)

  Hides the actors in the ActorsToShowHide array if Show is true, otherwise shows the actors.
- void FlickerInAndOut ()

  Iterates over the FlickerPattern array at a rate based on the FlickerRate variable.
- virtual void BeginPlay () override

  Called when the game starts or when spawned, initializes some fields.

Protected Attributes

- ActorHideShowStatus On

  Indicates what action to preform when this class is switched on: hide, show, or flicker actors.
- ActorHideShowStatus Off

  Indicates what action to preform when this class is switched off: hide, show, or flicker actors.
- TArray< AActor ∗ > ActorsToShowHide

  Holds a list of actors to be switch on and off.
- TArray< ActorHideShowStatus > FlickerPattern

  An array that holds the pattern that will be flickered.
- int TimeHidden

  Indicates the time the actor will spend hidden when flickering in and out.
- int TimeShown

  Indicates the time the actor will spend shown when flickering in and out.
- float TimeFlickering

  Indicates the time the actor will spend flickering when flickering in and out.
- float FlickerRate

  Indicates how often the FlickerInAndOut method will be called.

Private Member Functions

- void IncrementFlickerIndex ()

  Increments the index of the FlickerPattern array.
- void SetCurrentRunTime ()

  Sets CurrentRunTime based on the FlickerIndex value.
- virtual void GetMethodToCall (uint8 Status) override

  Gets the method to call when this switch is switched on or off. The method to call is determined by the Status.

Private Attributes

- int CurrentRunTime

  Indicates how long the current index of the FlickerPattern should run.
- bool FirstTimeFlickerCalled

  Indicates if this is the first time the FlickerInAndOut method has been called.
- bool HasBeenIncremented

  Indicates if the index of FlickerPattern has already been incremented during the indexes run time.
- int FlickerIndex

  The index of the FlickerPattern.
- int LastCalled

  The last time the world time was called, used to determine the Timer time and reset it.
- int Timer

  Indicates how long the current FlickerPattern index has been running.
- bool AreActorsHidden

  Indicates whether the actors are hidden or not.

Additional Inherited Members

3.12.1 Detailed Description

Switches a list of actors on and off.

An actor switch that contains an array of actors and preforms an action when turned off or on. The actions that can be preformed include: hide, show, flicker actors, or do nothing. When turned off or on, the array is iterated over and the defined "On" or "Off" action is preformed on all actors in the array.

3.12.2 Member Function Documentation

3.12.2.1 GetActionOff()

```
uint8 AHideShowActorSwitch::GetActionOff ( )  [override], [virtual]
```

Returns the "Off" variable.

Returns

uint8 value of the "Off" variable.

3.12.2.2 GetActionOn()

```
uint8 AHideShowActorSwitch::GetActionOn ( )  [override], [virtual]
```

Returns the "On" variable.

Returns

uint8 value of the "On" variable.

3.12.2.3 GetMethodToCall()

```
void AHideShowActorSwitch::GetMethodToCall (
             uint8 Status )  [override], [private], [virtual]
```

Gets the method to call when this switch is switched on or off. The method to call is determined by the Status.

Parameters

Status	the GolfGameEnum that determines what method to call.

3.12.2.4 HideActors()

```
void AHideShowActorSwitch::HideActors (
             bool Show )  [protected]
```

Hides the actors in the ActorsToShowHide array if Show is true, otherwise shows the actors.

Parameters

Show	will hide actors if true, will show if false.

3.12.2.5 SetActionOff()

```
void AHideShowActorSwitch::SetActionOff (
             uint8 Status )  [override], [virtual]
```

Sets the "Off" variable.

Parameters

Status	the GolfGameEnum that will be the new value of the "Off" variable.

3.12.2.6 SetActionOn()

```
void AHideShowActorSwitch::SetActionOn (
            uint8 Status )  [override], [virtual]
```

Sets the "On" variable.

Parameters

Status	the GolfGameEnum that will be the new value of the "On" variable.

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

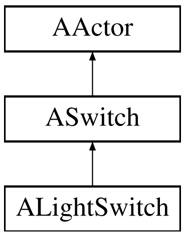
- HideShowActorSwitch.h
- HideShowActorSwitch.cpp

3.13 ALightSwitch Class Reference

Switches a list of lights on and off.

`#include <LightSwitch.h>`

Inheritance diagram for ALightSwitch:



Public Member Functions

- virtual void ActionOn () override

  The action that will be preformed when this switch is switched on, determined by the "On" variable.
- virtual void ActionOff () override

  The action that will be preformed when this switch is switched off, determined by the "Off" variable.
- virtual uint8 GetActionOff () override

  Returns the "Off" variable.
- virtual uint8 GetActionOn () override

  Returns the "On" variable.
- virtual void SetActionOn (uint8 Status) override

  Sets the "On" variable.
- virtual void SetActionOff (uint8 Status) override

  Sets the "Off" variable.

Protected Member Functions

- void FlickerLights ()
  Flickers the lights.
- void DimLights (float DimVar)
  Dims the lights to the light intensity based on the variable DimVar.
- void HideLights (bool Show)
  Hides the lights in the Lights array if Show is true, otherwise shows the Lights.

Protected Attributes

- LightStatus On
  Indicates what action to preform when this class is switched on: hide, show, flicker, dim lights, or do nothing.
- LightStatus Off
  Indicates what action to preform when this class is switched off: hide, show, flicker, dim lights, or do nothing.
- float FlickerRate
  Indicates the rate at which the lights will flicker.
- float FlickerBrighten
  The max value to increase the light brightness to when flickering.
- float FlickerDim
  The min value to decrease the light brightness to when flickering.
- float Dim
  The value of the light brightness to decrease the lights to.
- TArray$<$ ALight $*$ $>$ Lights
  The array of lights for this light switch.

Private Member Functions

- virtual void GetMethodToCall (uint8 Status) override
  Gets the method to call when this switch is switched on or off.

Private Attributes

- bool AreLightsHidden
  Indicates if the lights are currently hidden (off) or not.
- bool AreLightsDimmed
  Indicates if the lights are currently dimmed or not.

Additional Inherited Members

3.13.1 Detailed Description

Switches a list of lights on and off.

A light switch that contains an array of lights and preforms an action when turned off or on.The actions that can be preformed include : hide, show, flicker, dim lights, or do nothing.When turned off or on, the array is iterated overand the defined "On" or "Off" action is preformed on all lights in the array.

3.13.2 Member Function Documentation

3.13.2.1 DimLights()

```
void ALightSwitch::DimLights (
             float DimVar )  [protected]
```

Dims the lights to the light intensity based on the variable DimVar.

Parameters

DimVar	the new light intensity to set the lights to.

3.13.2.2 FlickerLights()

```
void ALightSwitch::FlickerLights ( )  [protected]
```

Flickers the lights.

FlickerLights() will switch between changing the light intensity to FlickerBrighten and FlickerDim at a rate based on the FlickerRate.

3.13.2.3 GetActionOff()

```
uint8 ALightSwitch::GetActionOff ( )  [override], [virtual]
```

Returns the "Off" variable.

Returns

uint8 value of the "Off" variable

3.13.2.4 GetActionOn()

```
uint8 ALightSwitch::GetActionOn ( )  [override], [virtual]
```

Returns the "On" variable.

Returns

uint8 value of the "On" variable.

3.13.2.5 GetMethodToCall()

```
void ALightSwitch::GetMethodToCall (
             uint8 Status )  [override], [private], [virtual]
```

Gets the method to call when this switch is switched on or off.

The method to call is determined by the Status UEnum. The method will either call Flicker(), HideLights(bool Show), or DimLights(float DimVar) depending on the UEnum passed.

Parameters

Status	the GolfGameEnum that determines what method to call.

3.13.2.6 HideLights()

```
void ALightSwitch::HideLights (
            bool Show )  [protected]
```

Hides the lights in the Lights array if Show is true, otherwise shows the Lights.

Parameters

Show	will hide the Lights if true, will show Lights if false.

3.13.2.7 SetActionOff()

```
void ALightSwitch::SetActionOff (
            uint8 Status )  [override], [virtual]
```

Sets the "Off" variable.

Parameters

Status	the GolfGameEnum that will be the new value of the "Off" variable.

3.13.2.8 SetActionOn()

```
void ALightSwitch::SetActionOn (
            uint8 Status )  [override], [virtual]
```

Sets the "On" variable.

Parameters

Status	the GolfGameEnum that will be the new value of the "On" variable.

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

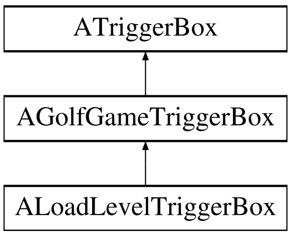
- LightSwitch.h
- LightSwitch.cpp

3.14 ALoadLevelTriggerBox Class Reference

Trigger box that loads the next level when overlapped.

```
#include <LoadLevelTriggerBox.h>
```

Inheritance diagram for ALoadLevelTriggerBox:



Public Member Functions

- void LevelLoad ()

  Loads the level specified by LevelToLoad.
- virtual void OverlapBeginAction () override

  Sets IsCharacterInTriggerBox to true once the ActorToCheck overlaps this trigger box.

Protected Attributes

- bool RollCredits

  Macro that sets up the class to support the infrastructure required by the engine.
- float LevelLoadTimeDelay

  The amount of time to wait for the next level to load.
- FName LevelToLoad

  The name of the next level to load.
- bool IsCharacterInTrigger

  Indicates whether the ActorToCheck is in the trigger box or not.

Private Member Functions

- virtual void OverlapEndAction () override

  Inherited method that is not implemented.

3.14.1 Detailed Description

Trigger box that loads the next level when overlapped.

Abstract trigger box that loads the designated next level when overlapped by the referenced ActorToCheck (from super GolfGameTriggerBox).

3.14.2 Member Data Documentation

3.14.2.1 RollCredits

```
bool ALoadLevelTriggerBox::RollCredits [protected]
```

Macro that sets up the class to support the infrastructure required by the engine.

Determines if end credits should play or the loading screen.

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

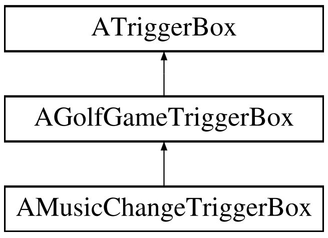
- LoadLevelTriggerBox.h
- LoadLevelTriggerBox.cpp

3.15 AMusicChangeTriggerBox Class Reference

When a specific actor begins to overlap MusicChangeTriggerBox, a selected Sound Base is loaded into the audio component for music in the GolfGameCharacter and played.

```
#include <MusicChangeTriggerBox.h>
```

Inheritance diagram for AMusicChangeTriggerBox:



Public Member Functions

- AMusicChangeTriggerBox ()

  Stock constructor.
- virtual void OverlapBeginAction () override

  Upon beginning the overlap, the new music cue is loaded to the player and is played.
- virtual void OverlapEndAction () override

  Trigger box overlap end.

Public Attributes

- class USoundBase ∗ Music

  Sound base for current music cue. Set in UE4 editor.
- class AGolfGameCharacter ∗ PlayerForAudio

  Reference to the player so that the new music cue can be loaded into the audio component for music. Set in UE4 editor.

Protected Member Functions

- virtual void BeginPlay () override

  Macro that sets up the class to support the infrastructure required by the engine.

Private Attributes

- bool MusicStarted = false

  Represents whether the music cue has already been loaded and played.

Additional Inherited Members

3.15.1 Detailed Description

When a specific actor begins to overlap MusicChangeTriggerBox, a selected Sound Base is loaded into the audio component for music in the GolfGameCharacter and played.

3.15.2 Member Function Documentation

3.15.2.1 BeginPlay()

```
void AMusicChangeTriggerBox::BeginPlay ( )  [override], [protected], [virtual]
```

Macro that sets up the class to support the infrastructure required by the engine.

Called when the game starts or when spawned.

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

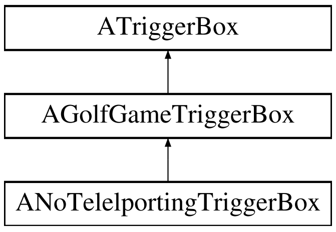
- MusicChangeTriggerBox.h
- MusicChangeTriggerBox.cpp

3.16 ANoTelelportingTriggerBox Class Reference

Disables the player from teleporting to the in game ball when it is in this trigger box.

```
#include <NoTelelportingTriggerBox.h>
```

Inheritance diagram for ANoTelelportingTriggerBox:



Public Member Functions

- virtual void OverlapBeginAction () override

    Disables the ability to teleport to the ball when the ball overlaps this trigger box.
- virtual void OverlapEndAction () override

    Enables the ability to teleport to the ball when the ball no longer overlaps this trigger box.

Protected Member Functions

- virtual void BeginPlay () override

    Calls super BeginPlay() and casts ActorToCheck (from super GolfGameTriggerBox) to Ball reference.

Private Attributes

- ABall ∗ Ball

    Reference to the ball in the game.

Additional Inherited Members

3.16.1 Detailed Description

Disables the player from teleporting to the in game ball when it is in this trigger box.

Trigger box that prevents the player from teleporting to the area this trigger box covers when the ball is overlapping this trigger box. Enables teleportation when the ball is no longer overlapping.

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

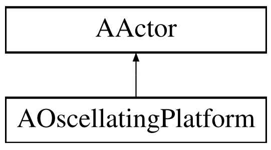
- NoTelelportingTriggerBox.h
- NoTelelportingTriggerBox.cpp

3.17 AOscellatingPlatform Class Reference

Class creates a platform that moves back and forth on the x, y, or z axis. The associated static mesh, amount of pause time on either end of the path, the length of the path, speed of movement, initial direction, and axis of movement can all be set inside the Unreal editor.

```
#include <OscellatingPlatform.h>
```

Inheritance diagram for AOscellatingPlatform:



Public Member Functions

- AOscellatingPlatform ()

  The constructor of the class, which activates the actor's ability to tick, creates a static mesh, and initializes all variables.
- void Tick (float DeltaTime) override

  Called every frame.

Public Attributes

- UStaticMeshComponent ∗ Mesh

  Visible mesh representing the actor. Set in UE4 editor.
- int PauseTime

  The time that the platform doesn't move when it reaches either end of its path. Set in UE4 editor.
- float PathHeight

  The length of the platform's path. Set in UE4 editor.
- float Speed

  The speed at which the platform moves. Set in UE4 editor.
- int Direction

  The initial direction that the platform will be moving along its path. Set in UE4 editor.
- FString Axis

  the axis along which the platform moves. Set in UE4 editor.
- bool Paused

  Whether the platform is currently paused.
- int CurTime

  The length of time left for the platform to be paused.
- float MaxX

  Calculated max X value of the platform.
- float MaxY

  Calculated max Y value of the platform.
- float MaxZ

  Calculated max Z value of the platform.
- float OriginalX

  Platform's orginal X location.
- float OriginalY

  Platform's orginal Y location.
- float OriginalZ

  Platform's orginal Z location.

Protected Member Functions

- void BeginPlay () override

  Called when the game starts or when spawned. Stores the values of the vector of the original location and calculates the max using those values.

3.17.1 Detailed Description

Class creates a platform that moves back and forth on the x, y, or z axis. The associated static mesh, amount of pause time on either end of the path, the length of the path, speed of movement, initial direction, and axis of movement can all be set inside the Unreal editor.

3.17.2 Constructor & Destructor Documentation

3.17.2.1 AOscellatingPlatform()

```
AOscellatingPlatform::AOscellatingPlatform ( )
```

The constructor of the class, which activates the actor's ability to tick, creates a static mesh, and initializes all variables.

\Macro that sets up the class to support the infrastructure required by the engine.

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

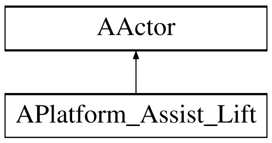
- OscellatingPlatform.h
- OscellatingPlatform.cpp

3.18 APlatform_Assist_Lift Class Reference

Class representing a continually moving actor in-game, positioned next to the first platform on the final level. Moves up and down to move player back to starting platform.

```
#include <Platform_Assist_Lift.h>
```

Inheritance diagram for APlatform_Assist_Lift:



Public Member Functions

- APlatform_Assist_Lift ()

  Macro that sets up the class to support the infrastructure required by the engine.
- virtual void Tick (float DeltaTime) override

  Called every frame and changes the location of the lift based on delta time.

Public Attributes

- UStaticMeshComponent ∗ VisualMesh

  For assist lift static mesh only.
- float ScaleFactor = 50.0

  Factor for which the platform will move in a given direction.

Protected Member Functions

- virtual void BeginPlay () override

  Called when the game starts or when spawned.

3.18.1 Detailed Description

Class representing a continually moving actor in-game, positioned next to the first platform on the final level. Moves up and down to move player back to starting platform.

3.18.2 Constructor & Destructor Documentation

3.18.2.1 APlatform_Assist_Lift()

```
APlatform_Assist_Lift::APlatform_Assist_Lift ( )
```

Macro that sets up the class to support the infrastructure required by the engine.

Sets default values for this actor's properties.

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

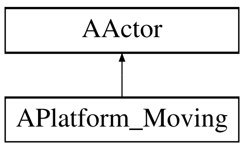
- Platform_Assist_Lift.h
- Platform_Assist_Lift.cpp

3.19 APlatform_Moving Class Reference

Class represents the four moving platforms in the final level.They have the ability to move up / down, left /right, and side / side with a scale factor that effects how far they move.

```
#include <Platform_Moving.h>
```

Inheritance diagram for APlatform_Moving:



Public Member Functions

- APlatform_Moving ()

  Macro that sets up the class to support the infrastructure required by the engine.
- virtual void Tick (float DeltaTime) override

  Called every frame. Sets new location of the moving platform based on delta time.

Public Attributes

- UStaticMeshComponent ∗ VisualMesh

    sStatic mesh representing the moving platform.
- float ScaleFactor = 50.0

    Factor for which the platform is to move from its starting location.
- bool IsPlatFormMoving = false

    Is the platform currently in motion.
- EMovementType Movement = EMovementType::UpDown

    Direction of movement for the platform from above enum.
- class UAudioComponent ∗ PlatformMovingAudio

    Audio component for moving platform, plays sound when platform is moving, not functional.
- class USoundBase ∗ MovingSound

    Sound played when the platform is in motion.

Protected Member Functions

- virtual void BeginPlay () override

    Sets random starting value.

Private Attributes

- int Random

    Random integer representing the time since creation so that all platforms do not move at the same pace.

3.19.1 Detailed Description

Class represents the four moving platforms in the final level.They have the ability to move up / down, left /right, and side / side with a scale factor that effects how far they move.

3.19.2 Constructor & Destructor Documentation

3.19.2.1 APlatform_Moving()

```
APlatform_Moving::APlatform_Moving ( )
```

Macro that sets up the class to support the infrastructure required by the engine.

Sets default values for this actor's properties. Sets up static mesh component.

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

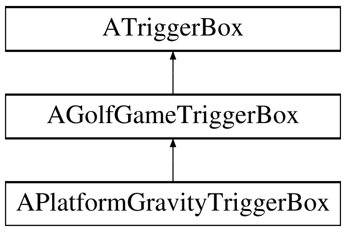
- Platform_Moving.h
- Platform_Moving.cpp

3.20 APlatformGravityTriggerBox Class Reference

∗Description: Upon a specific actor overlapping the triggerbox, the world gravity is set to a specified new gravity. The original gravity is then reset upon leaving the trigger box.

```
#include <PlatformGravityTriggerBox.h>
```

Inheritance diagram for APlatformGravityTriggerBox:



Public Member Functions

- virtual void OverlapBeginAction () override

  New world gravity set when specific actor enter trigger box.
- virtual void OverlapEndAction () override

  World gravity returned to original value upon leaving trigger box.

Public Attributes

- float GravityInsideTriggerBox

  New world gravity upon entering the trigger box.

Protected Member Functions

- virtual void BeginPlay () override

  Macro that sets up the class to support the infrastructure required by the engine.

Private Attributes

- float OriginalWorldGravity

  Original world gravity stored before overlap.

Additional Inherited Members

3.20.1 Detailed Description

∗Description: Upon a specific actor overlapping the triggerbox, the world gravity is set to a specified new gravity. The original gravity is then reset upon leaving the trigger box.

3.20.2 Member Function Documentation

3.20.2.1 BeginPlay()

```
void APlatformGravityTriggerBox::BeginPlay ( )  [override], [protected], [virtual]
```

Macro that sets up the class to support the infrastructure required by the engine.

Called when the game starts or when spawned. Stored world gravity in OriginalWorldGravity.

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

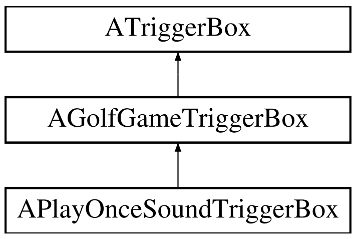
- PlatformGravityTriggerBox.h
- PlatformGravityTriggerBox.cpp

3.21 APlayOnceSoundTriggerBox Class Reference

Played a specific audio cue at the located specific one time when entering trigger box. The sound cannot be played again.

```
#include <PlayOnceSoundTriggerBox.h>
```

Inheritance diagram for APlayOnceSoundTriggerBox:



Public Member Functions

- virtual void OverlapBeginAction () override
  Sound cue is played for player when specific actor enters trigger box.
- virtual void OverlapEndAction () override
  Overlaps of specific actor ends.

Public Attributes

- class USoundBase ∗ Sound
  Macro that sets up the class to support the infrastructure required by the engine.

Private Attributes

- bool HasSoundPlayed = false

  Set to true when sound has played.

Additional Inherited Members

3.21.1 Detailed Description

Played a specific audio cue at the located specific one time when entering trigger box. The sound cannot be played again.

3.21.2 Member Data Documentation

3.21.2.1 Sound

```
class USoundBase* APlayOnceSoundTriggerBox::Sound
```

Macro that sets up the class to support the infrastructure required by the engine.

Sound to be played one time.

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

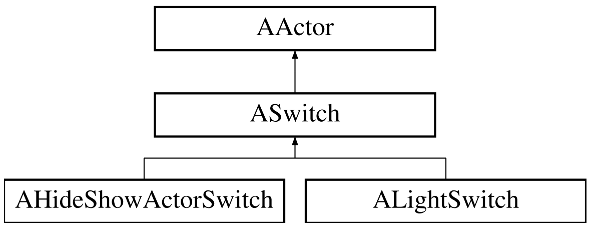
- PlayOnceSoundTriggerBox.h
- PlayOnceSoundTriggerBox.cpp

3.22 ASwitch Class Reference

Abstract class that switches actors on / off.

```
#include <Switch.h>
```

Inheritance diagram for ASwitch:



Public Attributes

- virtual void virtual GetMethodToCall(uint8 Status) PURE_VIRTUAL(ASwitch void virtual ActionOn() PURE↩ _VIRTUAL(ASwitch void virtual ActionOff() PURE_VIRTUAL(ASwitch uint8 GetActionOff () PURE_VIRTU↩ AL(ASwitch

  Gets the method to call when this switch is switched on or off. The method to call is determined by the Status.
- virtual uint8 GetActionOn () PURE_VIRTUAL(ASwitch

  Returns the "On" variable.

3.22.1 Detailed Description

Abstract class that switches actors on / off.

Ue4 interfaces do not work well, so an abstract class was used

3.22.2 Member Data Documentation

3.22.2.1 GetActionOff

```
virtual void virtual GetMethodToCall (uint8 Status) PURE_VIRTUAL(ASwitch void virtual Action↩
On () PURE_VIRTUAL(ASwitch void virtual ActionOff () PURE_VIRTUAL(ASwitch uint8 ASwitch::Get↩
ActionOff() PURE_VIRTUAL(ASwitch
```

Gets the method to call when this switch is switched on or off. The method to call is determined by the Status.

Parameters

Status	the GolfGameEnum that determines what method to call

The action that will be preformed when this switch is switched on, determined by the "On" variable.

The action that will be preformed when this switch is switched off, determined by the "Off" variable.

Returns the "Off" variable.

Returns

uint8 value of the "Off" variable.

3.22.2.2 GetActionOn

```
virtual uint8 ASwitch::GetActionOn() PURE_VIRTUAL(ASwitch
```

Returns the "On" variable.

Returns

uint8 value of the "On" variable.

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

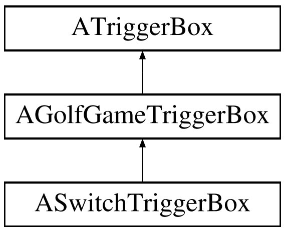
- Switch.h

3.23 ASwitchTriggerBox Class Reference

Trigger box that turns a list of switches on/off.

`#include <SwitchTriggerBox.h>`

Inheritance diagram for ASwitchTriggerBox:



Public Member Functions

- void SwitchOff ()

  Iterates through the switches array and turns each switch off.
- void SwitchOn ()

  Iterates through the switches array and turns each switch on.
- virtual void OverlapBeginAction () override

  Determines if this method can only be called once, has already been triggered or if OverlapEndAction has been disabled. Then calls OverlapBeginActionHelper to create the time that calls the SwitchOn method to turn on all switches.
- virtual void OverlapEndAction () override

  Determines if this method can only be called once or has previously been called. Creates the timer that calls the SwitchOff method to turn off all switches. the SwitchOn method.

Protected Attributes

- FTimerHandle TimerHandleSwitchOn

  The Timer handle for the Switch On Timer.
- FTimerHandle TimerHandleSwitchOff

  The Timer handle for the Switch Off Timer.
- TArray$<$ ASwitch $*$ $>$ Switches

  Array of switches to turn on or off.
- float DelayTimeOn

  The value of the time to wait before turning on the switch after overlap begins.
- float DelayTimeOff

  The value of the time to wait before turning off the switch after overlap ends.
- bool CanOnlyBeTriggeredOnce

  Indicates if this trigger box can only be triggered once or not.
- bool DisableOverlapEnd

  Indicates that the OverlapEndAction will be disabled and instead the off switch will execute, with the delay determined by DelayTimeOff, right after SwitchOn finishes executing.

Private Member Functions

- void OverlapBeginActionHelper ()

  Helper method for OverlapBeginAction that creates the timer for SwitchOn().

Private Attributes

- bool HasPreviouslyBeenTriggeredBegin

  Indicates if this trigger box has been previously overlapped by ActorToCheck.
- bool HasPreviouslyBeenTriggeredEnd

  Indicates if this trigger box has been previously stopped being overlapped by ActorToCheck.

3.23.1 Detailed Description

Trigger box that turns a list of switches on/off.

Triggerbox for any subclass that extends Switch.Holds an array of switches and when overlapped will flip the switches in the array on, when overlap ends flips the switches off.

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

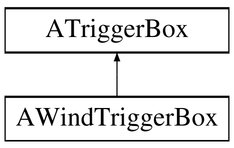
- SwitchTriggerBox.h
- SwitchTriggerBox.cpp

3.24 AWindTriggerBox Class Reference

A trigger box that once the ball enters the overlap area, a force is added directly to the ball every .1 seconds resulting in a wind like mechanic.

```
#include <WindTriggerBox.h>
```

Inheritance diagram for AWindTriggerBox:



Public Member Functions

- virtual void Tick ()

  Function for the timer tick and what happens in it.
- void AddForce (class AActor ∗OverlappedActor, class AActor ∗OtherActor)

  Function for adding the force the the ball during the tick.
- void OnOverlapBegin (class AActor ∗OverlappedActor, class AActor ∗OtherActor)

  Function of when the ball enters the box.
- void OnOverlapEnd (class AActor ∗OverlappedActor, class AActor ∗OtherActor)

  Function of when the ball exits the box.

Public Attributes

- class ABall ∗ Ball

  The ball that impact the switch to turn it on or off. Specific ball set in UE4 editor.
- int force = 1000

  Integer force to be added to the ball when add force is applied.
- bool IsTriggered = false

  Boolean to tell the trigger box whether or not wind has to be activated or not.
- bool IsUsable

  Boolean to tell the wind trigger box to turn on or not (can be accessed by a blueprint).

Protected Member Functions

- virtual void **BeginPlay** () override

Protected Attributes

- FTimerHandle InputADelayManager

  Handles the delay for the tick.

Private Attributes

- bool WindOn

  Boolean to tell the wind trigger box on whether or not the wind is currently on.
- FVector cameraForward

  Vector of where the wind should point.
- FTimerHandle loopHandle

  Handler for the wind tick.
- UStaticMeshComponent ∗ meshRootComp

  Mesh for inherited ball object.

3.24.1 Detailed Description

A trigger box that once the ball enters the overlap area, a force is added directly to the ball every .1 seconds resulting in a wind like mechanic.

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

- WindTriggerBox.h
- WindTriggerBox.cpp

3.25 AGolfGameCharacter::FTouchData Struct Reference

Public Attributes

- bool **bIsPressed**
- ETouchIndex::Type **FingerIndex**
- FVector **Location**
- bool **bMoved**

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

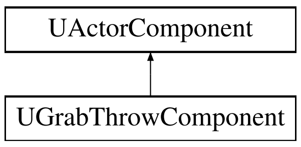
- GolfGameCharacter.h

3.26 UGrabThrowComponent Class Reference

This class is a component of the player character that allows the character to pick up, drop, and throw objects.

```
#include <GrabThrowComponent.h>
```

Inheritance diagram for UGrabThrowComponent:



Public Member Functions

- UGrabThrowComponent ()

  Sets default values for this component's properties.
- virtual void TickComponent (float DeltaTime, ELevelTick TickType, FActorComponentTickFunction ∗This↩ TickFunction) override

  Called every frame.
- bool Grab (UObject ∗WorldContextObject, class UPhysicsHandleComponent ∗Ph, UCameraComponent ∗FPCameraComponent)

  Attempts to grab an object of the appropriate type and within the appropriate distance in front of the player character.
- bool Throw (class UPhysicsHandleComponent ∗Ph, UCameraComponent ∗FPCameraComponent)

  Attempts to throw an object held by the player character.
- bool Release (class UPhysicsHandleComponent ∗Ph, bool bThrow)

  Attempts to relseae an object held by the player character.
- void TraceHandleLocation (class UPhysicsHandleComponent ∗Ph, UCameraComponent ∗FPCamera↩ Component)

  Traces the location of the physics handle related to the position of the player character's camera.
- void SummonGrabBall (ABall ∗Ball, FVector SummonLocation, UPhysicsHandleComponent ∗Ph)

  Summons the ball to the player and sets the ball to grabbed by the player.
- FORCEINLINE bool GetIsObjectHeld () const

  Returns if an object is currently being held.

Protected Member Functions

- virtual void BeginPlay () override

  Called when the game starts or when spawned.

Private Attributes

- FVector HandleLocation

  Physics handle destination.
- TArray< TEnumAsByte< EObjectTypeQuery > > PhysicsObjectType

  Type of object that can be picked up.
- TArray< AActor * > ActorsToIgnore

  A list of actors to ignore when attempting to grab objects.
- UPrimitiveComponent * HitComponent

  Cached reference to the hit component.
- class USoundBase * GrabSound

  Sound for when the object is grabbed.
- class USoundBase * ThrowSound

  Sound for when the object is thrown.
- class USoundBase * ReleaseSound

  Sound for when the object is released.
- bool bObjectHeld

  Checks if object is being held.
- bool bPhysicsHandleActive

  Check if physics handle is active.
- float MinGrabDist = 300.0f

  Min distance to allow pickup.
- float MaxGrabDist = 1500.0f

  Max distance to allow pickup.
- float PlayerObjectDist = 1.0f

  Distance between player and pickup.
- float ThrowingForce = 1500.0f

  The amount of force used to throw the object.
- float SnapDistance = 200.0f

  The distance from the character that an object will snap to when grabbed or summoned.

3.26.1 Detailed Description

This class is a component of the player character that allows the character to pick up, drop, and throw objects.

3.26.2 Member Function Documentation

3.26.2.1 Grab()

```
bool UGrabThrowComponent::Grab (
            UObject * WorldContextObject,
            class UPhysicsHandleComponent * Ph,
            UCameraComponent * FPCameraComponent )
```

Attempts to grab an object of the appropriate type and within the appropriate distance in front of the player character.

Parameters

WorldContextObject	A reference to the player character.
Ph	A reference to the player character's physics handle.
FPCameraComponent	A reference to the player character's camera component.

3.26.2.2 Release()

```
bool UGrabThrowComponent::Release (
            class UPhysicsHandleComponent * Ph,
            bool bThrow )
```

Attempts to relseae an object held by the player character.

Parameters

Ph	A reference to the player character's physics handle
bThrow	a boolean signifying whether a throw has been performed

3.26.2.3 SummonGrabBall()

```
void UGrabThrowComponent::SummonGrabBall (
            ABall * Ball,
            FVector SummonLocation,
            UPhysicsHandleComponent * Ph )
```

Summons the ball to the player and sets the ball to grabbed by the player.

Parameters

Ball	the in game ball that the player references.
SummonLocation	the location that the ball will be summoned to respective to the player and their grab position.
Ph	the players physics handle that aids in holding and moving the ball.

3.26.2.4 Throw()

```
bool UGrabThrowComponent::Throw (
            class UPhysicsHandleComponent * Ph,
            UCameraComponent * FPCameraComponent )
```

Attempts to throw an object held by the player character.

Parameters

Ph	A reference to the player character's physics handle.
FPCameraComponent	A reference to the player character's camera component.

3.26.2.5 TraceHandleLocation()

```
void UGrabThrowComponent::TraceHandleLocation (
             class UPhysicsHandleComponent * Ph,
             UCameraComponent * FPCameraComponent )
```

Traces the location of the physics handle related to the position of the player character's camera.

Parameters

Ph	A reference to the player character's physics handle.
FPCameraComponent	A reference to the player character's camera component.

3.26.3 Member Data Documentation

3.26.3.1 HandleLocation

```
FVector UGrabThrowComponent::HandleLocation  [private]
```

Physics handle destination.

\Macro that sets up the class to support the infrastructure required by the engine.

-

3.26.3.2 SnapDistance

```
float UGrabThrowComponent::SnapDistance = 200.0f  [private]
```

The distance from the character that an object will snap to when grabbed or summoned.

-

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

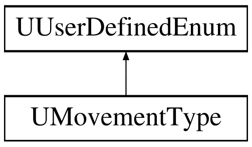
- GrabThrowComponent.h
- GrabThrowComponent.cpp

3.27 UMovementType Class Reference

Generated by Ue4, not directly utilized.

```
#include <MovementType.h>
```

Inheritance diagram for UMovementType:



3.27.1 Detailed Description

Generated by Ue4, not directly utilized.

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

- MovementType.h

Index