Dockit League

Generated by Doxygen 1.8.12

Contents

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:

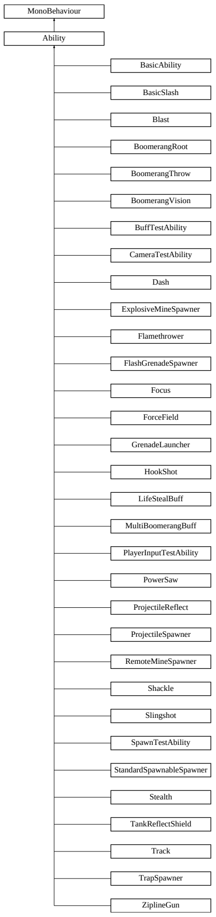
Chapter 3

Class Documentation

3.1 Ability Class Reference

Base class for all abilities.

Inheritance diagram for Ability:



Public Member Functions

- virtual void Initialize (Docking dock, Animator anim, int abId)

  Initialization that happens locally on every client.
- virtual void InitializeLocalPlayer (AbilityUI abilityUI)

  Initialization that only happens for the local player (Player controlling this ability). Called after Initialize, so the references are already set up.
- virtual void CooldownReady ()

  Called by the cooldown whenever its ready.
- abstract void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- virtual void ButtonUp ()

  Called when the associated ability button is released.
- virtual void CancelAbility ()

  Call for cancelling abilities. Override in abilities that may be interrupted.
- abstract void SetActive (bool state)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.
- virtual void SetModifier (bool state)

  Called by the Modifier. Appropriate place for doing local changes.
- void ReduceCooldown (float reductionAmount)

  Reduces the current cooldown for the ability.
- virtual void SetElement (ElementalContainer.ComboableElements element)

Public Attributes

- float **cooldownDuration**
- Sprite **icon**

Protected Member Functions

- virtual void Update ()

  Runs on every client, but only the local player has cooldown initialized.

Protected Attributes

- Docking **docking**
- Animator **animator**
- int **abilityId**
- AbilityCooldown **cooldown**

Properties

- bool AbilityLock `[get, set]`

  Get and Set ability lock. Lock prevents the player from using abilities.

3.1.1 Detailed Description

Base class for all abilities.

3.1.2 Member Function Documentation

3.1.2.1 ButtonDown()

```
abstract void Ability.ButtonDown ( )  [pure virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implemented in BoomerangThrow, Slingshot, BasicSlash, BoomerangRoot, PlayerInputTestAbility, ProjectileReflect, Focus, BoomerangVision, HookShot, CameraTestAbility, LifeStealBuff, PowerSaw, MultiBoomerangBuff, Stealth, ZiplineGun, TankReflectShield, BuffTestAbility, Flamethrower, ForceField, Blast, Dash, Track, TrapSpawner, ExplosiveMineSpawner, StandardSpawnableSpawner, FlashGrenadeSpawner, SpawnTestAbility, RemoteMineSpawner, GrenadeLauncher, ProjectileSpawner, BasicAbility, and Shackle.

3.1.2.2 ButtonUp()

```
virtual void Ability.ButtonUp ( )  [virtual]
```

Called when the associated ability button is released.

Reimplemented in BoomerangThrow, Slingshot, and ZiplineGun.

3.1.2.3 CancelAbility()

```
virtual void Ability.CancelAbility ( )  [virtual]
```

Call for cancelling abilities. Override in abilities that may be interrupted.

Reimplemented in Slingshot, PlayerInputTestAbility, CameraTestAbility, and Focus.

3.1.2.4 CooldownReady()

```
virtual void Ability.CooldownReady ( )  [virtual]
```

Called by the cooldown whenever its ready.

Reimplemented in PowerSaw, and Shackle.

3.1.2.5 Initialize()

```
virtual void Ability.Initialize (
             Docking dock,
             Animator anim,
             int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented in BoomerangThrow, ProjectileReflect, BoomerangVision, LifeStealBuff, Stealth, PowerSaw, BasicSlash, Flamethrower, TankReflectShield, BuffTestAbility, Blast, Track, ForceField, and Dash.

3.1.2.6 InitializeLocalPlayer()

```
virtual void Ability.InitializeLocalPlayer (
             AbilityUI abilityUI )  [virtual]
```

Initialization that only happens for the local player (Player controlling this ability). Called after Initialize, so the references are already set up.

Reimplemented in Slingshot, Focus, CameraTestAbility, PlayerInputTestAbility, and ZiplineGun.

3.1.2.7 ReduceCooldown()

```
void Ability.ReduceCooldown (
             float reductionAmount )
```

Reduces the current cooldown for the ability.

Parameters

reductionAmount	The amount deducted for the current cooldown.

3.1.2.8 SetActive()

```
abstract void Ability.SetActive (
             bool state )  [pure virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

Implemented in BoomerangThrow, Slingshot, PlayerInputTestAbility, BasicSlash, CameraTestAbility, BoomerangRoot, PowerSaw, Track, Focus, ProjectileReflect, TankReflectShield, Stealth, ZiplineGun, HookShot, BoomerangVision, LifeStealBuff, MultiBoomerangBuff, RemoteMineSpawner, BuffTestAbility, Shackle, Flamethrower, TrapSpawner, Blast, ForceField, ExplosiveMineSpawner, SpawnTestAbility, StandardSpawnableSpawner, FlashGrenadeSpawner, Dash, ProjectileSpawner, GrenadeLauncher, and BasicAbility.

3.1.2.9 SetElement()

```
virtual void Ability.SetElement (
             ElementalContainer.ComboableElements element )  [virtual]
```

Used for local spawning of elemental effect prefabs

Parameters

element	The element we want to set

Reimplemented in BoomerangThrow, and BasicSlash.

3.1.2.10 SetModifier()

```
virtual void Ability.SetModifier (
             bool state )  [virtual]
```

Called by the Modifier. Appropriate place for doing local changes.

Parameters

state	If the modifier should be activated or deactivated.

Reimplemented in BoomerangThrow, BasicSlash, ProjectileReflect, BoomerangVision, Stealth, LifeStealBuff, MultiBoomerangBuff, BuffTestAbility, and Flamethrower.

3.1.2.11 Update()

```
virtual void Ability.Update ( )  [protected], [virtual]
```

Runs on every client, but only the local player has cooldown initialized.

Reimplemented in BoomerangThrow, ProjectileReflect, BoomerangRoot, and BasicSlash.

3.1.3 Property Documentation

3.1.3.1 AbilityLock

```
bool Ability.AbilityLock  [get], [set]
```

Get and Set ability lock. Lock prevents the player from using abilities.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/Ability.cs

3.2 AbilityCooldown Class Reference

Handles the cooldown for abilities. Runs on the local player.

Public Member Functions

- AbilityCooldown (Ability ab, float duration, AbilityUI abUI)

    Constructor.
- void ReduceCooldown (float reductionAmount)

    Reduces the current cooldown for the ability.
- void Update ()

    Update loop. Handles timer and ability ui update.
- void Activate ()

    Called on ability activation. Activates cooldown.
- void ActivateHiddenCooldown (float hiddenCooldown)

    Can be called from abilities whenever they need a hidden cooldown, a simple short cooldown in addition to the standard cooldown for instance.
- bool IsReady ()

    Used for checking if the ability is on cooldown.

3.2.1 Detailed Description

Handles the cooldown for abilities. Runs on the local player.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 AbilityCooldown()

```
AbilityCooldown.AbilityCooldown (
             Ability ab,
             float duration,
             AbilityUI abUI )
```

Constructor.

Parameters

duration	Length of cooldown.

3.2.3 Member Function Documentation

3.2.3.1 Activate()

```
void AbilityCooldown.Activate ( )
```

Called on ability activation. Activates cooldown.

3.2.3.2 ActivateHiddenCooldown()

```
void AbilityCooldown.ActivateHiddenCooldown (
             float hiddenCooldown )
```

Can be called from abilities whenever they need a hidden cooldown, a simple short cooldown in addition to the standard cooldown for instance.

3.2.3.3 IsReady()

```
bool AbilityCooldown.IsReady ( )
```

Used for checking if the ability is on cooldown.

Returns

Whether the ability is on cooldown or not.

3.2.3.4 ReduceCooldown()

```
void AbilityCooldown.ReduceCooldown (
             float reductionAmount )
```

Reduces the current cooldown for the ability.

Parameters

reductionAmount	The amount deducted for the current cooldown.

3.2.3.5 Update()

```
void AbilityCooldown.Update ( )
```

Update loop. Handles timer and ability ui update.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/AbilityCooldown.cs

3.3 AbilityUI Class Reference

Handles the update of the abilitys UI.

Public Member Functions

- void Initialize (PlayerUIHandler uiHandler)

  Initialize the ability UI.
- void Activate ()

  Called on ability activation. Activates cooldown.
- void UpdateCooldown (float newTimeLeft)

  Updates the current cooldown time with the new time.
- void SetAbility (Ability newAbility)

  Changes sprites and cooldown to the new ability.
- void ClearAbility (Sprite emptySlot)

  Stops the update loop and resets the UI to its original empty state.

Public Attributes

- Image **abilityIcon**
- Image **darkMask**

3.3.1 Detailed Description

Handles the update of the abilitys UI.

3.3.2 Member Function Documentation

3.3.2.1 Activate()

```
void AbilityUI.Activate ( )
```

Called on ability activation. Activates cooldown.

3.3.2.2 ClearAbility()

```
void AbilityUI.ClearAbility (
            Sprite emptySlot )
```

Stops the update loop and resets the UI to its original empty state.

Parameters

emptySlot	Sprite used in an empty slot.

3.3.2.3 Initialize()

```
void AbilityUI.Initialize (
            PlayerUIHandler uiHandler )
```

Initialize the ability UI.

Initialize the ability UI.

Parameters

uiHandler	Reference to associated PlayerUIHandler.

3.3.2.4 SetAbility()

```
void AbilityUI.SetAbility (
            Ability newAbility )
```

Changes sprites and cooldown to the new ability.

Parameters

newAbility	Reference to the new ability.

3.3.2.5 UpdateCooldown()

```
void AbilityUI.UpdateCooldown (
            float newTimeLeft )
```

Updates the current cooldown time with the new time.

Parameters

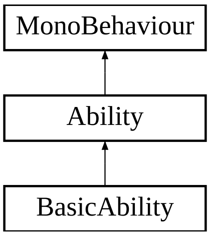
newTimeLeft	The new current cooldown time.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/UI/AbilityUI.cs

3.4 BasicAbility Class Reference

Inheritance diagram for BasicAbility:



Public Member Functions

- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Public Attributes

- float **damage** = 10f
- string **animatorTrigger**

Additional Inherited Members

3.4.1 Member Function Documentation

3.4.1.1 ButtonDown()

```
override void BasicAbility.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.4.1.2 SetActive()

```
override void BasicAbility.SetActive (
            bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

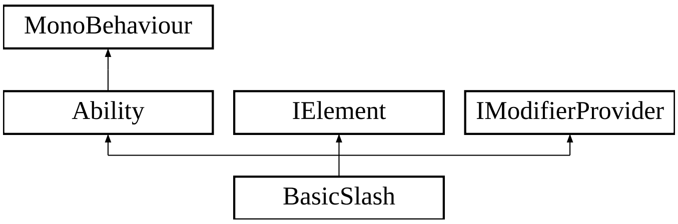
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BasicAbility.cs

3.5 BasicSlash Class Reference

Inheritance diagram for BasicSlash:



Public Member Functions

- override void Initialize (Docking dock, Animator anim, int abId)

  Initialization that happens locally on every client.
- override void ButtonDown ()

  Callback for what this ability should do once its associated button has been pressed
- override void SetActive (bool state=false)

  Callback for what this ability is supposed to do depending on given state. State is always false here
- override void SetModifier (bool state=false)

  Called by the Modifier. Appropriate place for doing local changes.
- override void SetElement (ElementalContainer.ComboableElements element)

  Callback for what this ability is supposed to do locally when applying a element

Public Attributes

- float **damageDealt** = 20f
- string **animatorTrigger**
- LifeStealBuff **lifeStealBuff**
- ElementalModifiers **elementalModifiers** = new ElementalModifiers()
- bool **swingActive**

Protected Member Functions

- override void Update ()

  Runs on every client, but only the local player has cooldown initialized.

Additional Inherited Members

3.5.1 Member Function Documentation

3.5.1.1 ButtonDown()

```
override void BasicSlash.ButtonDown ( )  [virtual]
```

Callback for what this ability should do once its associated button has been pressed

Implements Ability.

3.5.1.2 Initialize()

```
override void BasicSlash.Initialize (
            Docking dock,
            Animator anim,
            int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented from Ability.

3.5.1.3 SetActive()

```
override void BasicSlash.SetActive (
             bool state = false )  [virtual]
```

Callback for what this ability is supposed to do depending on given state. State is always false here

Parameters

state	Whether the ability is to be active or now

Implements Ability.

3.5.1.4 SetElement()

```
override void BasicSlash.SetElement (
             ElementalContainer.ComboableElements element )  [virtual]
```

Callback for what this ability is supposed to do locally when applying a element

Parameters

element	

Reimplemented from Ability.

3.5.1.5 SetModifier()

```
override void BasicSlash.SetModifier (
             bool state = false )  [virtual]
```

Called by the Modifier. Appropriate place for doing local changes.

Parameters

state	If the modifier should be activated or deactivated.

Reimplemented from Ability.

3.5.1.6 Update()

```
override void BasicSlash.Update ( )  [protected], [virtual]
```

Runs on every client, but only the local player has cooldown initialized.

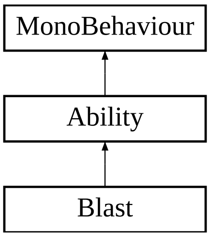
Reimplemented from Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BrawlerKit/BasicSlash.cs

3.6 Blast Class Reference

Inheritance diagram for Blast:



Public Member Functions

- override void Initialize (Docking dock, Animator anim, int abId)

  Initialization that happens locally on every client.
- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.
- void **OnTriggerEnter** (Collider other)

Public Attributes

- float **blastForce**
- string **animatorTrigger**

Additional Inherited Members

3.6.1 Member Function Documentation

3.6.1.1 ButtonDown()

```
override void Blast.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.6.1.2 Initialize()

```
override void Blast.Initialize (
            Docking dock,
            Animator anim,
            int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented from Ability.

3.6.1.3 SetActive()

```
override void Blast.SetActive (
            bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

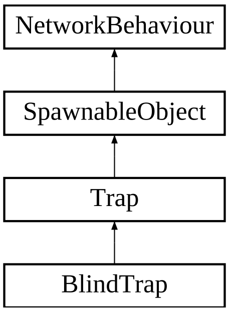
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BomberKit/Blast.cs

3.7 BlindTrap Class Reference

Inheritance diagram for BlindTrap:



Public Member Functions

- override void HandleTrigger (PlayerStatus playerStatus)

    Callback that allows this trap to do whatever it wants whenever it is triggered This one simply applies the member structs containing modifier info

Public Attributes

- ModifierInfo **blindInfo**

Additional Inherited Members

3.7.1 Member Function Documentation

3.7.1.1 HandleTrigger()

```
override void BlindTrap.HandleTrigger (
            PlayerStatus playerStatus )  [virtual]
```

Callback that allows this trap to do whatever it wants whenever it is triggered This one simply applies the member structs containing modifier info

Parameters

playerStatus	The PlayerStatus component of the player that is in the trap

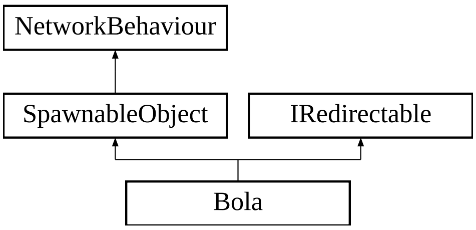
Reimplemented from Trap.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TrapperKit/BlindTrap.cs

3.8 Bola Class Reference

Inheritance diagram for Bola:



Public Attributes

- float **moveSpeed** = 8f
- float **moveSpeedOnHit** = 20f
- float **lifetime** = 10f
- float **hitRadius** = 2f
- float **rotationSpeed** = 500f
- ModifierInfo **slowModifier**
- ModifierInfo **stunModifier**
- Transform **visuals**
- Transform **leftBall**
- Transform **rightBall**
- LineRenderer **lineRenderer**

Additional Inherited Members

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/SniperKit/Bola.cs

3.9 BoomerangDataContainer Class Reference

Public Attributes

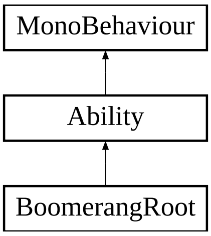
- const int **NUM_CONTROL_POINTS** = 4
- Transform [ ] **bezierControlPoints** = new Transform[NUM_CONTROL_POINTS]
- Vector3 [ ] **storedPositions** = new Vector3[NUM_CONTROL_POINTS]

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BoomerangKit/BoomerangThrow.cs

3.10 BoomerangRoot Class Reference

Inheritance diagram for BoomerangRoot:



Public Member Functions

- override void ButtonDown ()

  Callback for what this ability does locally when its associated button is pressed
- override void SetActive (bool state=false)

  Callback for what this ability is supposed to do locally on all clients when the ability state is changed

Public Attributes

- ModifierInfo **rootInfo**
- SpriteRenderer [ ] **rootIndicators**
- Animator **animationController**
- float **activeDuration** = 0.5f
- string **animationTrigger** = "Root"
- Color **activeColor**
- bool **rootActive** = false

Protected Member Functions

- override void Update ()

  Runs on every client, but only the local player has cooldown initialized.

Additional Inherited Members

3.10.1 Member Function Documentation

3.10.1.1 ButtonDown()

```
override void BoomerangRoot.ButtonDown ( )  [virtual]
```

Callback for what this ability does locally when its associated button is pressed

Implements Ability.

3.10.1.2 SetActive()

```
override void BoomerangRoot.SetActive (
             bool state = false )  [virtual]
```

Callback for what this ability is supposed to do locally on all clients when the ability state is changed

Parameters

state	The new ability state

Implements Ability.

3.10.1.3 Update()

```
override void BoomerangRoot.Update ( )  [protected], [virtual]
```

Runs on every client, but only the local player has cooldown initialized.

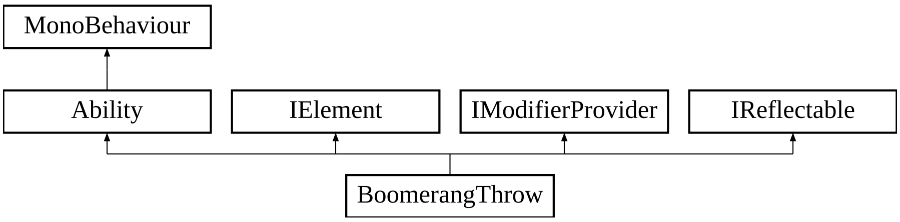
Reimplemented from Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BoomerangKit/BoomerangRoot.cs

3.11 BoomerangThrow Class Reference

Inheritance diagram for BoomerangThrow:



Public Member Functions

- override void Initialize (Docking dock, Animator anim, int abId)

    Initialization that happens locally on every client.
- override void ButtonDown ()

    Callback for what this ability does locally when its associated button is pressed
- override void ButtonUp ()

    Callback for what this ability does locally when its associated button is released
- override void SetActive (bool state=false)

    Callback for what this ability is supposed to do locally on all clients when the ability state is changed
- override void SetModifier (bool state=false)

    Callback for what this ability is supposed to do when a modifier state changes
- override void SetElement (ElementalContainer.ComboableElements element)

    Callback for what this ability is supposed to do locally when applying a element
- void **ReflectVelocity** ()

Public Attributes

- List< LineRenderer > **approximatePathRenderers** = new List<LineRenderer>()
- BoomerangDataContainer [ ] **boomerangData** = new BoomerangDataContainer[NUM_BOOMERANGS]
- List< TrailRenderer > **trailRenderers** = new List<TrailRenderer>()
- GameObject [ ] **boomerangObjs** = new GameObject[NUM_BOOMERANGS]
- BoomerangRoot **boomerangRootScript**
- MultiBoomerangBuff **boomerangBuffScript**
- AnimationCurve **velocityCurve**
- float **damageDealt** = 10f
- float **boomerangSpeed** = 5f
- float **spinMultiplierWhileActive** = 4f
- const int **NUM_BOOMERANGS** = 3
- ElementalModifiers **elementalModifiers** = new ElementalModifiers()

Protected Member Functions

- override void Update ()

  Runs on every client, but only the local player has cooldown initialized.

Additional Inherited Members

3.11.1 Member Function Documentation

3.11.1.1 ButtonDown()

```
override void BoomerangThrow.ButtonDown ( )  [virtual]
```

Callback for what this ability does locally when its associated button is pressed

Implements Ability.

3.11.1.2 ButtonUp()

```
override void BoomerangThrow.ButtonUp ( )  [virtual]
```

Callback for what this ability does locally when its associated button is released

Reimplemented from Ability.

3.11.1.3 Initialize()

```
override void BoomerangThrow.Initialize (
             Docking dock,
             Animator anim,
             int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented from Ability.

3.11.1.4 SetActive()

```
override void BoomerangThrow.SetActive (
             bool state = false )  [virtual]
```

Callback for what this ability is supposed to do locally on all clients when the ability state is changed

Parameters

state	The new ability state

Implements Ability.

3.11.1.5 SetElement()

```
override void BoomerangThrow.SetElement (
            ElementalContainer.ComboableElements element )  [virtual]
```

Callback for what this ability is supposed to do locally when applying a element

Parameters

element	

Reimplemented from Ability.

3.11.1.6 SetModifier()

```
override void BoomerangThrow.SetModifier (
            bool state = false )  [virtual]
```

Callback for what this ability is supposed to do when a modifier state changes

Parameters

state	The new modifier state

Reimplemented from Ability.

3.11.1.7 Update()

```
override void BoomerangThrow.Update ( )  [protected], [virtual]
```

Runs on every client, but only the local player has cooldown initialized.

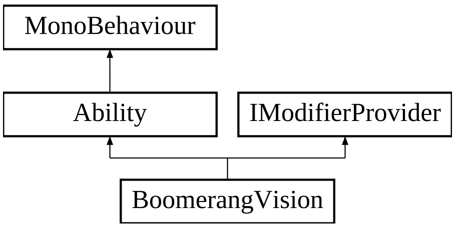
Reimplemented from Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BoomerangKit/BoomerangThrow.cs

3.12 BoomerangVision Class Reference

Inheritance diagram for BoomerangVision:



Public Member Functions

- override void Initialize (Docking dock, Animator anim, int abId)

  Initialization that happens locally on every client.
- override void ButtonDown ()

  Callback for what this ability does locally when its associated button is pressed
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.
- override void SetModifier (bool state)

  Callback for what this ability is supposed to do when a modifier state changes

Public Attributes

- BoomerangThrow **boomerangThrowScript**
- MultiBoomerangBuff **boomerangBuffScript**
- GameObject **visionIndicator**
- float **visionRadiusWhileActive** = 10f
- float **visionRadiusExtraBoomerangs** = 5f
- float **visionLerpSpeed** = 5f
- ModifierInfo **visionModifier**

Additional Inherited Members

3.12.1 Member Function Documentation

3.12.1.1 ButtonDown()

```
override void BoomerangVision.ButtonDown ( )  [virtual]
```

Callback for what this ability does locally when its associated button is pressed

Implements Ability.

3.12.1.2 Initialize()

```
override void BoomerangVision.Initialize (
            Docking dock,
            Animator anim,
            int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented from Ability.

3.12.1.3 SetActive()

```
override void BoomerangVision.SetActive (
             bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

Implements Ability.

3.12.1.4 SetModifier()

```
override void BoomerangVision.SetModifier (
             bool state )  [virtual]
```

Callback for what this ability is supposed to do when a modifier state changes

Parameters

state	The new modifier state

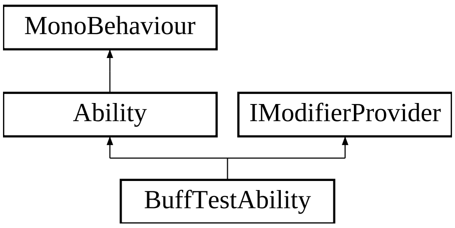
Reimplemented from Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BoomerangKit/BoomerangVision.cs

3.13 BuffTestAbility Class Reference

Inheritance diagram for BuffTestAbility:



Public Member Functions

- override void Initialize (Docking dock, Animator anim, int abId)

    Initialization that happens locally on every client.
- override void ButtonDown ()

    Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state=false)

    Synchronized state call between clients. Appropriate place for starting local animations and sounds.
- override void SetModifier (bool state=false)

    Called by the Modifier. Appropriate place for doing local changes.

Public Attributes

- SpriteRenderer [ ] **visuals**
- Color **activeColor**
- ModifierInfo **buff**

Additional Inherited Members

3.13.1 Member Function Documentation

3.13.1.1 ButtonDown()

```
override void BuffTestAbility.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.13.1.2 Initialize()

```
override void BuffTestAbility.Initialize (
             Docking dock,
             Animator anim,
             int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented from Ability.

3.13.1.3 SetActive()

```
override void BuffTestAbility.SetActive (
             bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

Implements Ability.

3.13.1.4 SetModifier()

```
override void BuffTestAbility.SetModifier (
             bool state = false )  [virtual]
```

Called by the Modifier. Appropriate place for doing local changes.

Parameters

state	If the modifier should be activated or deactivated.

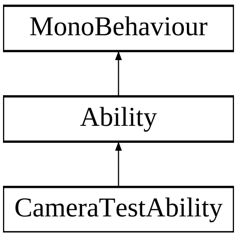
Reimplemented from Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TestAbilities/BuffTestAbility.cs

3.14 CameraTestAbility Class Reference

Inheritance diagram for CameraTestAbility:



Public Member Functions

- override void InitializeLocalPlayer (AbilityUI abilityUI)

  Initialization that only happens for the local player (Player controlling this ability). Called after Initialize, so the references are already set up.

- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void CancelAbility ()

  Call for cancelling abilities. Override in abilities that may be interrupted.
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Public Attributes

- Transform **target**
- float **targetOrthoSize**
- float **targetViewAngle**
- float **targetViewRadius**
- float **lerpSpeed**

Additional Inherited Members

3.14.1 Member Function Documentation

3.14.1.1 ButtonDown()

```
override void CameraTestAbility.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.14.1.2 CancelAbility()

```
override void CameraTestAbility.CancelAbility ( )  [virtual]
```

Call for cancelling abilities. Override in abilities that may be interrupted.

Reimplemented from Ability.

3.14.1.3 InitializeLocalPlayer()

```
override void CameraTestAbility.InitializeLocalPlayer (
             AbilityUI abilityUI )  [virtual]
```

Initialization that only happens for the local player (Player controlling this ability). Called after Initialize, so the references are already set up.

Reimplemented from Ability.

3.14.1.4 SetActive()

```
override void CameraTestAbility.SetActive (
             bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

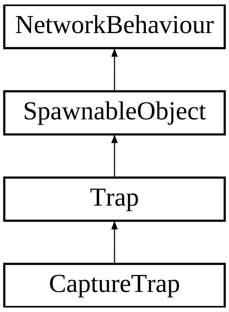
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TestAbilities/CameraTestAbility.cs

3.15 CaptureTrap Class Reference

Inheritance diagram for CaptureTrap:



Public Member Functions

- override void HandleTrigger (PlayerStatus playerStatus)

  Callback for when the trap is triggered. Sets relevant gameobjects as active to display visuals and starts a coroutine for spawning the walls.

Public Attributes

- GameObject **walls**
- float **timeBeforeWallsSpawn** = 1f
- float **pullForce** = 10f
- float **fadeSpeed** = 10f
- float **fadeTimeOffsetMultiplier** = 1.5f

Additional Inherited Members

3.15.1 Member Function Documentation

3.15.1.1 HandleTrigger()

```
override void CaptureTrap.HandleTrigger (
            PlayerStatus playerStatus )  [virtual]
```

Callback for when the trap is triggered. Sets relevant gameobjects as active to display visuals and starts a coroutine for spawning the walls.

Parameters

playerStatus	The PlayerStatus component of the player that is in the trap

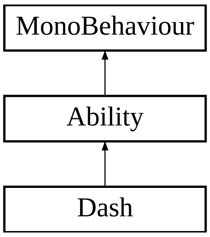
Reimplemented from Trap.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TrapperKit/CaptureTrap.cs

3.16 Dash Class Reference

Inheritance diagram for Dash:



Public Member Functions

- override void Initialize (Docking dock, Animator anim, int abId)
  Initialization that happens locally on every client.
- override void ButtonDown ()
  Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state=false)
  Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Additional Inherited Members

3.16.1 Member Function Documentation

3.16.1.1 ButtonDown()

```
override void Dash.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.16.1.2 Initialize()

```
override void Dash.Initialize (
            Docking dock,
            Animator anim,
            int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented from Ability.

3.16.1.3 SetActive()

```
override void Dash.SetActive (
            bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

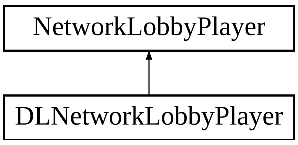
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/MarksmanKit/Dash.cs

3.17 DLNetworkLobbyPlayer Class Reference

Inheritance diagram for DLNetworkLobbyPlayer:



Public Member Functions

- override void OnClientEnterLobby ()

  Callback that initialises all necessary data for when a player enters the lobby. This includes player name, player color, adding the player to a team and telling the network manager that a new player has joined
- override void OnStartAuthority ()

  Callback that simply sets up the client side parts of a lobby for the connecting player
- override void OnClientReady (bool readyState)

  Makes the local UI uninteractable once the player has chosen to be ready
- void OnNameChange (string newName)

  SyncVarHook for handling name changes
- void OnColorChange (Color newColor)

SyncVarHook for handling color changes
- void OnReadyStateChange (bool state)

  SyncVarHook for handling ready states
- void OnColorClicked ()

  A function that simply calls the CmdColorChange() command
- void OnReadyClicked ()

  A simple function that tells the network that this player is ready to begin
- void OnNameChanged (string str)

  A simple function that calls the CmdNameChanged(str) command
- void ToggleReadyButton (bool enabled)

  Sets the state of the ready button on the UI to the parameter one
- GameObject GetVisuals ()

  Returns the UI elements of a player
- void CmdColorChange ()

  Updates the server when a player has chosen a new team/color
- void CmdNameChanged (string name)

  Updates the server when a player has chosen a new name
- void CmdUpdateReadyState (bool state)

  Updates the server when a player is ready
- void OnDestroy ()

  Callback for when a lobby player leaves the lobby and gets destroyed It tells the lobbyHandler to remove this player and tells the networkManager that a player has left.

Public Attributes

- Button **colorButton**
- InputField **nameInput**
- Button **readyButton**
- GameObject **visuals**
- string **playerName** = ""
- Color **playerColor** = Color.white
- bool **isReady** = false

3.17.1 Member Function Documentation

3.17.1.1 CmdColorChange()

```
void DLNetworkLobbyPlayer.CmdColorChange ( )
```

Updates the server when a player has chosen a new team/color

3.17.1.2 CmdNameChanged()

```
void DLNetworkLobbyPlayer.CmdNameChanged (
             string name )
```

Updates the server when a player has chosen a new name

Parameters

name	The new player name

3.17.1.3 CmdUpdateReadyState()

```
void DLNetworkLobbyPlayer.CmdUpdateReadyState (
            bool state )
```

Updates the server when a player is ready

Parameters

state	The ready state

3.17.1.4 GetVisuals()

```
GameObject DLNetworkLobbyPlayer.GetVisuals ( )
```

Returns the UI elements of a player

Returns

The player visuals

3.17.1.5 OnClientEnterLobby()

```
override void DLNetworkLobbyPlayer.OnClientEnterLobby ( )
```

Callback that initialises all necessary data for when a player enters the lobby. This includes player name, player color, adding the player to a team and telling the network manager that a new player has joined

3.17.1.6 OnClientReady()

```
override void DLNetworkLobbyPlayer.OnClientReady (
            bool readyState )
```

Makes the local UI uninteractable once the player has chosen to be ready

Parameters

readyState	Whether the client is ready or not

3.17.1.7 OnColorChange()

```
void DLNetworkLobbyPlayer.OnColorChange (
             Color newColor )
```

SyncVarHook for handling color changes

Parameters

newColor	The new team color

3.17.1.8 OnColorClicked()

```
void DLNetworkLobbyPlayer.OnColorClicked ( )
```

A function that simply calls the CmdColorChange() command

3.17.1.9 OnDestroy()

```
void DLNetworkLobbyPlayer.OnDestroy ( )
```

Callback for when a lobby player leaves the lobby and gets destroyed It tells the lobbyHandler to remove this player and tells the networkManager that a player has left.

3.17.1.10 OnNameChange()

```
void DLNetworkLobbyPlayer.OnNameChange (
             string newName )
```

SyncVarHook for handling name changes

Parameters

newName	The new player name

3.17.1.11 OnNameChanged()

```
void DLNetworkLobbyPlayer.OnNameChanged (
             string str )
```

A simple function that calls the CmdNameChanged(str) command

Parameters

str	The new player name

3.17.1.12 OnReadyClicked()

```
void DLNetworkLobbyPlayer.OnReadyClicked ( )
```

A simple function that tells the network that this player is ready to begin

3.17.1.13 OnReadyStateChange()

```
void DLNetworkLobbyPlayer.OnReadyStateChange (
             bool state )
```

SyncVarHook for handling ready states

Parameters

state	If the is player ready or not

3.17.1.14 OnStartAuthority()

```
override void DLNetworkLobbyPlayer.OnStartAuthority ( )
```

Callback that simply sets up the client side parts of a lobby for the connecting player

3.17.1.15 ToggleReadyButton()

```
void DLNetworkLobbyPlayer.ToggleReadyButton (
             bool enabled )
```

Sets the state of the ready button on the UI to the parameter one

Parameters

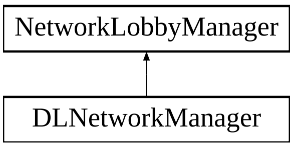
enabled	The state of the button

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Networking/DLNetworkLobbyPlayer.cs

3.18 DLNetworkManager Class Reference

Inheritance diagram for DLNetworkManager:



Public Member Functions

- void OnPlayerNumberModified (int count)

  Updates the playerCount variable by adding the parameter
- override bool OnLobbyServerSceneLoadedForPlayer (GameObject lobbyPlayer, GameObject gamePlayer)

  A callback for when all players are ready and the game is about to start. It takes each lobby player and applies the saved data of those to the actual game players
- override void **OnLobbyServerDisconnect** (NetworkConnection conn)
- override GameObject OnLobbyServerCreateLobbyPlayer (NetworkConnection conn, short playerController↩ Id)

  Callback what the server has to do once it creates a lobby player The server instantiates the player and toggles relevant UI for all players
- override void OnClientError (NetworkConnection conn, int errorCode)

  Callback for handling client errors. It currently only sends the player out of the lobby.

Public Attributes

- int **playerCount** = 0

3.18.1 Member Function Documentation

3.18.1.1 OnClientError()

```
override void DLNetworkManager.OnClientError (
              NetworkConnection conn,
              int errorCode )
```

Callback for handling client errors. It currently only sends the player out of the lobby.

Parameters

conn	The network connection
errorCode	The error code

3.18.1.2 OnLobbyServerCreateLobbyPlayer()

```
override GameObject DLNetworkManager.OnLobbyServerCreateLobbyPlayer (
              NetworkConnection conn,
              short playerControllerId )
```

Callback what the server has to do once it creates a lobby player The server instantiates the player and toggles relevant UI for all players

Parameters

conn	The network connection. Currently not used
player↩ ControllerId	The local player contoller Id. Currently not used

Returns

The instantiated lobby player object

3.18.1.3 OnLobbyServerSceneLoadedForPlayer()

```
override bool DLNetworkManager.OnLobbyServerSceneLoadedForPlayer (
             GameObject lobbyPlayer,
             GameObject gamePlayer )
```

A callback for when all players are ready and the game is about to start. It takes each lobby player and applies the saved data of those to the actual game players

Parameters

lobbyPlayer	The lobby player
gamePlayer	The game player that we are transferring data to

Returns

3.18.1.4 OnPlayerNumberModified()

```
void DLNetworkManager.OnPlayerNumberModified (
             int count )
```

Updates the playerCount variable by adding the parameter

Parameters

count	The amount of new players

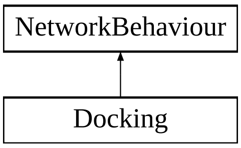
The documentation for this class was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Networking/DLNetworkManager.cs

3.19 Docking Class Reference

Handles the DockingKit interactions for each Player.

Inheritance diagram for Docking:



Public Member Functions

- override void OnStartClient ()

  Called when this object is activated on a client. Sets up the initial state and references.
- override void OnStartLocalPlayer ()

  Called when the local player object has been set up. Sets up local references.
- void CmdSetDockingKitId (DockingKitId newKit)

  Command which sets the SyncVar "dockingKitId". This is synchronized to all clients.
- void CmdSetSwitchState (bool state)

  Command for setting the state of the switchingKit member. switchingKit determines whether we want to spawn a pickup on undocking or not
- void RpcSetSwitchState (bool state)

  ClientRpc for synchronizing the switchingKit state
- void SetDockingKit (DockingKitId newKitId)

  Spawns in the DockingKit locally for the given new DockingKitId. Updates UI when called for local player.
- DockingKit GetDockingKit ()

  Get the active DockingKit for this Docking.
- void SetDockingKitStats (DockingKit kit)

  Updates the stats given by the current DockingKit.
- void SetPlayerInputRestriction (bool state, params InputType[ ] inputTypes)

  Passes the parameters along to the PlayerInput if called by the local player.
- void OnDockingButtonDown ()

  Called when the dock button is pressed.
- void CmdOnPlayerDocking (GameObject pickup)

  Command called when the local player wants to dock to a DockingKitPickup.
- void OnUndockingButtonDown ()

  Called when the undock button is pressed.
- void OnAbilityButtonChange (int abilityId, bool down)

  Called when the ability button is initially pressed or released.
- void CancelAbilities ()

  Cancels all the abilities in the current docking kit.
- void CmdSetActive (int abilityId, bool state)

  Command for activating an ability. Synchronizes activation to all clients.
- void RpcSetActive (int abilityId, bool state)

  ClientRpc for activating an ability. Runs locally on every client. Returns immediately for the local player, as the activation already happened locally.
- void CmdSpawnObject (int abilityId, int prefabId, Vector3 position, Vector3 rotation)

  Command for spawning prefab objects. Used by the abilities.
- void CmdSpawnObjectReference (int abilityId, int prefabId, Vector3 position, Vector3 rotation)

  Command for spawning prefab objects. Used by the abilities. Returns a reference to the spawned GameObject to the client/ability that called the Command.
- void TargetSetSpawnObjectReference (NetworkConnection connection, GameObject spawnedObject, int abilityId)

  TargetRpc for getting the reference to a spawned object.
- void CmdDestroyObject (GameObject destroyGameObject)

  Command used to destroy objects by objects that don't have authority to Command themselves.
- void CmdSpawnDockingKitPickup (DockingKitId kitId)

  Command for spawning docking kit pickup on undocking.
- void CmdSetModifier (int abilityId, int modifierId, bool apply)

  Command called by abilities by the local player to apply or remove a modifier.
- void SetModifier (int abilityId, bool state)

  Called by Modifiers OnClient functions to change the state of the modifier on each client.

- void CmdServerCallback (int abilityId, int functionId)

  Command used by abilities to run code on the server, as they're not NetworkBehaviour (or has authority) to call commands.
- void **CmdServerCallbackTwoVector3** (int abilityId, int functionId, Vector3 firstVec3, Vector3 secondVec3)
- void **CmdServerCallbackGameObject** (int abilityId, int functionId, GameObject go)
- void **CmdServerCallbackFloat** (int abilityId, int functionId, float param)
- void **CmdServerCallbackGameObjectFloat** (int abilityId, int functionId, GameObject param1, float param2)
- void RpcClientCallback (int abilityId, int functionId)

  ClientRpc used by abilities to run code on every client, as they're not NetworkBehaviour (or has authority) to call client rpcs.
- void **RpcClientCallbackVector3** (int abilityId, int functionId, Vector3 firstVec3)
- void **RpcClientCallbackTwoVector3** (int abilityId, int functionId, Vector3 firstVec3, Vector3 secondVec3)
- void **RpcClientCallbackGameObject** (int abilityId, int functionId, GameObject go)
- void **RpcClientCallbackFloat** (int abilityId, int functionId, float param)
- void TargetClientCallback (NetworkConnection connection, int abilityId, int functionId)

  TargetRpc used by abilities to run code on a target client, as they're not NetworkBehaviour (or has authority) to call target rpcs.
- void TargetReduceCooldown (NetworkConnection connection, int abilityId, float reductionAmount)

  TargetRpc for reducing the cooldown an ability by a certain amount.
- void **CmdSetElement** (ElementalContainer.ComboableElements element, int abilityId)
- void **RpcSetElement** (ElementalContainer.ComboableElements element, int abilityId)

Public Attributes

- GameObject **dockingKitPickupPrefab**
- DockingKit **basicDockingKit**
- float **dockingTime** = 2f
- DockingKitId **dockingKitId** = DockingKitId.Empty

3.19.1 Detailed Description

Handles the DockingKit interactions for each Player.

3.19.2 Member Function Documentation

3.19.2.1 CancelAbilities()

```
void Docking.CancelAbilities ( )
```

Cancels all the abilities in the current docking kit.

3.19.2.2 CmdDestroyObject()

```
void Docking.CmdDestroyObject (
             GameObject destroyGameObject )
```

Command used to destroy objects by objects that don't have authority to Command themselves.

Parameters

destroyGameObject	The reference to the object to be destroyed.

3.19.2.3 CmdOnPlayerDocking()

```
void Docking.CmdOnPlayerDocking (
             GameObject pickup )
```

Command called when the local player wants to dock to a DockingKitPickup.

Parameters

pickup	Reference to the networked pickup object.

3.19.2.4 CmdServerCallback()

```
void Docking.CmdServerCallback (
             int abilityId,
             int functionId )
```

Command used by abilities to run code on the server, as they're not NetworkBehaviour (or has authority) to call commands.

Parameters

abilityId	The id of the ability calling the command.
functionId	The id of the function to be run on the server.

3.19.2.5 CmdSetActive()

```
void Docking.CmdSetActive (
             int abilityId,
             bool state )
```

Command for activating an ability. Synchronizes activation to all clients.

Parameters

abilityId	Index of the ability to activate.
state	If the ability should be activated or deactivated.

3.19.2.6 CmdSetDockingKitId()

```
void Docking.CmdSetDockingKitId (
            DockingKitId newKit )
```

Command which sets the SyncVar "dockingKitId". This is synchronized to all clients.

Parameters

newKit	The new DockingKitId.

3.19.2.7 CmdSetModifier()

```
void Docking.CmdSetModifier (
            int abilityId,
            int modifierId,
            bool apply )
```

Command called by abilities by the local player to apply or remove a modifier.

Parameters

abilityId	The id of the ability that applied the modifier.
modifierIndex	The index of the modifier.
apply	If the modifier should be applied or removed.

3.19.2.8 CmdSetSwitchState()

```
void Docking.CmdSetSwitchState (
            bool state )
```

Command for setting the state of the switchingKit member. switchingKit determines whether we want to spawn a pickup on undocking or not

Parameters

state	The new state of the bool

3.19.2.9 CmdSpawnDockingKitPickup()

```
void Docking.CmdSpawnDockingKitPickup (
            DockingKitId kitId )
```

Command for spawning docking kit pickup on undocking.

Parameters

kit↩ Id	Which docking kit to spawn.

3.19.2.10 CmdSpawnObject()

```
void Docking.CmdSpawnObject (
             int abilityId,
             int prefabId,
             Vector3 position,
             Vector3 rotation )
```

Command for spawning prefab objects. Used by the abilities.

Parameters

abilityId	Index of the ability calling the Command.
prefab↩ Id	Index of the prefab to spawn from the ability.
position	Position of the new object.
rotation	Orientation of the new object (in eulerAngles).

3.19.2.11 CmdSpawnObjectReference()

```
void Docking.CmdSpawnObjectReference (
             int abilityId,
             int prefabId,
             Vector3 position,
             Vector3 rotation )
```

Command for spawning prefab objects. Used by the abilities. Returns a reference to the spawned GameObject to the client/ability that called the Command.

Parameters

abilityId	Index of the ability calling the Command.
prefab↩ Id	Index of the prefab to spawn from the ability.
position	Position of the new object.
rotation	Orientation of the new object (in eulerAngles).

3.19.2.12 GetDockingKit()

```
DockingKit Docking.GetDockingKit ( )
```

Get the active DockingKit for this Docking.

Returns

The current DockingKit.

3.19.2.13 OnAbilityButtonChange()

```
void Docking.OnAbilityButtonChange (
             int abilityId,
             bool down )
```

Called when the ability button is initially pressed or released.

Parameters

ability↩ Id	Index of the ability where the button state changed.
down	If this was the initial press.

3.19.2.14 OnDockingButtonDown()

```
void Docking.OnDockingButtonDown ( )
```

Called when the dock button is pressed.

3.19.2.15 OnStartClient()

```
override void Docking.OnStartClient ( )
```

Called when this object is activated on a client. Sets up the initial state and references.

3.19.2.16 OnStartLocalPlayer()

```
override void Docking.OnStartLocalPlayer ( )
```

Called when the local player object has been set up. Sets up local references.

3.19.2.17 OnUndockingButtonDown()

```
void Docking.OnUndockingButtonDown ( )
```

Called when the undock button is pressed.

3.19.2.18 RpcClientCallback()

```
void Docking.RpcClientCallback (
             int abilityId,
             int functionId )
```

ClientRpc used by abilities to run code on every client, as they're not NetworkBehaviour (or has authority) to call client rpcs.

Parameters

abilityId	The id of the ability calling the rpc.
function↩ Id	The id of the function to be run on every client.

3.19.2.19 RpcSetActive()

```
void Docking.RpcSetActive (
             int abilityId,
             bool state )
```

ClientRpc for activating an ability. Runs locally on every client. Returns immediately for the local player, as the activation already happened locally.

Parameters

ability↩ Id	Index of the ability to activate.
state	If the ability should be activated or deactivated.

3.19.2.20 RpcSetSwitchState()

```
void Docking.RpcSetSwitchState (
             bool state )
```

ClientRpc for synchronizing the switchingKit state

Parameters

state	The new state of the bool

3.19.2.21 SetDockingKit()

```
void Docking.SetDockingKit (
             DockingKitId newKitId )
```

Spawns in the DockingKit locally for the given new DockingKitId. Updates UI when called for local player.

Parameters

new↩ KitId	The new DockingKitId.

3.19.2.22 SetDockingKitStats()

```
void Docking.SetDockingKitStats (
             DockingKit kit )
```

Updates the stats given by the current DockingKit.

Parameters

kit	Which DockingKit to retrieve the stats from.

3.19.2.23 SetModifier()

```
void Docking.SetModifier (
             int abilityId,
             bool state )
```

Called by Modifiers OnClient functions to change the state of the modifier on each client.

Parameters

ability↩ Id	The id of the ability that applied the modifier.
state	The active state of the modifier.

3.19.2.24 SetPlayerInputRestriction()

```
void Docking.SetPlayerInputRestriction (
             bool state,
             params InputType [] inputTypes )
```

Passes the parameters along to the PlayerInput if called by the local player.

Parameters

state	The new state of the input restriction.
inputTypes	The types to set restriction for.

3.19.2.25 TargetClientCallback()

```
void Docking.TargetClientCallback (
             NetworkConnection connection,
             int abilityId,
             int functionId )
```

TargetRpc used by abilities to run code on a target client, as they're not NetworkBehaviour (or has authority) to call target rpcs.

Parameters

connection	Needed so TargetRpc finds the correct client.
abilityId	The id of the ability calling the target rpc.
functionId	The id of the function to be run on the targeted client.

3.19.2.26 TargetReduceCooldown()

```
void Docking.TargetReduceCooldown (
            NetworkConnection connection,
            int abilityId,
            float reductionAmount )
```

TargetRpc for reducing the cooldown an ability by a certain amount.

Parameters

connection	The NetworkConnection associated with the player given the reduction.
abilityId	The id of the ability to get cooldown reduction.
reductionAmount	The amount deducted for the current cooldown.

3.19.2.27 TargetSetSpawnObjectReference()

```
void Docking.TargetSetSpawnObjectReference (
            NetworkConnection connection,
            GameObject spawnedObject,
            int abilityId )
```

TargetRpc for getting the reference to a spawned object.

Parameters

connection	Needed so TargetRpc finds the correct client.
spawnedObject	Reference to the GameObject spawned.
abilityId	The id of the ability that called the spawn command.

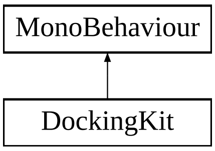
The documentation for this class was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Player/Docking.cs

3.20 DockingKit Class Reference

Handles the interaction between the Docking and the abilities.

Inheritance diagram for DockingKit:



Public Member Functions

- void Initialize (Docking dock)

  Initialization that happens locally on every client.
- void OnLocalPlayerInitialization (PlayerUIHandler playerUIHandler)

  Initialization that only happens for the local player (Player controlling this docking kit).
- void OnLocalPlayerDocking (float dockingTime, PlayerUIHandler playerUIHandler)

  Initialization called for the local player (Player controlling this docking kit) on docking.
- void OnUndocking (float dockingDuration, DockingKitId spawnPickupId, bool spawnPickup=true)

  Called for every client when undocking.
- void OnAbilityButtonChange (int abilityId, bool down)

  Called when the ability button is initially pressed or released.
- void CancelAbilities ()

  Cancels all the abilities in this docking kit.
- void SetAbilityLock (bool state, params int[ ] abilityNumbers)

  Used by Abilities to lock abilities in this docking kit.

Public Attributes

- float **moveSpeed** = 60f
- float **rotationSpeed** = 6f
- float **maxHealth** = 100f
- List< Ability > **abilities**

3.20.1 Detailed Description

Handles the interaction between the Docking and the abilities.

3.20.2 Member Function Documentation

3.20.2.1 CancelAbilities()

```
void DockingKit.CancelAbilities ( )
```

Cancels all the abilities in this docking kit.

3.20.2.2 Initialize()

```
void DockingKit.Initialize (
            Docking dock )
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.

3.20.2.3 OnAbilityButtonChange()

```
void DockingKit.OnAbilityButtonChange (
             int abilityId,
             bool down )
```

Called when the ability button is initially pressed or released.

ButtonDown may be called without ButtonUp running afterwards, handle this in Ability.CancelAbility (if the ability is locked in between). ButtonUp may be called without ButtonDown running first (if the ability is unlocked in between).

Parameters

abilityId	Index of the ability where the button state changed.
down	If this was the initial press.

3.20.2.4 OnLocalPlayerDocking()

```
void DockingKit.OnLocalPlayerDocking (
             float dockingTime,
             PlayerUIHandler playerUIHandler )
```

Initialization called for the local player (Player controlling this docking kit) on docking.

Parameters

dockingTime	The time used to dock. (Immobile duration)

3.20.2.5 OnLocalPlayerInitialization()

```
void DockingKit.OnLocalPlayerInitialization (
             PlayerUIHandler playerUIHandler )
```

Initialization that only happens for the local player (Player controlling this docking kit).

3.20.2.6 OnUndocking()

```
void DockingKit.OnUndocking (
             float dockingDuration,
             DockingKitId spawnPickupId,
             bool spawnPickup = true )
```

Called for every client when undocking.

Parameters

dockingDuration	The time used to undock. (Immobile duration)
spawnPickupId	The DockingKitId of the pickup to be spawned on undocking.

3.20.2.7 SetAbilityLock()

```
void DockingKit.SetAbilityLock (
            bool state,
            params int [] abilityNumbers )
```

Used by Abilities to lock abilities in this docking kit.

Parameters

state	To lock or unlock.
abilityNumbers	Toggles lock for these abilities.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/DockingKit.cs

3.21 DockingKitDescriptions Struct Reference

Public Attributes

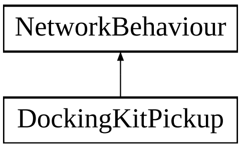
- Sprite **icon**
- string **name**
- string **description**

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/ShopItemData.cs

3.22 DockingKitPickup Class Reference

Inheritance diagram for DockingKitPickup:



Public Member Functions

- override void OnStartClient ()

  Calls the SyncVar hook manually to get the correct initial state. Used by clients connecting after pickup already spawned.
- void OnPlayerDocking (GameObject player)

  Server call from the Docking called when a player tries to dock.

Public Attributes

- DockingKitId **dockingKitId** = DockingKitId.Empty

3.22.1 Member Function Documentation

3.22.1.1 OnPlayerDocking()

```
void DockingKitPickup.OnPlayerDocking (
             GameObject player )
```

Server call from the Docking called when a player tries to dock.

Parameters

player	Reference to the player docking.

3.22.1.2 OnStartClient()

```
override void DockingKitPickup.OnStartClient ( )
```

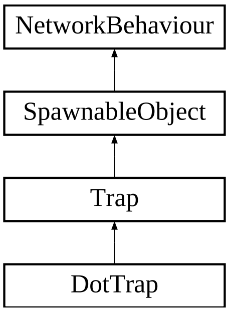
Calls the SyncVar hook manually to get the correct initial state. Used by clients connecting after pickup already spawned.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/DockingKitPickup.cs

3.23 DotTrap Class Reference

Inheritance diagram for DotTrap:



Public Member Functions

- override void HandleTrigger (PlayerStatus playerStatus)

  Callback that allows this trap to do whatever it wants whenever it is triggered. This one simply applies the member structs containing modifier info

Public Attributes

- ModifierInfo **dotInfo**
- ModifierInfo **slowInfo**

Additional Inherited Members

3.23.1 Member Function Documentation

3.23.1.1 HandleTrigger()

```
override void DotTrap.HandleTrigger (
            PlayerStatus playerStatus )  [virtual]
```

Callback that allows this trap to do whatever it wants whenever it is triggered. This one simply applies the member structs containing modifier info

Parameters

playerStatus	The PlayerStatus component of the player that is in the trap

Reimplemented from Trap.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TrapperKit/DotTrap.cs

3.24 ElementalModifiers Class Reference

Public Member Functions

- void **Initialize** ()
- void **SetModifier** (bool state)
- ModifierInfo **GetModifierInfo** (int modifierId)
- void TransferElementalModifier (Collider other, Docking docking, int abilityId)

  Handles the transferring of the elemental buff by applying it as a debuff to the player that was hit
- void **ApplyElement** (ElementalContainer.ComboableElements element, Docking docking, int abilityId)
- void **SetElement** (ElementalContainer.ComboableElements element)

Public Attributes

- Transform **elementEffectTransform**
- ModifierInfo **fireBuff**
- ModifierInfo **iceBuff**
- ModifierInfo **electricBuff**
- ModifierInfo **fireDebuff**
- ModifierInfo **iceDebuff**
- ModifierInfo **electricDebuff**

3.24.1 Member Function Documentation

3.24.1.1 TransferElementalModifier()

```
void ElementalModifiers.TransferElementalModifier (
             Collider other,
             Docking docking,
             int abilityId )
```

Handles the transferring of the elemental buff by applying it as a debuff to the player that was hit

Parameters

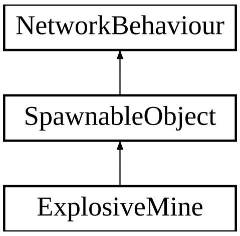
other	The collider we want to apply the debuff to

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/ElementalModifiers.cs

3.25 ExplosiveMine Class Reference

Inheritance diagram for ExplosiveMine:



Public Member Functions

- void **Initialize** (GameObject owner)
- void **Start** ()
- void **OnTriggerEnter** (Collider other)
- void RpcRemoveMine ()

  Destroy the mine and remove it from the list of mines.

Public Attributes

- float **baseDamage**
- float **maxDamageTapering**
- float **explosionForce**
- float **explosionRadius**
- float **activationTime**
- int **myId**
- GameObject **spawnerReference**

Additional Inherited Members

3.25.1 Member Function Documentation

3.25.1.1 RpcRemoveMine()

```
void ExplosiveMine.RpcRemoveMine ( )
```

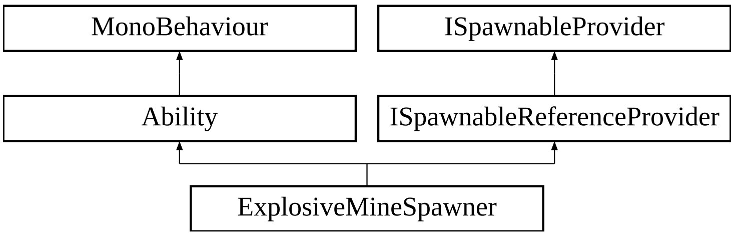
Destroy the mine and remove it from the list of mines.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BomberKit/ExplosiveMine.cs

3.26 ExplosiveMineSpawner Class Reference

Inheritance diagram for ExplosiveMineSpawner:



Public Member Functions

- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.
- void RemoveMine (int mineId)

  Removes the mine that got triggered
- void OnDestroy ()

  Clean up mines when docking kit is not equipped anymore.

Public Attributes

- string **animatorTrigger**
- GameObject [ ] **minePrefab**
- int **maxMineAmount**

Additional Inherited Members

3.26.1 Member Function Documentation

3.26.1.1 ButtonDown()

```
override void ExplosiveMineSpawner.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.26.1.2 OnDestroy()

```
void ExplosiveMineSpawner.OnDestroy ( )
```

Clean up mines when docking kit is not equipped anymore.

3.26.1.3 RemoveMine()

```
void ExplosiveMineSpawner.RemoveMine (
             int mineId )
```

Removes the mine that got triggered

Parameters

mine↩ Id	The ID of the mine.

3.26.1.4 SetActive()

```
override void ExplosiveMineSpawner.SetActive (
             bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

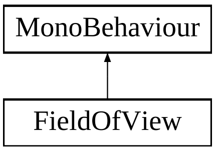
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BomberKit/ExplosiveMineSpawner.cs

3.27 FieldOfView Class Reference

Inheritance diagram for FieldOfView:



Public Member Functions

- Vector3 **DirFromAngle** (float angleInDegrees, bool angleIsGlobal)
- void **SetViewRadius** (float newRadius, float speed)
- void **ResetViewRadius** (float speed)
- void **SetViewAngle** (float newAngle, float speed)
- void **ResetViewAngle** (float speed)

Public Attributes

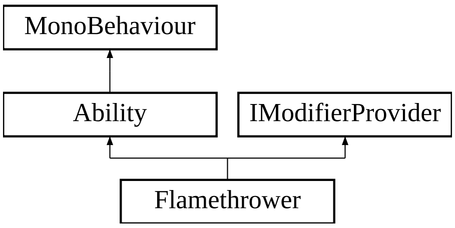
- float **viewRadius**
- float **viewAngle**
- LayerMask **obstacleMask**
- float **meshResolution** = 1
- int **edgeResolveIterations** = 1
- float **edgeDstThreshold** = 0.5f
- float **maskCutawayDst** = 0.4f
- MeshFilter **viewMeshFilter**

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Player/FieldOfView.cs

3.28 Flamethrower Class Reference

Inheritance diagram for Flamethrower:



Public Member Functions

- override void Initialize (Docking dock, Animator anim, int abId)

  Initialization that happens locally on every client.
- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.
- override void SetModifier (bool state)

  Called by the Modifier. Appropriate place for doing local changes.
- void SetBuffState (bool state)

  Sets the visual state of the flamethrower to the given parameter state

Public Attributes

- SpriteRenderer **head**
- ModifierInfo **buff**
- ModifierInfo **dot**
- GameObject **flamethrowerContainer**
- Color **headColorWhileActive**

Additional Inherited Members

3.28.1 Member Function Documentation

3.28.1.1 ButtonDown()

```
override void Flamethrower.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.28.1.2 Initialize()

```
override void Flamethrower.Initialize (
            Docking dock,
            Animator anim,
            int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented from Ability.

3.28.1.3 SetActive()

```
override void Flamethrower.SetActive (
             bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

Implements Ability.

3.28.1.4 SetBuffState()

```
void Flamethrower.SetBuffState (
             bool state )
```

Sets the visual state of the flamethrower to the given parameter state

Parameters

state	The state of the flamethrower

3.28.1.5 SetModifier()

```
override void Flamethrower.SetModifier (
             bool state )  [virtual]
```

Called by the Modifier. Appropriate place for doing local changes.

Parameters

state	If the modifier should be activated or deactivated.

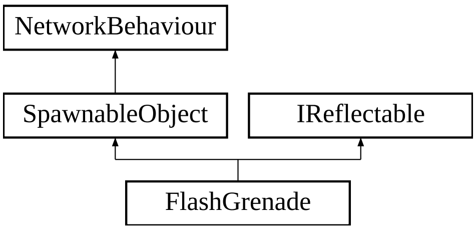
Reimplemented from Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TrapperKit/Flamethrower.cs

3.29 FlashGrenade Class Reference

Inheritance diagram for FlashGrenade:



Public Attributes

- float **timeBeforeExplosion** = 2f
- float **initialSpeed** = 5f
- SphereCollider **explosionCollider**
- GameObject **visuals**
- int **lifeTimeAfterExplosion** = 1
- float **visionRadius** = 20
- float **lerpSpeed** = 10f
- ParticleSystem **explosionParticles**
- ModifierInfo **stunInfo**

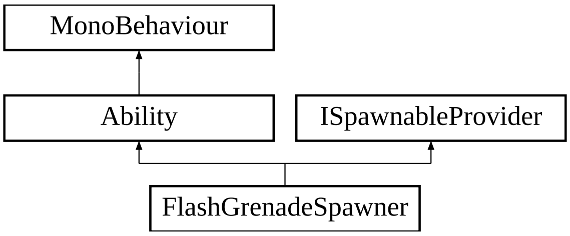
Additional Inherited Members

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BrawlerKit/FlashGrenade.cs

3.30 FlashGrenadeSpawner Class Reference

Inheritance diagram for FlashGrenadeSpawner:



Public Member Functions

- override void ButtonDown ()

  Callback for what this ability should do once its associated button has been pressed
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Public Attributes

- GameObject **flashGrenadePrefab**
- float **offset** = 5

Additional Inherited Members

3.30.1 Member Function Documentation

3.30.1.1 ButtonDown()

```
override void FlashGrenadeSpawner.ButtonDown ( )  [virtual]
```

Callback for what this ability should do once its associated button has been pressed

Implements Ability.

3.30.1.2 SetActive()

```
override void FlashGrenadeSpawner.SetActive (
             bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

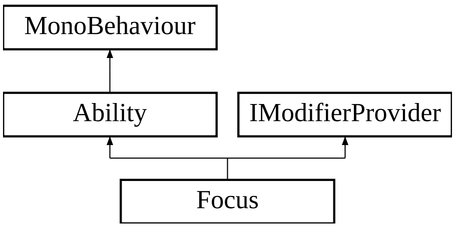
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BrawlerKit/FlashGrenadeSpawner.cs

3.31 Focus Class Reference

Inheritance diagram for Focus:



Public Member Functions

- override void InitializeLocalPlayer (AbilityUI abilityUI)

  Initialization that only happens for the local player (Player controlling this ability). Called after Initialize, so the references are already set up.
- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void CancelAbility ()

  Call for cancelling abilities. Override in abilities that may be interrupted.
- override void SetActive (bool state)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Public Attributes

- string **animatorBool**
- float **maxDuration** = 15f
- Transform **target**
- float **targetOrthoSize**
- Slingshot **slingshot**
- Transform **leftSlingHandle**
- Transform **rightSlingHandle**
- ModifierInfo **focusModifier**
- float **targetViewAngle**
- float **targetViewRadius**
- float **lerpSpeed**

Additional Inherited Members

3.31.1 Member Function Documentation

3.31.1.1 ButtonDown()

```
override void Focus.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.31.1.2 CancelAbility()

```
override void Focus.CancelAbility ( )  [virtual]
```

Call for cancelling abilities. Override in abilities that may be interrupted.

Reimplemented from Ability.

3.31.1.3 InitializeLocalPlayer()

```
override void Focus.InitializeLocalPlayer (
            AbilityUI abilityUI )  [virtual]
```

Initialization that only happens for the local player (Player controlling this ability). Called after Initialize, so the references are already set up.

Reimplemented from Ability.

3.31.1.4 SetActive()

```
override void Focus.SetActive (
            bool state )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

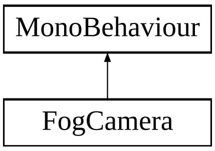
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/SniperKit/Focus.cs

3.32 FogCamera Class Reference

Inheritance diagram for FogCamera:



Public Attributes

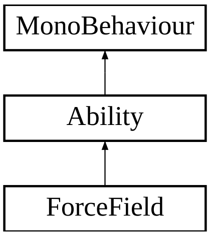
- Shader **replacementShader**
- Color **fogColor**

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/FogCamera.cs

3.33 ForceField Class Reference

Inheritance diagram for ForceField:



Public Member Functions

- override void Initialize (Docking dock, Animator anim, int abId)

    Initialization that happens locally on every client.
- override void ButtonDown ()

    Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state=false)

    Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Public Attributes

- float **playerForce** = 10f
- string **animatorTrigger**

Additional Inherited Members

3.33.1 Member Function Documentation

3.33.1.1 ButtonDown()

```
override void ForceField.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.33.1.2 Initialize()

```
override void ForceField.Initialize (
             Docking dock,
             Animator anim,
             int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented from Ability.

3.33.1.3 SetActive()

```
override void ForceField.SetActive (
            bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

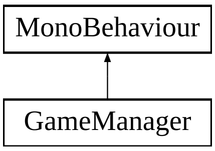
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TankKit/ForceField.cs

3.34 GameManager Class Reference

Inheritance diagram for GameManager:



Public Member Functions

- GameObject GetDockingKit (DockingKitId id)

  Used for retrieving a DockingKit prefab from a DockingKitId.

Public Attributes

- List< GameObject > **dockingKitPrefabs**
- PlayerUIHandler **playerUIHandler**

3.34.1 Member Function Documentation

3.34.1.1 GetDockingKit()

```
GameObject GameManager.GetDockingKit (
            DockingKitId id )
```

Used for retrieving a DockingKit prefab from a DockingKitId.

Parameters

id	Index of DockingKit to return.

Returns

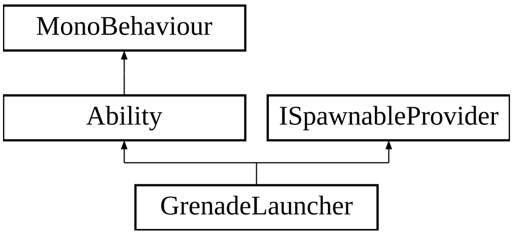
The DockingKit prefab for the given DockingKitId.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/GameManager.cs

3.35 GrenadeLauncher Class Reference

Inheritance diagram for GrenadeLauncher:



Public Member Functions

- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.
- void Fire ()

Public Attributes

- string **animatorTrigger**
- float **spawnOffset**
- GameObject **shellPrefab**

Additional Inherited Members

3.35.1 Member Function Documentation

3.35.1.1 ButtonDown()

```
override void GrenadeLauncher.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.35.1.2 Fire()

```
void GrenadeLauncher.Fire ( )
```

3.35.1.3 SetActive()

```
override void GrenadeLauncher.SetActive (
             bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

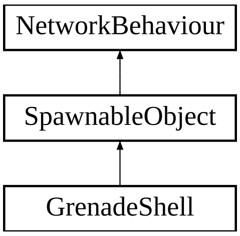
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BomberKit/GrenadeLauncher.cs

3.36 GrenadeShell Class Reference

Inheritance diagram for GrenadeShell:



Public Attributes

- float **launchForce**
- float **lifetime**
- float **explosionRadius**
- float **explosionForce**
- float **baseDaamge**

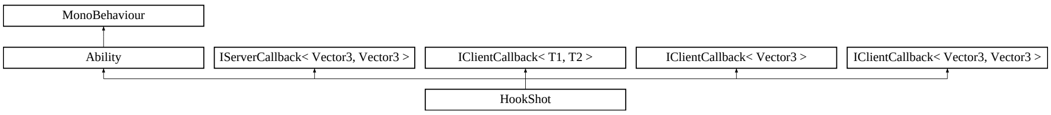
Additional Inherited Members

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BomberKit/GrenadeShell.cs

3.37 HookShot Class Reference

Inheritance diagram for HookShot:


MonoBehaviour
Ability
IServerCallback< Vector3, Vector3 >
IClientCallback< T1, T2 >
IClientCallback< Vector3 >
IClientCallback< Vector3, Vector3 >
HookShot

Public Member Functions

- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Public Attributes

- string **animatorBool**
- Transform **hookSpawnPoint**
- Collider **hook**
- LineRenderer **lineRenderer**
- float **hookSpeed** = 40f
- float **hookReturnSpeed** = 30f
- float **hookRange** = 80f
- float **hookPullForce** = 5.8f
- float **hookOnHitHoldTime** = 0.5f
- ModifierInfo **hookModifier**

Additional Inherited Members

3.37.1 Member Function Documentation

3.37.1.1 ButtonDown()

```
override void HookShot.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.37.1.2 SetActive()

```
override void HookShot.SetActive (
             bool state )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

Implements Ability.

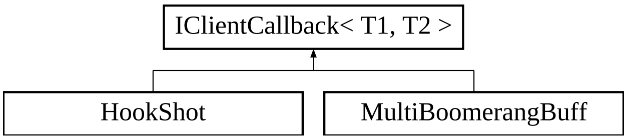
The documentation for this class was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TankKit/HookShot.cs

3.38 IClientCallback< T1, T2 > Interface Template Reference

Can recieve client callbacks from the Docking with two parameters.

Inheritance diagram for IClientCallback< T1, T2 >:



Public Member Functions

- void ClientCallback (int functionId)

  Called from the Docking to give abilities a way to run code on every client.
- void **ClientCallback** (int functionId, T param)
- void **ClientCallback** (int functionId, T1 first, T2 second)

3.38.1 Detailed Description

Can recieve client callbacks from the Docking with two parameters.

3.38.2 Member Function Documentation

3.38.2.1 ClientCallback()

```
void IClientCallback< T1, T2 >.ClientCallback (
             int functionId )
```

Called from the Docking to give abilities a way to run code on every client.

Parameters

function↩ Id	The id of the function to be run on every client.

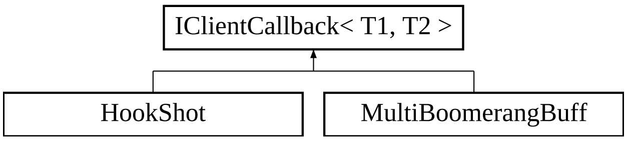
The documentation for this interface was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/Ability/IClientCallback.cs

3.39 IClientCallback< T1, T2 > Interface Template Reference

Can recieve client callbacks from the Docking with two parameters.

Inheritance diagram for IClientCallback< T1, T2 >:



Public Member Functions

- void ClientCallback (int functionId)

    Called from the Docking to give abilities a way to run code on every client.
- void **ClientCallback** (int functionId, T param)
- void **ClientCallback** (int functionId, T1 first, T2 second)

3.39.1 Detailed Description

Can recieve client callbacks from the Docking with two parameters.

3.39.2 Member Function Documentation

3.39.2.1 ClientCallback()

```
void IClientCallback< T1, T2 >.ClientCallback (
            int functionId )
```

Called from the Docking to give abilities a way to run code on every client.

Parameters

function↩ Id	The id of the function to be run on every client.

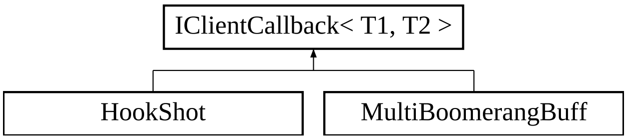
The documentation for this interface was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/Ability/IClientCallback.cs

3.40 IClientCallback< T1, T2 > Interface Template Reference

Can recieve client callbacks from the Docking with two parameters.

Inheritance diagram for IClientCallback< T1, T2 >:



Public Member Functions

- void ClientCallback (int functionId)

  Called from the Docking to give abilities a way to run code on every client.
- void **ClientCallback** (int functionId, T param)
- void **ClientCallback** (int functionId, T1 first, T2 second)

3.40.1 Detailed Description

Can recieve client callbacks from the Docking with two parameters.

3.40.2 Member Function Documentation

3.40.2.1 ClientCallback()

```
void IClientCallback< T1, T2 >.ClientCallback (
            int functionId )
```

Called from the Docking to give abilities a way to run code on every client.

Parameters

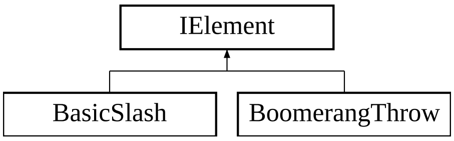
function← *Id*	The id of the function to be run on every client.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/Ability/IClientCallback.cs

3.41 IElement Interface Reference

Inheritance diagram for IElement:



Public Member Functions

- void **ApplyElement** (ElementalContainer.ComboableElements element)

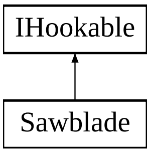
The documentation for this interface was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/IElement.cs

3.42 IHookable Interface Reference

Used by spawnables that can be hooked.

Inheritance diagram for IHookable:



Public Member Functions

- void Hooked (GameObject playerObject, Transform hook)

  Hooks the spawnable.

3.42.1 Detailed Description

Used by spawnables that can be hooked.

3.42.2 Member Function Documentation

3.42.2.1 Hooked()

```
void IHookable.Hooked (
             GameObject playerObject,
             Transform hook )
```

Hooks the spawnable.

Parameters

playerObject	The player that owns the hook.
hook	The hook transform.

Implemented in Sawblade.

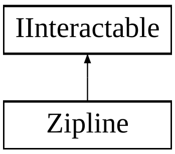
The documentation for this interface was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/IHookable.cs

3.43 IInteractable Interface Reference

Used by objects that can receive interaction calls from PlayerInput.

Inheritance diagram for IInteractable:



Public Member Functions

- void Interact (Player player)

  Called when the object is interacted with.

3.43.1 Detailed Description

Used by objects that can receive interaction calls from PlayerInput.

3.43.2 Member Function Documentation

3.43.2.1 Interact()

```
void IInteractable.Interact (
            Player player )
```

Called when the object is interacted with.

Parameters

player	Reference to the Player interacting.

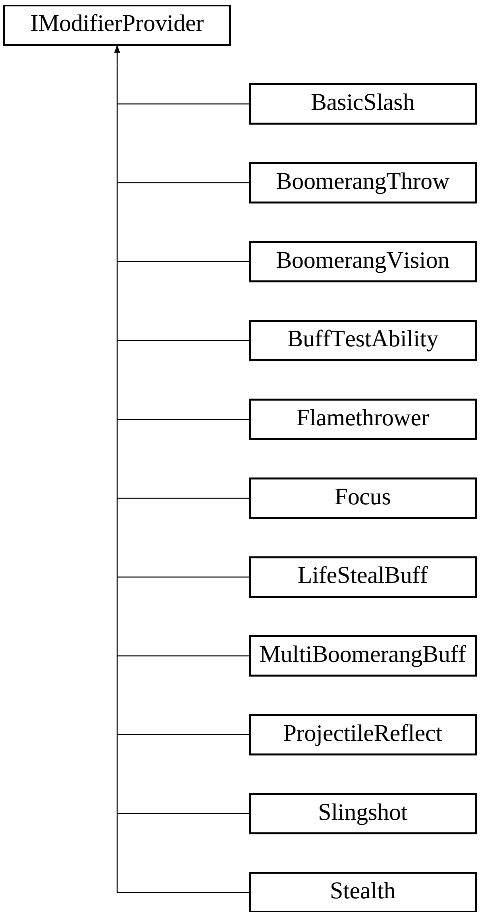
The documentation for this interface was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/IInteractable.cs

3.44 IModifierProvider Interface Reference

Can return reference to modifier info.

Inheritance diagram for IModifierProvider:



Public Member Functions

- ModifierInfo GetModifierInfo (int modifierId)

  Used by the Docking to get the correct modifier from the abilities. Parameter only used if the ability has a list of modifiers.

3.44.1 Detailed Description

Can return reference to modifier info.

3.44.2 Member Function Documentation

3.44.2.1 GetModifierInfo()

```
ModifierInfo IModifierProvider.GetModifierInfo (
             int modifierId )
```

Used by the Docking to get the correct modifier from the abilities. Parameter only used if the ability has a list of modifiers.

Parameters

modifier← Id	The Id of the modifier info.

Returns

Reference to the ModifierInfo.

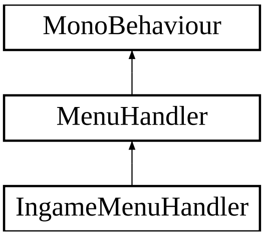
The documentation for this interface was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/Ability/IModifierProvider.cs

3.45 IngameMenuHandler Class Reference

Handles ingame menus like the Shop and "Pause" menu

Inheritance diagram for IngameMenuHandler:



Public Member Functions

- void OnShopDisplay ()

  Gets called whenever the player activates the Shop UI. Caches references to the local player if not already cached.
- void OnShopSelectionChange ()

  Handles the updating of the shop UI as different docking kits are selected
- void DisplayVerificationPrompt ()

  Displays the verification prompt for shop purchases
- void CompleteShopPurchase ()

  Completes a shop purchase and tells docking to switch kit
- void SetFirstSelectedShopObject ()

  Makes sure to set the selection of the first element in the shop as the menu is opened
- void SetLastSelectedShopObject ()

  Can be used when going back from menus like the verification prompt to set the last highlighted shop item as selected again
- void StopHost ()

  Simple function that calls the NetworkManager to disconnect from the game. Can be called from UI buttons using their OnClick interface in the editor
- void CheckPriceAndEquipAvailability ()

  Checks all shop item prices and adds a dark overlay to items that the player is unable to purchase. Also displays a "e" on the currently equipped docking kit

Public Attributes

- GameObject **pauseMenu**
- GameObject **shopMenu**
- GameObject **shopDescriptionsContainer**
- GameObject **shopItemPrefab**
- GameObject **purchaseVerificationPrompt**

Additional Inherited Members

3.45.1 Detailed Description

Handles ingame menus like the Shop and "Pause" menu

3.45.2 Member Function Documentation

3.45.2.1 CheckPriceAndEquipAvailability()

```
void IngameMenuHandler.CheckPriceAndEquipAvailability ( )
```

Checks all shop item prices and adds a dark overlay to items that the player is unable to purchase. Also displays a "e" on the currently equipped docking kit

3.45.2.2 CompleteShopPurchase()

```
void IngameMenuHandler.CompleteShopPurchase ( )
```

Completes a shop purchase and tells docking to switch kit

3.45.2.3 DisplayVerificationPrompt()

```
void IngameMenuHandler.DisplayVerificationPrompt ( )
```

Displays the verification prompt for shop purchases

3.45.2.4 OnShopDisplay()

```
void IngameMenuHandler.OnShopDisplay ( )
```

Gets called whenever the player activates the Shop UI. Caches references to the local player if not already cached.

3.45.2.5 OnShopSelectionChange()

```
void IngameMenuHandler.OnShopSelectionChange ( )
```

Handles the updating of the shop UI as different docking kits are selected

3.45.2.6 SetFirstSelectedShopObject()

```
void IngameMenuHandler.SetFirstSelectedShopObject ( )
```

Makes sure to set the selection of the first element in the shop as the menu is opened

3.45.2.7 SetLastSelectedShopObject()

```
void IngameMenuHandler.SetLastSelectedShopObject ( )
```

Can be used when going back from menus like the verification prompt to set the last highlighted shop item as selected again

3.45.2.8 StopHost()

```
void IngameMenuHandler.StopHost ( )
```

Simple function that calls the NetworkManager to disconnect from the game. Can be called from UI buttons using their OnClick interface in the editor

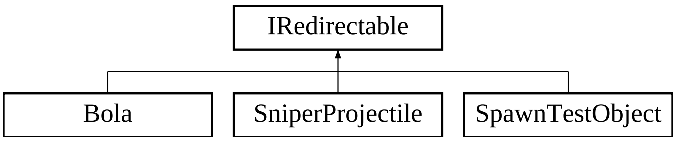
The documentation for this class was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/UI/IngameMenuHandler.cs

3.46 IRedirectable Interface Reference

Used by spawnables that can be redirected.

Inheritance diagram for IRedirectable:



Public Member Functions

- void RedirectDirection (Vector3 newDirection, int newPlayerId=-1)

  Redirects direction of the spawnable.

3.46.1 Detailed Description

Used by spawnables that can be redirected.

3.46.2 Member Function Documentation

3.46.2.1 RedirectDirection()

```
void IRedirectable.RedirectDirection (
              Vector3 newDirection,
              int newPlayerId = -1 )
```

Redirects direction of the spawnable.

Parameters

newDirection	The new direction.
newPlayerId	The player id of the new owner, -1 if current owner is kept.

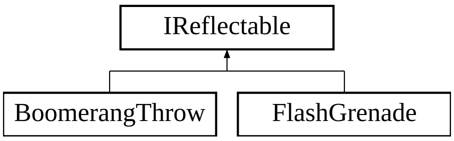
Implemented in SpawnTestObject.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/IRedirectable.cs

3.47 IReflectable Interface Reference

Inheritance diagram for IReflectable:



Public Member Functions

- void **ReflectVelocity** ()

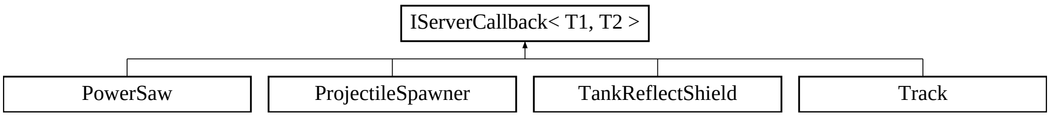
The documentation for this interface was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/IReflectable.cs

3.48 IServerCallback< T1, T2 > Interface Template Reference

Can recieve server callbacks from the Docking with two parameters.

Inheritance diagram for IServerCallback< T1, T2 >:



Public Member Functions

- void ServerCallback (int functionId)

  Called from the Docking to give abilities a way to run server code.
- void **ServerCallback** (int functionId, T param)
- void **ServerCallback** (int functionId, T1 first, T2 second)

3.48.1 Detailed Description

Can recieve server callbacks from the Docking with two parameters.

3.48.2 Member Function Documentation

3.48.2.1 ServerCallback()

```
void IServerCallback< T1, T2 >.ServerCallback (
              int functionId )
```

Called from the Docking to give abilities a way to run server code.

Parameters

function↩ Id	The id of the function to be run on the server.

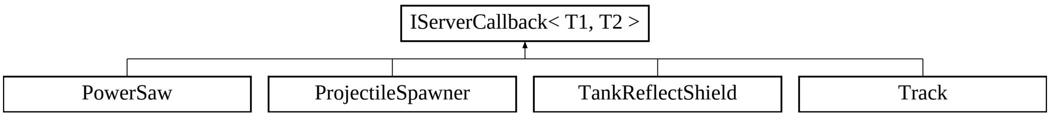
The documentation for this interface was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/Ability/IServerCallback.cs

3.49 IServerCallback< T1, T2 > Interface Template Reference

Can recieve server callbacks from the Docking with two parameters.

Inheritance diagram for IServerCallback< T1, T2 >:



Public Member Functions

- void ServerCallback (int functionId)

    Called from the Docking to give abilities a way to run server code.
- void **ServerCallback** (int functionId, T param)
- void **ServerCallback** (int functionId, T1 first, T2 second)

3.49.1 Detailed Description

Can recieve server callbacks from the Docking with two parameters.

3.49.2 Member Function Documentation

3.49.2.1 ServerCallback()

```
void IServerCallback< T1, T2 >.ServerCallback (
             int functionId )
```

Called from the Docking to give abilities a way to run server code.

Parameters

function↩ Id	The id of the function to be run on the server.

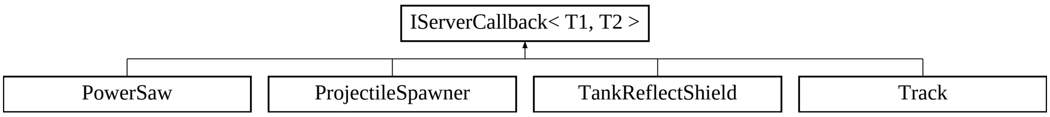
The documentation for this interface was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/Ability/IServerCallback.cs

3.50 IServerCallback< T1, T2 > Interface Template Reference

Can recieve server callbacks from the Docking with two parameters.

Inheritance diagram for IServerCallback< T1, T2 >:



Public Member Functions

- void ServerCallback (int functionId)

    Called from the Docking to give abilities a way to run server code.
- void **ServerCallback** (int functionId, T param)
- void **ServerCallback** (int functionId, T1 first, T2 second)

3.50.1 Detailed Description

Can recieve server callbacks from the Docking with two parameters.

3.50.2 Member Function Documentation

3.50.2.1 ServerCallback()

```
void IServerCallback< T1, T2 >.ServerCallback (
              int functionId )
```

Called from the Docking to give abilities a way to run server code.

Parameters

function↩ Id	The id of the function to be run on the server.

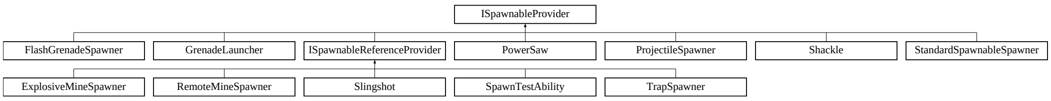
The documentation for this interface was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/Ability/IServerCallback.cs

3.51 ISpawnableProvider Interface Reference

Can return reference to a spawnable prefab.

Inheritance diagram for ISpawnableProvider:



Public Member Functions

- GameObject GetSpawnablePrefab (int spawnableId)

  Used by the Docking to get the correct prefab to spawn from the abilities. Parameter only used if the ability has a list of prefabs.

3.51.1 Detailed Description

Can return reference to a spawnable prefab.

3.51.2 Member Function Documentation

3.51.2.1 GetSpawnablePrefab()

```
GameObject ISpawnableProvider.GetSpawnablePrefab (
             int spawnableId )
```

Used by the Docking to get the correct prefab to spawn from the abilities. Parameter only used if the ability has a list of prefabs.

Parameters

spawnable↩ Id	The Id of the spawnable object.

Returns

Reference to the prefab GameObject.

Implemented in StandardSpawnableSpawner.

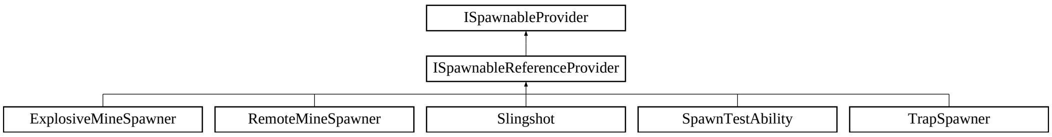
The documentation for this interface was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/Ability/ISpawnableProvider.cs

3.52 ISpawnableReferenceProvider Interface Reference

Can return reference to a spawnable prefab and catch the reference to the spawned object.

Inheritance diagram for ISpawnableReferenceProvider:



Public Member Functions

- void SetSpawnedObjectReference (GameObject spawnedObject)

  Called from the Docking to set up local references from spawned network objects.

3.52.1 Detailed Description

Can return reference to a spawnable prefab and catch the reference to the spawned object.

3.52.2 Member Function Documentation

3.52.2.1 SetSpawnedObjectReference()

```
void ISpawnableReferenceProvider.SetSpawnedObjectReference (
            GameObject spawnedObject )
```

Called from the Docking to set up local references from spawned network objects.

Parameters

spawnedObject	Reference to spawned object.

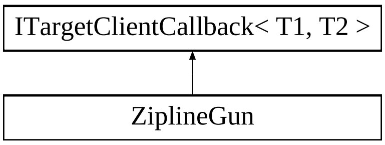
The documentation for this interface was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/Ability/ISpawnableProvider.cs

3.53 ITargetClientCallback< T1, T2 > Interface Template Reference

Can recieve target client callbacks from the Docking with two parameters.

Inheritance diagram for ITargetClientCallback< T1, T2 >:



Public Member Functions

- void TargetClientCallback (int functionId)

  Called from the Docking to give abilities a way to run code on a target client.
- void **TargetClientCallback** (int functionId, T param)
- void **TargetClientCallback** (int functionId, T1 first, T2 second)

3.53.1 Detailed Description

Can recieve target client callbacks from the Docking with two parameters.

3.53.2 Member Function Documentation

3.53.2.1 TargetClientCallback()

```
void ITargetClientCallback< T1, T2 >.TargetClientCallback (
            int functionId )
```

Called from the Docking to give abilities a way to run code on a target client.

Parameters

function←↩ Id	The id of the function to be run on the targeted client.

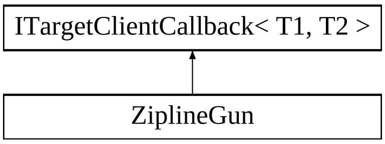
The documentation for this interface was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/Ability/ITargetClientCallback.cs

3.54 ITargetClientCallback< T1, T2 > Interface Template Reference

Can recieve target client callbacks from the Docking with two parameters.

Inheritance diagram for ITargetClientCallback< T1, T2 >:



Public Member Functions

- void TargetClientCallback (int functionId)

  Called from the Docking to give abilities a way to run code on a target client.
- void **TargetClientCallback** (int functionId, T param)
- void **TargetClientCallback** (int functionId, T1 first, T2 second)

3.54.1 Detailed Description

Can recieve target client callbacks from the Docking with two parameters.

3.54.2 Member Function Documentation

3.54.2.1 TargetClientCallback()

```
void ITargetClientCallback< T1, T2 >.TargetClientCallback (
             int functionId )
```

Called from the Docking to give abilities a way to run code on a target client.

Parameters

function↩ Id	The id of the function to be run on the targeted client.

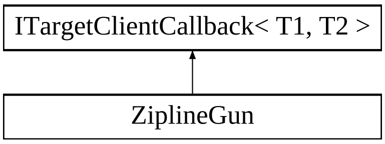
The documentation for this interface was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/Ability/ITargetClientCallback.cs

3.55 ITargetClientCallback< T1, T2 > Interface Template Reference

Can recieve target client callbacks from the Docking with two parameters.

Inheritance diagram for ITargetClientCallback< T1, T2 >:



Public Member Functions

- void TargetClientCallback (int functionId)

  Called from the Docking to give abilities a way to run code on a target client.
- void **TargetClientCallback** (int functionId, T param)
- void **TargetClientCallback** (int functionId, T1 first, T2 second)

3.55.1 Detailed Description

Can recieve target client callbacks from the Docking with two parameters.

3.55.2 Member Function Documentation

3.55.2.1 TargetClientCallback()

```
void ITargetClientCallback< T1, T2 >.TargetClientCallback (
            int functionId )
```

Called from the Docking to give abilities a way to run code on a target client.

Parameters

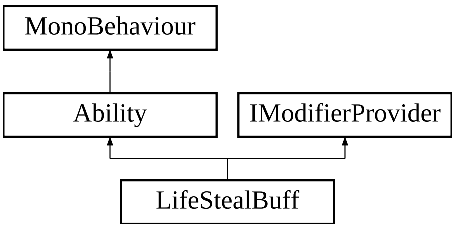
function← Id	The id of the function to be run on the targeted client.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Interfaces/Ability/ITargetClientCallback.cs

3.56 LifeStealBuff Class Reference

Inheritance diagram for LifeStealBuff:



Public Member Functions

- override void Initialize (Docking dock, Animator anim, int abId)

  Initialization that happens locally on every client.
- override void ButtonDown ()

  Callback for what this ability should do once its associated button has been pressed
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.
- override void SetModifier (bool state=false)

  Callback for what this ability should do when a new modifier state is set
- bool IsBuffActive ()

  A simple getter function for whether the life steal buff is currently active
- int **GetAbilityId** ()
- int **GetBuffModifierId** ()

Public Attributes

- float **damageMultiplier** = 1.5f
- float **healPercentage** = 0.5f
- SpriteRenderer [ ] **axeVisuals**
- ParticleSystem **activeParticles**
- Color **axeColorWhileActive**
- ModifierInfo **buff**

Additional Inherited Members

3.56.1 Member Function Documentation

3.56.1.1 ButtonDown()

```
override void LifeStealBuff.ButtonDown ( )  [virtual]
```

Callback for what this ability should do once its associated button has been pressed

Implements Ability.

3.56.1.2 Initialize()

```
override void LifeStealBuff.Initialize (
            Docking dock,
            Animator anim,
            int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented from Ability.

3.56.1.3 IsBuffActive()

```
bool LifeStealBuff.IsBuffActive ( )
```

A simple getter function for whether the life steal buff is currently active

Returns

Whether the buff is currently active

3.56.1.4 SetActive()

```
override void LifeStealBuff.SetActive (
            bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

Implements Ability.

3.56.1.5 SetModifier()

```
override void LifeStealBuff.SetModifier (
            bool state = false )  [virtual]
```

Callback for what this ability should do when a new modifier state is set

Parameters

state	The modifier state

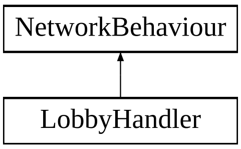
Reimplemented from Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BrawlerKit/LifeStealBuff.cs

3.57 LobbyHandler Class Reference

Inheritance diagram for LobbyHandler:



Public Member Functions

- int GetPlayerCount ()

    Gets the amount of connected players to the lobby
- void AddPlayer (DLNetworkLobbyPlayer player)

    Adds a player to the connectedPlayers list and then calls DecideEntryTeam(player)
- void SetPlayerTeam (DLNetworkLobbyPlayer player)

Adds the player to the correct team list and puts sets the parent of the player's visuals to the correct team panel

- void DisplayLobby ()

  Displays the lobby on the client and hides the "please wait while connecting" text
- void ResetLocalLobby ()

  Does the opposite of DisplayLobby()
- void RemovePlayer (DLNetworkLobbyPlayer player)

  Removes a disconnecting player from the correct team and destroys the visuals of that player.
- List< DLNetworkLobbyPlayer > GetConnectedPlayers ()

  Returns a list of connected players

Public Attributes

- RectTransform **redTeamPanel**
- RectTransform **blueTeamPanel**
- GameObject **waitingScreenObj**

3.57.1 Member Function Documentation

3.57.1.1 AddPlayer()

```
void LobbyHandler.AddPlayer (
            DLNetworkLobbyPlayer player )
```

Adds a player to the connectedPlayers list and then calls DecideEntryTeam(player)

Parameters

player	The player that we are adding

3.57.1.2 DisplayLobby()

```
void LobbyHandler.DisplayLobby ( )
```

Displays the lobby on the client and hides the "please wait while connecting" text

3.57.1.3 GetConnectedPlayers()

```
List<DLNetworkLobbyPlayer> LobbyHandler.GetConnectedPlayers ( )
```

Returns a list of connected players

Returns

A list of connected players

3.57.1.4 GetPlayerCount()

```
int LobbyHandler.GetPlayerCount ( )
```

Gets the amount of connected players to the lobby

Returns

The number of connected players

3.57.1.5 RemovePlayer()

```
void LobbyHandler.RemovePlayer (
            DLNetworkLobbyPlayer player )
```

Removes a disconnecting player from the correct team and destroys the visuals of that player.

Parameters

player	The player that just disconnected

3.57.1.6 ResetLocalLobby()

```
void LobbyHandler.ResetLocalLobby ( )
```

Does the opposite of DisplayLobby()

3.57.1.7 SetPlayerTeam()

```
void LobbyHandler.SetPlayerTeam (
            DLNetworkLobbyPlayer player )
```

Adds the player to the correct team list and puts sets the parent of the player's visuals to the correct team panel

Parameters

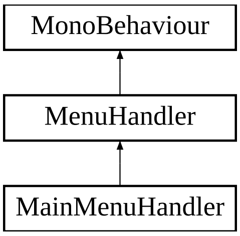
player	The player that we are adding

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/UI/LobbyHandler.cs

3.58 MainMenuHandler Class Reference

Inheritance diagram for MainMenuHandler:



Public Member Functions

- void NavigateTo (GameObject nextMenu)

  Navigates to a given menu gameObject and places the current one in the stack
- void NavigateBack ()

  Pops all menus from back stack until it hits a stopPop menu and navigates to that.
- void AddPropertyToStackTop (int enumId)

  Adds a property to the previous menu that is in the stack. This is mostly used as a workaround to the fact that the Unity Inspector's OnClick interface only supports none/single parameter functions
- void CreateOnlineMatch ()

  Uses the Unity match maker to create a new online match
- void StartMatchMaker ()

  Starts the Unity match maker

Public Attributes

- Text **hostRoomNameText**
- LobbyHandler **lobbyHandler**

Additional Inherited Members

3.58.1 Member Function Documentation

3.58.1.1 AddPropertyToStackTop()

```
void MainMenuHandler.AddPropertyToStackTop (
             int enumId )
```

Adds a property to the previous menu that is in the stack. This is mostly used as a workaround to the fact that the Unity Inspector's OnClick interface only supports none/single parameter functions

Parameters

enumId	The id of the property we are adding

3.58.1.2 CreateOnlineMatch()

```
void MainMenuHandler.CreateOnlineMatch ( )
```

Uses the Unity match maker to create a new online match

3.58.1.3 NavigateBack()

```
void MainMenuHandler.NavigateBack ( )
```

Pops all menus from back stack until it hits a stopPop menu and navigates to that.

3.58.1.4 NavigateTo()

```
void MainMenuHandler.NavigateTo (
            GameObject nextMenu )
```

Navigates to a given menu gameObject and places the current one in the stack

Parameters

nextMenu	The menu we are navigating to

3.58.1.5 StartMatchMaker()

```
void MainMenuHandler.StartMatchMaker ( )
```

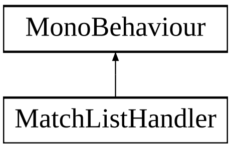
Starts the Unity match maker

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/UI/MainMenuHandler.cs

3.59 MatchListHandler Class Reference

Inheritance diagram for MatchListHandler:



Public Member Functions

- void OnMatchButtonClick (int buttonNumber, UnityEngine.Networking.Match.MatchInfoSnapshot match)

  A button listener callback that makes the client join the match that has been selected. Also navigates to the lobby menu screen.

Public Attributes

- GameObject **dynamicMatchButtonPrefab**
- int **matchButtonOffset** = 90
- MainMenuHandler **mainMenuHandler**
- GameObject **lobbyObj**
- GameObject **lobbyVerfifPromptObj**
- GameObject **noMatchesFoundObj**

3.59.1 Member Function Documentation

3.59.1.1 OnMatchButtonClick()

```
void MatchListHandler.OnMatchButtonClick (
            int buttonNumber,
            UnityEngine.Networking.Match.MatchInfoSnapshot match )
```

A button listener callback that makes the client join the match that has been selected. Also navigates to the lobby menu screen.

Parameters

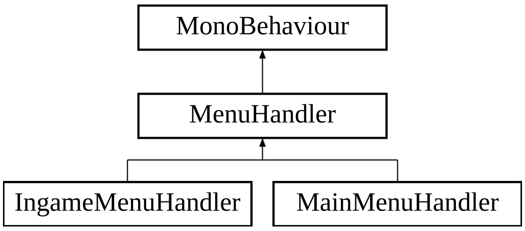
buttonNumber	The index of the button
match	The match maker match

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/UI/MatchListHandler.cs

3.60 MenuHandler Class Reference

Inheritance diagram for MenuHandler:



Public Member Functions

- void SetCurrentMenuVerificationPrompt (GameObject verifPrompt)

  Takes a verification prompt as parameter and connects it to the current menu.
- void OnClickSetFirstSelected ()

  Allows OnClick interfaces to use SetFirstSelectedGameObject(). Useful when a menu has submenus or verification prompts and you need to return control to the user after using these
- IEnumerator SetFirstSelectedGameObject (GameObject specific)

  Sets a button as selected the next frame after it has been called. If null is passed it sets the first selected button it finds. If a specific gameObject is passed it will look for buttons on that one instead

Public Attributes

- MenuStackComponent **currentActiveMenu**
- GameObject **menuRoot**

Protected Member Functions

- void **Start** ()

3.60.1 Member Function Documentation

3.60.1.1 OnClickSetFirstSelected()

```
void MenuHandler.OnClickSetFirstSelected ( )
```

Allows OnClick interfaces to use SetFirstSelectedGameObject(). Useful when a menu has submenus or verification prompts and you need to return control to the user after using these

3.60.1.2 SetCurrentMenuVerificationPrompt()

```
void MenuHandler.SetCurrentMenuVerificationPrompt (
            GameObject verifPrompt )
```

Takes a verification prompt as parameter and connects it to the current menu.

Parameters

verifPrompt	The gameObject of the verification prompt

3.60.1.3 SetFirstSelectedGameObject()

```
IEnumerator MenuHandler.SetFirstSelectedGameObject (
            GameObject specific )
```

Sets a button as selected the next frame after it has been called. If null is passed it sets the first selected button it finds. If a specific gameObject is passed it will look for buttons on that one instead

Parameters

specific	A gameObject containing buttons

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/UI/MenuHandler.cs

3.61 MenuStackComponent Class Reference

Public Member Functions

- **MenuStackComponent** (GameObject obj, menuStackProperty prop, bool hasVerPrompt, GameObject ver←PromptObj)

Public Attributes

- GameObject **menuObject**
- menuStackProperty **property**
- bool **currentMenuHasVerificationPrompt**
- GameObject **verificationPromptObj**

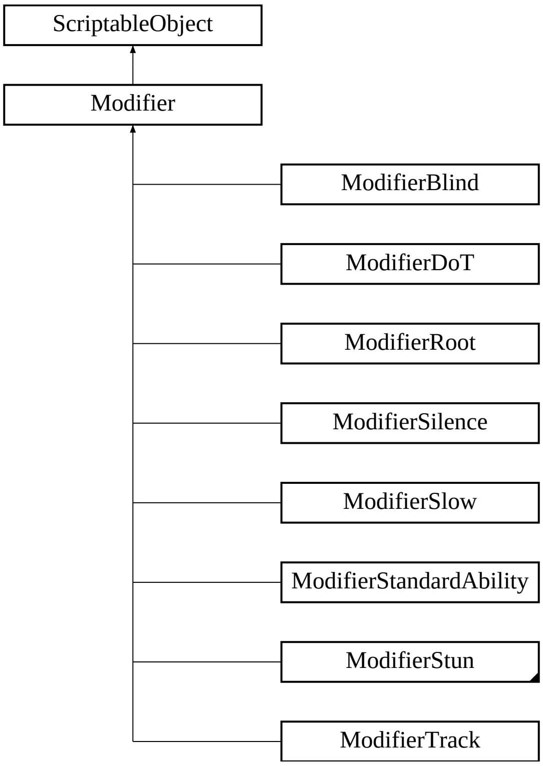
The documentation for this class was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/UI/MenuHandler.cs

3.62 Modifier Class Reference

Base class for every modifier.

Inheritance diagram for Modifier:



Public Member Functions

- virtual void OnServerStart (PlayerStatus playerStatus, int abilityId)

  Called on the server when the modifiers starts.
- virtual void OnClientStart (PlayerStatus playerStatus, int abilityId)

  Called on every client when the modifiers starts.
- virtual void OnLocalClientStart (PlayerStatus playerStatus)

  Called on the local client (the client the modifier is applied to) when the modifiers starts.
- virtual void OnServerEnd (PlayerStatus playerStatus, int abilityId)

  Called on the server when the modifiers ends.
- virtual void OnClientEnd (PlayerStatus playerStatus, int abilityId)

  Called on every client when the modifiers ends.
- virtual void OnLocalClientEnd (PlayerStatus playerStatus)

  Called on the local client (the client the modifier is applied to) when the modifiers ends.
- virtual void OnServerTick (PlayerStatus playerStatus)

  Called on the server whenever the modifier applies a tick.

Static Public Member Functions

- static Modifier GetModifierAsset (string modifierName)

  Looks up the Modifier with modifierName from the Resource/PlayerModifiers folder.

Public Attributes

- string **modifierName**
- Sprite **icon**
- GameObject **playerEffectObject**
- GameObject **localPlayerEffectObject**
- bool **unique**
- StatusType **statusType**

3.62.1 Detailed Description

Base class for every modifier.

3.62.2 Member Function Documentation

3.62.2.1 GetModifierAsset()

```
static Modifier Modifier.GetModifierAsset (
             string modifierName )  [static]
```

Looks up the Modifier with modifierName from the Resource/PlayerModifiers folder.

Parameters

modifierName	The modifier file name.

Returns

The modifier at path if found, otherwise null.

3.62.2.2 OnClientEnd()

```
virtual void Modifier.OnClientEnd (
             PlayerStatus playerStatus,
             int abilityId )  [virtual]
```

Called on every client when the modifiers ends.

Parameters

playerStatus	Reference to the associated PlayerStatus.
abilityId	The Id of the ability that applied the modifier if any, -1 otherwise.

Reimplemented in ModifierBlind, and ModifierStandardAbility.

3.62.2.3 OnClientStart()

```
virtual void Modifier.OnClientStart (
             PlayerStatus playerStatus,
             int abilityId )  [virtual]
```

Called on every client when the modifiers starts.

Parameters

playerStatus	Reference to the associated PlayerStatus.
abilityId	The Id of the ability that applied the modifier if any, -1 otherwise.

Reimplemented in ModifierBlind, and ModifierStandardAbility.

3.62.2.4 OnLocalClientEnd()

```
virtual void Modifier.OnLocalClientEnd (
             PlayerStatus playerStatus )  [virtual]
```

Called on the local client (the client the modifier is applied to) when the modifiers ends.

Parameters

playerStatus	Reference to the associated PlayerStatus.

Reimplemented in ModifierSlow, ModifierTrack, ModifierSilence, ModifierStun, and ModifierRoot.

3.62.2.5 OnLocalClientStart()

```
virtual void Modifier.OnLocalClientStart (
             PlayerStatus playerStatus )  [virtual]
```

Called on the local client (the client the modifier is applied to) when the modifiers starts.

Parameters

playerStatus	Reference to the associated PlayerStatus.

Reimplemented in ModifierFlashStun, ModifierSlow, ModifierTrack, ModifierRoot, ModifierSilence, and ModifierStun.

3.62.2.6 OnServerEnd()

```
virtual void Modifier.OnServerEnd (
             PlayerStatus playerStatus,
             int abilityId )  [virtual]
```

Called on the server when the modifiers ends.

Parameters

playerStatus	Reference to the associated PlayerStatus.
abilityId	The Id of the ability that applied the modifier if any, -1 otherwise.

3.62.2.7 OnServerStart()

```
virtual void Modifier.OnServerStart (
             PlayerStatus playerStatus,
             int abilityId )  [virtual]
```

Called on the server when the modifiers starts.

Parameters

playerStatus	Reference to the associated PlayerStatus.
abilityId	The Id of the ability that applied the modifier if any, -1 otherwise.

3.62.2.8 OnServerTick()

```
virtual void Modifier.OnServerTick (
             PlayerStatus playerStatus )  [virtual]
```

Called on the server whenever the modifier applies a tick.

Parameters

playerStatus	Reference to the associated PlayerStatus.

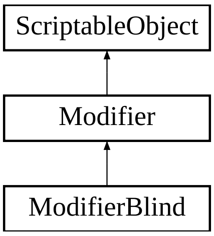
Reimplemented in ModifierDoT.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/Modifier.cs

3.63 ModifierBlind Class Reference

Inheritance diagram for ModifierBlind:



Public Member Functions

- override void OnClientStart (PlayerStatus playerStatus, int abilityId)

  Called on every client when the modifiers starts.
- override void OnClientEnd (PlayerStatus playerStatus, int abilityId)

  Called on every client when the modifiers ends.

Public Attributes

- float **blindLerpSpeed** = 10

Additional Inherited Members

3.63.1 Member Function Documentation

3.63.1.1 OnClientEnd()

```
override void ModifierBlind.OnClientEnd (
            PlayerStatus playerStatus,
            int abilityId )  [virtual]
```

Called on every client when the modifiers ends.

Parameters

playerStatus	Reference to the associated PlayerStatus.
abilityId	The Id of the ability that applied the modifier if any, -1 otherwise.

Reimplemented from Modifier.

3.63.1.2 OnClientStart()

```
override void ModifierBlind.OnClientStart (
             PlayerStatus playerStatus,
             int abilityId )  [virtual]
```

Called on every client when the modifiers starts.

Parameters

playerStatus	Reference to the associated PlayerStatus.
abilityId	The Id of the ability that applied the modifier if any, -1 otherwise.

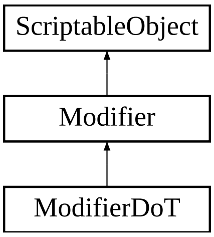
Reimplemented from Modifier.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/ModifierBlind.cs

3.64 ModifierDoT Class Reference

Inheritance diagram for ModifierDoT:



Public Member Functions

- override void OnServerTick (PlayerStatus playerStatus)

  Called on the server whenever the modifier applies a tick.

Public Attributes

- float **damagePerTick** = 5f

Additional Inherited Members

3.64.1 Member Function Documentation

3.64.1.1 OnServerTick()

```
override void ModifierDoT.OnServerTick (
            PlayerStatus playerStatus )  [virtual]
```

Called on the server whenever the modifier applies a tick.

Parameters

playerStatus	Reference to the associated PlayerStatus.

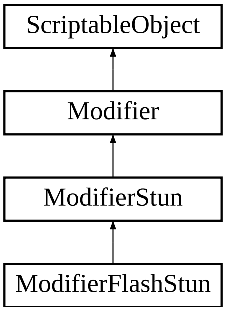
Reimplemented from Modifier.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/ModifierDoT.cs

3.65 ModifierFlashStun Class Reference

Inheritance diagram for ModifierFlashStun:



Public Member Functions

- override void OnLocalClientStart (PlayerStatus playerStatus)

  Called on the local client (the client the modifier is applied to) when the modifiers starts.

Public Attributes

- GameObject **flashPrefab**

Additional Inherited Members

3.65.1 Member Function Documentation

3.65.1.1 OnLocalClientStart()

```
override void ModifierFlashStun.OnLocalClientStart (
            PlayerStatus playerStatus )  [virtual]
```

Called on the local client (the client the modifier is applied to) when the modifiers starts.

Parameters

playerStatus	Reference to the associated PlayerStatus.

Reimplemented from Modifier.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/ModifierFlashStun.cs

3.66 ModifierInfo Struct Reference

Struct used in abilities to store modifier information.

Public Attributes

- Modifier **modifier**
- float **duration**
- int **tickCount**
- float **tickInterval**

3.66.1 Detailed Description

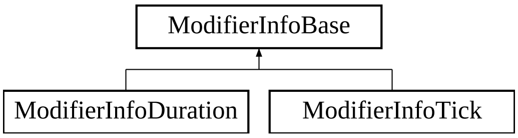
Struct used in abilities to store modifier information.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/Modifier.cs

3.67 ModifierInfoBase Class Reference

Inheritance diagram for ModifierInfoBase:



Public Attributes

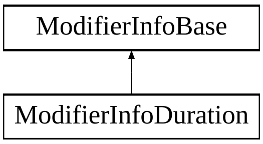
- Modifier **modifier**

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/Modifier.cs

3.68 ModifierInfoDuration Class Reference

Inheritance diagram for ModifierInfoDuration:



Public Attributes

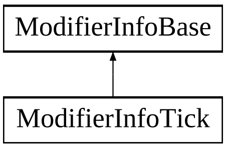
- float **duration**

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/Modifier.cs

3.69 ModifierInfoTick Class Reference

Inheritance diagram for ModifierInfoTick:



Public Attributes

- int **tickCount**
- float **tickInterval**

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/Modifier.cs

3.70 ModifierInstanceClient Class Reference

The instance used when a modifier is active. Only exists on the clients.

Public Member Functions

- ModifierInstanceClient (Modifier mod, PlayerStatus plStatus, PlayerUIHandler playerUIHandler, int modId, int abId, float duration)

  Constructor that instantiates effect objects and calls the correct modifier functions.
- void OnEnd ()

  Called when the modifier effect has ended.
- void SetNewDuration (float newDuration)

  Updates the UI elements with the new duration.
- Modifier GetModifier ()

  Returns the modifier used by this instance.
- int GetAbilityId ()

  Returns the ability ID in this instance, equal to or above 0 means this modifier was applied by an ability, -1 otherwise.
- int GetModifierId ()

  Returns the unique modifier ID for this instance.

3.70.1 Detailed Description

The instance used when a modifier is active. Only exists on the clients.

3.70.2 Constructor & Destructor Documentation

3.70.2.1 ModifierInstanceClient()

```
ModifierInstanceClient.ModifierInstanceClient (
             Modifier mod,
             PlayerStatus plStatus,
             PlayerUIHandler playerUIHandler,
             int modId,
             int abId,
             float duration )
```

Constructor that instantiates effect objects and calls the correct modifier functions.

Parameters

mod	The modifier for this instance.
plStatus	The playerStatus the modifier is applied to.
playerUIHandler	Reference to the UIHandler, used by the local client to add modifier UI.
modId	Unique identifier for this modifier instance.
abId	The Id of the ability that applied the modifier if any, -1 otherwise.
duration	The initial modifier duration.
effectParent	Either null or the transform we want to put as parent for this modifier

3.70.3 Member Function Documentation

3.70.3.1 GetAbilityId()

```
int ModifierInstanceClient.GetAbilityId ( )
```

Returns the ability ID in this instance, equal to or above 0 means this modifier was applied by an ability, -1 otherwise.

Returns

The ability ID.

3.70.3.2 GetModifier()

```
Modifier ModifierInstanceClient.GetModifier ( )
```

Returns the modifier used by this instance.

Returns

The active modifier.

3.70.3.3 GetModifierId()

```
int ModifierInstanceClient.GetModifierId ( )
```

Returns the unique modifier ID for this instance.

Returns

The modifier ID.

3.70.3.4 OnEnd()

```
void ModifierInstanceClient.OnEnd ( )
```

Called when the modifier effect has ended.

3.70.3.5 SetNewDuration()

```
void ModifierInstanceClient.SetNewDuration (
             float newDuration )
```

Updates the UI elements with the new duration.

Parameters

newDuration	The new duration.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/ModifierInstanceClient.cs

3.71 ModifierInstanceServer Class Reference

The instance used when a modifier is active. Only exists on the server.

Public Member Functions

- ModifierInstanceServer (ModifierInfo info, PlayerStatus plStatus, int modId, int abId)

  Constructor which starts the correct update loop as a Coroutine on the playerStatus MonoBehaviour.
- IEnumerator DurationLoop ()

  Update loop when the duration is used.
- IEnumerator TickLoop ()

  Update loop when the ticks are used.
- void OnEnd ()

  Called when the modifier effect has ended.
- void OnCancel ()

  Called when the ability modifier effect is cancelled (e.g. undocking).
- void MaxDuration (float newDuration)

  Used for unique modifiers that doesn't stack. Uses the largest of the given durations.
- Modifier GetModifier ()

  Returns the modifier used by this instance.
- int GetAbilityId ()

  Returns the ability ID in this instance, equal to or above 0 means this modifier was applied by an ability, -1 otherwise.
- int GetModifierId ()

  Returns the unique modifier ID for this instance.

3.71.1 Detailed Description

The instance used when a modifier is active. Only exists on the server.

3.71.2 Constructor & Destructor Documentation

3.71.2.1 ModifierInstanceServer()

```
ModifierInstanceServer.ModifierInstanceServer (
            ModifierInfo info,
            PlayerStatus plStatus,
            int modId,
            int abId )
```

Constructor which starts the correct update loop as a Coroutine on the playerStatus MonoBehaviour.

Parameters

info	Information about this modifier.
plStatus	The playerStatus the modifier is applied to.
modId	Unique identifier for this modifier instance.
abId	The Id of the ability that applied the modifier if any, -1 otherwise.

3.71.3 Member Function Documentation

3.71.3.1 DurationLoop()

```
IEnumerator ModifierInstanceServer.DurationLoop ( )
```

Update loop when the duration is used.

3.71.3.2 GetAbilityId()

```
int ModifierInstanceServer.GetAbilityId ( )
```

Returns the ability ID in this instance, equal to or above 0 means this modifier was applied by an ability, -1 otherwise.

Returns

The ability ID.

3.71.3.3 GetModifier()

```
Modifier ModifierInstanceServer.GetModifier ( )
```

Returns the modifier used by this instance.

Returns

The active modifier.

3.71.3.4 GetModifierId()

```
int ModifierInstanceServer.GetModifierId ( )
```

Returns the unique modifier ID for this instance.

Returns

The modifier ID.

3.71.3.5 MaxDuration()

```
void ModifierInstanceServer.MaxDuration (
             float newDuration )
```

Used for unique modifiers that doesn't stack. Uses the largest of the given durations.

Parameters

newDuration	The duration to compare the current duration with.

3.71.3.6 OnCancel()

```
void ModifierInstanceServer.OnCancel ( )
```

Called when the ability modifier effect is cancelled (e.g. undocking).

3.71.3.7 OnEnd()

```
void ModifierInstanceServer.OnEnd ( )
```

Called when the modifier effect has ended.

3.71.3.8 TickLoop()

```
IEnumerator ModifierInstanceServer.TickLoop ( )
```

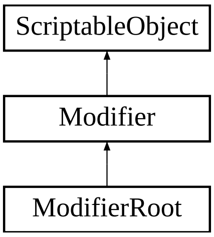
Update loop when the ticks are used.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/ModifierInstanceServer.cs

3.72 ModifierRoot Class Reference

Inheritance diagram for ModifierRoot:



Public Member Functions

- override void OnLocalClientStart (PlayerStatus playerStatus)

  Called on the local client (the client the modifier is applied to) when the modifiers starts.
- override void OnLocalClientEnd (PlayerStatus playerStatus)

  Called on the local client (the client the modifier is applied to) when the modifiers ends.

Additional Inherited Members

3.72.1 Member Function Documentation

3.72.1.1 OnLocalClientEnd()

```
override void ModifierRoot.OnLocalClientEnd (
              PlayerStatus playerStatus )  [virtual]
```

Called on the local client (the client the modifier is applied to) when the modifiers ends.

Parameters

playerStatus	Reference to the associated PlayerStatus.

Reimplemented from Modifier.

3.72.1.2 OnLocalClientStart()

```
override void ModifierRoot.OnLocalClientStart (
             PlayerStatus playerStatus )  [virtual]
```

Called on the local client (the client the modifier is applied to) when the modifiers starts.

Parameters

playerStatus	Reference to the associated PlayerStatus.

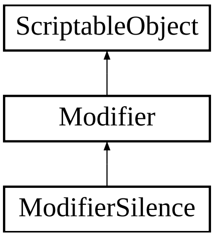
Reimplemented from Modifier.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/ModifierRoot.cs

3.73 ModifierSilence Class Reference

Inheritance diagram for ModifierSilence:



Public Member Functions

- override void OnLocalClientStart (PlayerStatus playerStatus)

  Called on the local client (the client the modifier is applied to) when the modifiers starts.
- override void OnLocalClientEnd (PlayerStatus playerStatus)

  Called on the local client (the client the modifier is applied to) when the modifiers ends.

Additional Inherited Members

3.73.1 Member Function Documentation

3.73.1.1 OnLocalClientEnd()

```
override void ModifierSilence.OnLocalClientEnd (
            PlayerStatus playerStatus )  [virtual]
```

Called on the local client (the client the modifier is applied to) when the modifiers ends.

Parameters

playerStatus	Reference to the associated PlayerStatus.

Reimplemented from Modifier.

3.73.1.2 OnLocalClientStart()

```
override void ModifierSilence.OnLocalClientStart (
            PlayerStatus playerStatus )  [virtual]
```

Called on the local client (the client the modifier is applied to) when the modifiers starts.

Parameters

playerStatus	Reference to the associated PlayerStatus.

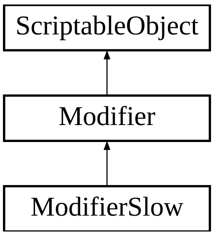
Reimplemented from Modifier.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/ModifierSilence.cs

3.74 ModifierSlow Class Reference

Inheritance diagram for ModifierSlow:



Public Member Functions

- override void OnLocalClientStart (PlayerStatus playerStatus)

  Called on the local client (the client the modifier is applied to) when the modifiers starts.
- override void OnLocalClientEnd (PlayerStatus playerStatus)

  Called on the local client (the client the modifier is applied to) when the modifiers ends.

Public Attributes

- float **slowPercentage** = 0.5f

Additional Inherited Members

3.74.1 Member Function Documentation

3.74.1.1 OnLocalClientEnd()

```
override void ModifierSlow.OnLocalClientEnd (
             PlayerStatus playerStatus )  [virtual]
```

Called on the local client (the client the modifier is applied to) when the modifiers ends.

Parameters

playerStatus	Reference to the associated PlayerStatus.

Reimplemented from Modifier.

3.74.1.2 OnLocalClientStart()

```
override void ModifierSlow.OnLocalClientStart (
             PlayerStatus playerStatus )  [virtual]
```

Called on the local client (the client the modifier is applied to) when the modifiers starts.

Parameters

playerStatus	Reference to the associated PlayerStatus.

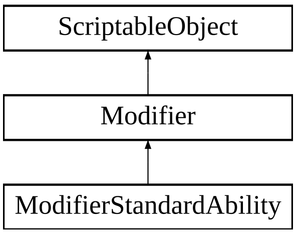
Reimplemented from Modifier.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/ModifierSlow.cs

3.75 ModifierStandardAbility Class Reference

Inheritance diagram for ModifierStandardAbility:



Public Member Functions

- override void OnClientStart (PlayerStatus playerStatus, int abilityId)

  Called on every client when the modifiers starts.
- override void OnClientEnd (PlayerStatus playerStatus, int abilityId)

  Called on every client when the modifiers ends.

Additional Inherited Members

3.75.1 Member Function Documentation

3.75.1.1 OnClientEnd()

```
override void ModifierStandardAbility.OnClientEnd (
             PlayerStatus playerStatus,
             int abilityId )  [virtual]
```

Called on every client when the modifiers ends.

Parameters

playerStatus	Reference to the associated PlayerStatus.
abilityId	The Id of the ability that applied the modifier if any, -1 otherwise.

Reimplemented from Modifier.

3.75.1.2 OnClientStart()

```
override void ModifierStandardAbility.OnClientStart (
             PlayerStatus playerStatus,
             int abilityId )  [virtual]
```

Called on every client when the modifiers starts.

Parameters

playerStatus	Reference to the associated PlayerStatus.
abilityId	The Id of the ability that applied the modifier if any, -1 otherwise.

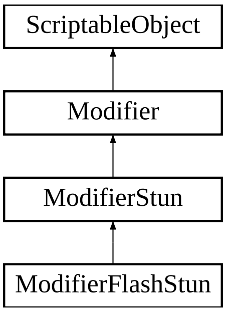
Reimplemented from Modifier.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/ModifierStandardAbility.cs

3.76 ModifierStun Class Reference

Inheritance diagram for ModifierStun:



Public Member Functions

- override void OnLocalClientStart (PlayerStatus playerStatus)

    Called on the local client (the client the modifier is applied to) when the modifiers starts.
- override void OnLocalClientEnd (PlayerStatus playerStatus)

    Called on the local client (the client the modifier is applied to) when the modifiers ends.

Additional Inherited Members

3.76.1 Member Function Documentation

3.76.1.1 OnLocalClientEnd()

```
override void ModifierStun.OnLocalClientEnd (
            PlayerStatus playerStatus )   [virtual]
```

Called on the local client (the client the modifier is applied to) when the modifiers ends.

Parameters

playerStatus	Reference to the associated PlayerStatus.

Reimplemented from Modifier.

3.76.1.2 OnLocalClientStart()

```
override void ModifierStun.OnLocalClientStart (
            PlayerStatus playerStatus )   [virtual]
```

Called on the local client (the client the modifier is applied to) when the modifiers starts.

Parameters

playerStatus	Reference to the associated PlayerStatus.

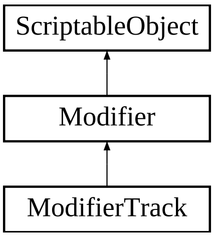
Reimplemented from Modifier.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/ModifierStun.cs

3.77 ModifierTrack Class Reference

Inheritance diagram for ModifierTrack:



Public Member Functions

- override void OnLocalClientStart (PlayerStatus playerStatus)

  Called on the local client (the client the modifier is applied to) when the modifiers starts.
- override void OnLocalClientEnd (PlayerStatus playerStatus)

  Called on the local client (the client the modifier is applied to) when the modifiers ends.

Public Attributes

- float **damageMultiplier**

Additional Inherited Members

3.77.1 Member Function Documentation

3.77.1.1 OnLocalClientEnd()

```
override void ModifierTrack.OnLocalClientEnd (
            PlayerStatus playerStatus )  [virtual]
```

Called on the local client (the client the modifier is applied to) when the modifiers ends.

Parameters

playerStatus	Reference to the associated PlayerStatus.

Reimplemented from Modifier.

3.77.1.2 OnLocalClientStart()

```
override void ModifierTrack.OnLocalClientStart (
             PlayerStatus playerStatus )  [virtual]
```

Called on the local client (the client the modifier is applied to) when the modifiers starts.

Parameters

playerStatus	Reference to the associated PlayerStatus.

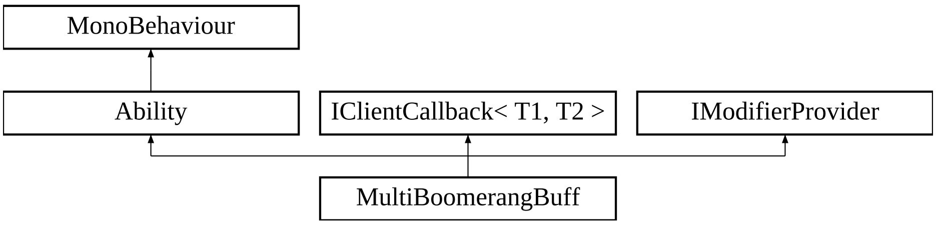
Reimplemented from Modifier.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Modifiers/ModifierTrack.cs

3.78 MultiBoomerangBuff Class Reference

Inheritance diagram for MultiBoomerangBuff:



Public Types

- enum **ClientCallback** { **BuffApplied** }

Public Member Functions

- override void ButtonDown ()

  Callback for what this ability does locally when its associated button is pressed
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.
- override void SetModifier (bool state=false)

  Callback for what this ability is supposed to do when a modifier state changes
- IEnumerator ResetBuff ()

  Coroutine used for resetting any visuals to default state. It waits for the end of the deactivation animation before doing anything.
- int **GetAbilityId** ()
- int **GetBuffModifierId** ()

Public Attributes

- BoomerangThrow **boomerangScript**
- ModifierInfo **buff**
- GameObject [ ] **otherBoomerangVisuals**
- Animator **boomerangAnimator**
- AnimationClip **boomerangAnimationClip**
- string **animationTrigger**
- bool **buffActive** = false
- bool **buffApplied** = false

Additional Inherited Members

3.78.1 Member Function Documentation

3.78.1.1 ButtonDown()

```
override void MultiBoomerangBuff.ButtonDown ( )  [virtual]
```

Callback for what this ability does locally when its associated button is pressed

Implements Ability.

3.78.1.2 ResetBuff()

```
IEnumerator MultiBoomerangBuff.ResetBuff ( )
```

Coroutine used for resetting any visuals to default state. It waits for the end of the deactivation animation before doing anything.

3.78.1.3 SetActive()

```
override void MultiBoomerangBuff.SetActive (
            bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

Implements Ability.

3.78.1.4 SetModifier()

```
override void MultiBoomerangBuff.SetModifier (
            bool state = false )  [virtual]
```

Callback for what this ability is supposed to do when a modifier state changes

Parameters

state	The new modifier state

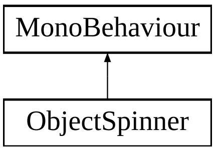
Reimplemented from Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BoomerangKit/MultiBoomerangBuff.cs

3.79 ObjectSpinner Class Reference

Inheritance diagram for ObjectSpinner:



Public Attributes

- float **rotationSpeed** = 10f
- Vector3 **axis** = new Vector3(0, 1, 0)

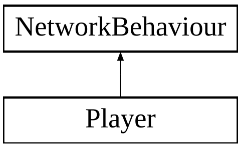
The documentation for this class was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Utilities/ObjectSpinner.cs

3.80 Player Class Reference

Handles the initialization for the local and remote events for each Player.

Inheritance diagram for Player:



Public Member Functions

- PlayerCamera **GetPlayerCamera** ()
- FieldOfView **GetPlayerFOV** ()
- GameObject **GetPlayerMask** ()
- int **GetPlayerTeamId** ()
- void TargetAddForce (NetworkConnection connection, float strength, ForceMode mode, Vector3 towards↩ Position)

  TargetRpc for adding force to the player rigidbody. Needed because the local player has authority, and needs to be the one adding force.
- void TargetAddForce2 (NetworkConnection connection, float strength, ForceMode mode, Vector3 forceOrigin)

  TargetRpc for adding force to the player rigidbody where the force origin relative to the player matters
- void TargetAddExplosionForce (NetworkConnection connection, float explosionForce, Vector3 explosion↩ Origin, float explosionRadius)

  TargetRpc for adding explosion force to the player rigidbody.
- void CmdInteract (GameObject interactableObject)

  Command called from PlayerInput when interacting with networked interactable objects.

Public Attributes

- ToggleEvent **onToggleShared**
- ToggleEvent **onToggleLocal**
- ToggleEvent **onToggleRemote**

3.80.1 Detailed Description

Handles the initialization for the local and remote events for each Player.

3.80.2 Member Function Documentation

3.80.2.1 CmdInteract()

```
void Player.CmdInteract (
             GameObject interactableObject )
```

Command called from PlayerInput when interacting with networked interactable objects.

Parameters

interactableObject	The networked gameobject interacted with.

3.80.2.2 TargetAddExplosionForce()

```
void Player.TargetAddExplosionForce (
             NetworkConnection connection,
             float explosionForce,
             Vector3 explosionOrigin,
             float explosionRadius )
```

TargetRpc for adding explosion force to the player rigidbody.

Parameters

connection	Needed so TargetRpc finds the correct client.
explosionForce	Amount of force in the explosion
explosionOrigin	Center of the explosion
explosionRadius	Radius of the explosion

3.80.2.3 TargetAddForce()

```
void Player.TargetAddForce (
            NetworkConnection connection,
            float strength,
            ForceMode mode,
            Vector3 towardsPosition )
```

TargetRpc for adding force to the player rigidbody. Needed because the local player has authority, and needs to be the one adding force.

Parameters

connection	Needed so TargetRpc finds the correct client.
strength	The force applied.
mode	The force mode used.

3.80.2.4 TargetAddForce2()

```
void Player.TargetAddForce2 (
            NetworkConnection connection,
            float strength,
            ForceMode mode,
            Vector3 forceOrigin )
```

TargetRpc for adding force to the player rigidbody where the force origin relative to the player matters

Parameters

connection	Needed so TargetRpc finds the correct client.
strength	Amount of force applied.
mode	The force mode used.
forceOrigin	Origin of the force.

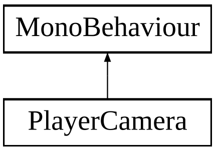
The documentation for this class was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Player/Player.cs

3.81 PlayerCamera Class Reference

Handles all Camera interactions.

Inheritance diagram for PlayerCamera:



Public Member Functions

- void SetPlayerTransform (Transform newPlayerTarget, bool returnToPlayer=false, bool smoothReturn=false)

  Sets the associated player transform.
- void SetTarget (Transform newTarget, bool smoothing=false)

  Set temporary target to follow, which will override the player transform. This will use the default move speed.
- void SetTarget (Transform t, bool smoothing, float speed)

  Set temporary target to follow using custom move speed, which will override the player transform.
- void SetOrthoSizeTarget (float targetSize)

  Set the orthographicSize for the cameras, will lerp between current and targetSize using the default speed.
- void SetOrthoSizeTarget (float targetSize, float speed)

  Set the orthographicSize for the cameras, will lerp between current and targetSize using the given speed.
- void ReturnToPlayer (bool smooth)

  Call for returning to the player transform using the default speed.
- void ReturnToPlayer (bool smooth, float speed)

  Call for returning to the player transform using the given speed.

Public Attributes

- float **height** = 25f
- float **defaultMoveSpeed** = 20f
- float **defaultScaleSpeed** = 2.5f

3.81.1 Detailed Description

Handles all Camera interactions.

3.81.2 Member Function Documentation

3.81.2.1 ReturnToPlayer() [1/2]

```
void PlayerCamera.ReturnToPlayer (
            bool smooth )
```

Call for returning to the player transform using the default speed.

Parameters

smooth	Whether the return is smooth or instant.

3.81.2.2 ReturnToPlayer() [2/2]

```
void PlayerCamera.ReturnToPlayer (
            bool smooth,
            float speed )
```

Call for returning to the player transform using the given speed.

Parameters

smooth	Whether the return is smooth or instant.
speed	The move speed utilized.

3.81.2.3 SetOrthoSizeTarget() [1/2]

```
void PlayerCamera.SetOrthoSizeTarget (
            float targetSize )
```

Set the orthographicSize for the cameras, will lerp between current and targetSize using the default speed.

Parameters

targetSize	The new orthographicSize.

3.81.2.4 SetOrthoSizeTarget() [2/2]

```
void PlayerCamera.SetOrthoSizeTarget (
            float targetSize,
            float speed )
```

Set the orthographicSize for the cameras, will lerp between current and targetSize using the given speed.

Parameters

targetSize	The new orthographicSize.
speed	The lerp speed utilized.

3.81.2.5 SetPlayerTransform()

```
void PlayerCamera.SetPlayerTransform (
            Transform newPlayerTarget,
            bool returnToPlayer = false,
            bool smoothReturn = false )
```

Sets the associated player transform.

Parameters

newPlayerTarget	The new player transform.
returnToPlayer	Whether to move the camera to this transform.
smoothReturn	Whether the return is smooth or instant.

3.81.2.6 SetTarget() [1/2]

```
void PlayerCamera.SetTarget (
            Transform newTarget,
            bool smoothing = false )
```

Set temporary target to follow, which will override the player transform. This will use the default move speed.

Parameters

newTarget	The new transform to follow.
smoothing	Whether to smoothly follow target.

3.81.2.7 SetTarget() [2/2]

```
void PlayerCamera.SetTarget (
            Transform t,
            bool smoothing,
            float speed )
```

Set temporary target to follow using custom move speed, which will override the player transform.

Parameters

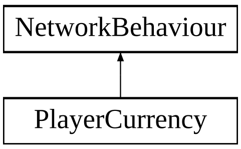
newTarget	The new transform to follow.
smoothing	Whether to smoothly follow target.
speed	The move speed utilized.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Player/PlayerCamera.cs

3.82 PlayerCurrency Class Reference

Inheritance diagram for PlayerCurrency:



Public Member Functions

- void CmdAddCurrency (int amount)

  Command for adding a new amount to the currency. This will automatically trigger the OnCurrencyChange hook

Public Attributes

- int **currency** = 0

3.82.1 Member Function Documentation

3.82.1.1 CmdAddCurrency()

```
void PlayerCurrency.CmdAddCurrency (
             int amount )
```

Command for adding a new amount to the currency. This will automatically trigger the OnCurrencyChange hook

Parameters

amount	The amount we add/decrease from the currency total

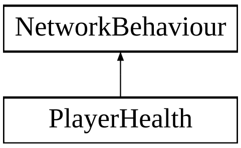
The documentation for this class was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Player/PlayerCurrency.cs

3.83 PlayerHealth Class Reference

Handles functionality related to the player health.

Inheritance diagram for PlayerHealth:



Public Member Functions

- override void OnStartClient ()

  Called when this object is activated on a client. Sets up the initial state and references.
- override void OnStartLocalPlayer ()

  Called when the local player object has been set up.
- void CmdSetMaxHealth (float newMaxHealth)

  Command called when a DockingKit changes the maxHealth.
- void CmdSetDamageMultiplier (float multiplier)

  Command called when the player receives damageMultiplier change, multiplicative.
- void TakeDamage (float damage, Vector3 knockbackForce)

  ServerCallback called when the player takes damage, and a knockback is required.
- void TakeDamage (float damage)

  ServerCallback called when the player takes damage.
- void Heal (float healing)

  ServerCallback called when the player receives health.

Public Attributes

- float **maxHealth** = 100f
- float **damageMultiplier** = 1f
- SpriteRenderer **damageHealthObject**
- float **flashSpeed** = 8f

3.83.1 Detailed Description

Handles functionality related to the player health.

3.83.2 Member Function Documentation

3.83.2.1 CmdSetDamageMultiplier()

```
void PlayerHealth.CmdSetDamageMultiplier (
            float multiplier )
```

Command called when the player receives damageMultiplier change, multiplicative.

Parameters

multiplier	change to multiplier

3.83.2.2 CmdSetMaxHealth()

```
void PlayerHealth.CmdSetMaxHealth (
            float newMaxHealth )
```

Command called when a DockingKit changes the maxHealth.

Parameters

newMaxHealth	

3.83.2.3 Heal()

```
void PlayerHealth.Heal (
             float healing )
```

ServerCallback called when the player receives health.

Parameters

healing	The amount of health received.

3.83.2.4 OnStartClient()

```
override void PlayerHealth.OnStartClient ( )
```

Called when this object is activated on a client. Sets up the initial state and references.

3.83.2.5 OnStartLocalPlayer()

```
override void PlayerHealth.OnStartLocalPlayer ( )
```

Called when the local player object has been set up.

3.83.2.6 TakeDamage() [1/2]

```
void PlayerHealth.TakeDamage (
             float damage,
             Vector3 knockbackForce )
```

ServerCallback called when the player takes damage, and a knockback is required.

Parameters

damage	The amount of damage taken.
knockbackForce	The force applied to the player rigidbody.

3.83.2.7 TakeDamage() [2/2]

```
void PlayerHealth.TakeDamage (
             float damage )
```

ServerCallback called when the player takes damage.

Parameters

damage	The amount of damage taken.

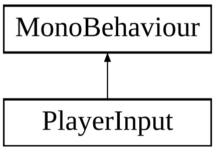
The documentation for this class was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Player/PlayerHealth.cs

3.84 PlayerInput Class Reference

Handles all player inputs.

Inheritance diagram for PlayerInput:



Public Member Functions

- Vector3 GetDirectionVector ()

  directionVector is set every frame based on the movement axis from player input.
- Vector2 GetRotationVector ()

  rotationVector is set every frame based on the rotation axis from player input.
- void SetInputRestrictions (bool state, InputType[ ] inputTypes)

  Used by the local player to self restrict input type. Using int stacks for situations where one modifier removes the restriction, but the restriction is still active by another.

Public Attributes

- string **moveHorizontal** = "Horizontal"
- string **moveVertical** = "Vertical"
- string **rotateHorizontal** = "HorizontalRotation"
- string **rotateVertical** = "VerticalRotation"
- string **dock** = "Dock"
- string **undock** = "Undock"
- string **interact** = "Interact"
- string [ ] **abilityButtons**
- float **moveSpeed**
- float **rotationSpeed**
- IngameMenuHandler **menuHandler**

3.84.1 Detailed Description

Handles all player inputs.

3.84.2 Member Function Documentation

3.84.2.1 GetDirectionVector()

```
Vector3 PlayerInput.GetDirectionVector ( )
```

directionVector is set every frame based on the movement axis from player input.

Returns

The direction vector.

3.84.2.2 GetRotationVector()

```
Vector2 PlayerInput.GetRotationVector ( )
```

rotationVector is set every frame based on the rotation axis from player input.

Returns

The rotation vector.

3.84.2.3 SetInputRestrictions()

```
void PlayerInput.SetInputRestrictions (
             bool state,
             InputType [] inputTypes )
```

Used by the local player to self restrict input type. Using int stacks for situations where one modifier removes the restriction, but the restriction is still active by another.

Parameters

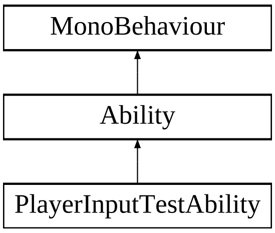
state	The new state of the input restriction.
types	The types to set restriction for.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Player/PlayerInput.cs

3.85 PlayerInputTestAbility Class Reference

Inheritance diagram for PlayerInputTestAbility:



Public Member Functions

- override void InitializeLocalPlayer (AbilityUI abilityUI)

  Initialization that only happens for the local player (Player controlling this ability). Called after Initialize, so the references are already set up.
- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void CancelAbility ()

  Call for cancelling abilities. Override in abilities that may be interrupted.
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Public Attributes

- Transform **target**
- float **moveSpeed** = 4f
- float **maxDistance** = 10f

Additional Inherited Members

3.85.1 Member Function Documentation

3.85.1.1 ButtonDown()

```
override void PlayerInputTestAbility.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.85.1.2 CancelAbility()

```
override void PlayerInputTestAbility.CancelAbility ( )  [virtual]
```

Call for cancelling abilities. Override in abilities that may be interrupted.

Reimplemented from Ability.

3.85.1.3 InitializeLocalPlayer()

```
override void PlayerInputTestAbility.InitializeLocalPlayer (
            AbilityUI abilityUI )  [virtual]
```

Initialization that only happens for the local player (Player controlling this ability). Called after Initialize, so the references are already set up.

Reimplemented from Ability.

3.85.1.4 SetActive()

```
override void PlayerInputTestAbility.SetActive (
            bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

Implements Ability.

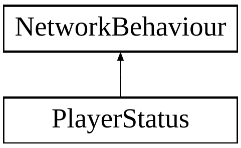
The documentation for this class was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TestAbilities/PlayerInputTestAbility.cs

3.86 PlayerStatus Class Reference

Handles the modifiers and status effects for the player.

Inheritance diagram for PlayerStatus:



Public Member Functions

- void ApplyModifier (ModifierInfo modifierInfo, int abilityId=-1)

  ServerCallback for applying a modifier. Searches through list of current modifiers if the modifier is unique.
- void RemoveModifier (ModifierInstanceServer instance, bool sync=true)

  Removes the modifier instance passed to it. Called by the ModifierInstance when the modifier has ended.
- void RemoveModifier (Modifier modifier)

  Removes the first instance equal to the modifier passed in. Used by abilities through the Docking.
- void RemoveAllModifiers ()

  Iterates through modifier list and stops everything.
- void RemoveAllAbilityModifiers ()

  Iterates through modifier list and stops ability (self applied) modifiers. (Modifiers with a valid abilityId).
- void TargetSetUIDuration (NetworkConnection connection, int modifierId, float newDuration)

  TargetRpc for updating UI elements duration.

Public Attributes

- ModifierInfo **stun**
- ModifierInfo **root**
- ModifierInfo **silence**
- ModifierInfo **dot**

3.86.1 Detailed Description

Handles the modifiers and status effects for the player.

3.86.2 Member Function Documentation

3.86.2.1 ApplyModifier()

```
void PlayerStatus.ApplyModifier (
            ModifierInfo modifierInfo,
            int abilityId = -1 )
```

ServerCallback for applying a modifier. Searches through list of current modifiers if the modifier is unique.

Parameters

modifierInfo	The information needed to apply the modifier.
abilityId	The Id of the ability that applied the modifier if any, -1 otherwise.

3.86.2.2 RemoveAllAbilityModifiers()

```
void PlayerStatus.RemoveAllAbilityModifiers ( )
```

Iterates through modifier list and stops ability (self applied) modifiers. (Modifiers with a valid abilityId).

3.86.2.3 RemoveAllModifiers()

```
void PlayerStatus.RemoveAllModifiers ( )
```

Iterates through modifier list and stops everything.

3.86.2.4 RemoveModifier() [1/2]

```
void PlayerStatus.RemoveModifier (
            ModifierInstanceServer instance,
            bool sync = true )
```

Removes the modifier instance passed to it. Called by the ModifierInstance when the modifier has ended.

Parameters

instance	The ModifierInstance that should be removed.
sync	Should this be synced to the clients.

3.86.2.5 RemoveModifier() [2/2]

```
void PlayerStatus.RemoveModifier (
            Modifier modifier )
```

Removes the first instance equal to the modifier passed in. Used by abilities through the Docking.

Parameters

modifier	The modifier to remove.

Works as long as abilities only self apply unique instances of modifiers, as this only removes based on modifier type (not unique id).

3.86.2.6 TargetSetUIDuration()

```
void PlayerStatus.TargetSetUIDuration (
            NetworkConnection connection,
            int modifierId,
            float newDuration )
```

TargetRpc for updating UI elements duration.

Parameters

connection	Needed so TargetRpc finds the correct client.
modifierId	Used to find correct modifier instance.
newDuration	The new duration.

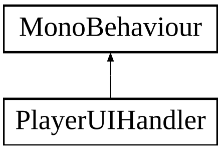
The documentation for this class was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Player/PlayerStatus.cs

3.87 PlayerUIHandler Class Reference

Handler for the player UI (Abilities, status modifiers, health).

Inheritance diagram for PlayerUIHandler:



Public Member Functions

- void SetDockingKitUI (DockingKit newDockingKit)

  Initialize the AbilityUI with the new DockingKit abilities.
- StatusUI AddStatusModifier (Modifier modifier, float duration)

  Adds a StatusUI element to the PlayerUI.
- void RemoveStatusModifier (StatusUI statusModifier)

  Removed the status modifier from the list of elements.
- void SetCurrentHealth (float health, float maxHealth)

  Updates the HealthUI based on health and maxHealth.
- void PlayCurrencyChangeAnimation (float currencyDifference)

  Starts a coroutine that interpolates text containing the amount of currency earned/spent

Public Attributes

- AbilityUI [ ] **abilities**
- Sprite **emptySlot**
- Transform [ ] **statusBars**
- GameObject **statusPrefab**
- Text **currencyText**
- Text **animatedCurrencyText**
- Text **healthPercentageText**
- Text **healthRatioText**
- Image **healthMask**
- Color **currencyAddColor**
- Color **currencyRemoveColor**
- float **animatedTextTargetOffset** = 75f
- IngameMenuHandler **ingameMenuHandler**

3.87.1 Detailed Description

Handler for the player UI (Abilities, status modifiers, health).

3.87.2 Member Function Documentation

3.87.2.1 AddStatusModifier()

```
StatusUI PlayerUIHandler.AddStatusModifier (
            Modifier modifier,
            float duration )
```

Adds a StatusUI element to the PlayerUI.

Parameters

modifier	The modifier to be added.
duration	The initial duration of the status modifier.

Returns

The instantiated statusUI element.

3.87.2.2 PlayCurrencyChangeAnimation()

```
void PlayerUIHandler.PlayCurrencyChangeAnimation (
            float currencyDifference )
```

Starts a coroutine that interpolates text containing the amount of currency earned/spent

Parameters

currencyDifference	The currency difference from the old total

3.87.2.3 RemoveStatusModifier()

```
void PlayerUIHandler.RemoveStatusModifier (
            StatusUI statusModifier )
```

Removed the status modifier from the list of elements.

Parameters

statusModifier	The statusUI removed.

3.87.2.4 SetCurrentHealth()

```
void PlayerUIHandler.SetCurrentHealth (
            float health,
            float maxHealth )
```

Updates the HealthUI based on health and maxHealth.

Parameters

health	The current health.
maxHealth	The current max health.

3.87.2.5 SetDockingKitUI()

```
void PlayerUIHandler.SetDockingKitUI (
            DockingKit newDockingKit )
```

Initialize the AbilityUI with the new DockingKit abilities.

Parameters

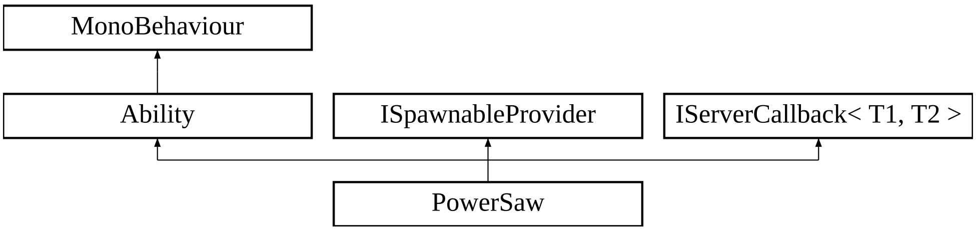
newDockingKit	Reference to the new DockingKit.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/UI/PlayerUIHandler.cs

3.88 PowerSaw Class Reference

Inheritance diagram for PowerSaw:



Public Member Functions

- override void Initialize (Docking dock, Animator anim, int abId)

  Initialization that happens locally on every client.
- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void CooldownReady ()

  Called from AbilityCooldown when the ability is ready. Setting active to false returns the sawblades to the docking kit visuals.
- override void SetActive (bool state)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Public Attributes

- float **triggerToSpawnTime**
- float **sawDamage**
- string **animatorBool**
- Collider **leftBlade**
- Collider **rightBlade**
- GameObject **bladePrefab**

Additional Inherited Members

3.88.1 Member Function Documentation

3.88.1.1 ButtonDown()

```
override void PowerSaw.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.88.1.2 CooldownReady()

```
override void PowerSaw.CooldownReady ( )  [virtual]
```

Called from AbilityCooldown when the ability is ready. Setting active to false returns the sawblades to the docking kit visuals.

Reimplemented from Ability.

3.88.1.3 Initialize()

```
override void PowerSaw.Initialize (
            Docking dock,
            Animator anim,
            int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented from Ability.

3.88.1.4 SetActive()

```
override void PowerSaw.SetActive (
            bool state )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

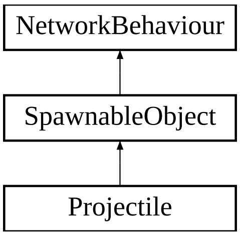
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TankKit/PowerSaw.cs

3.89 Projectile Class Reference

Inheritance diagram for Projectile:



Public Member Functions

- void **Initialize** (Stealth stealthRef, bool firedFromStealth=false)
- void **OnTriggerEnter** (Collider other)

Public Attributes

- float **projectileSpeed**
- float **lifetime**
- bool **hasStealthBonus**
- float **projectileDamage**
- Stealth **stealthBuff**

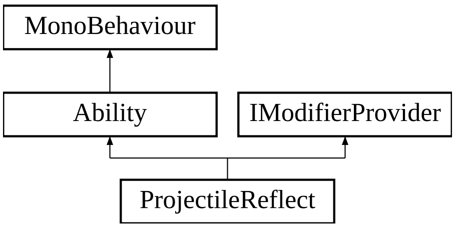
Additional Inherited Members

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/MarksmanKit/Projectile.cs

3.90 ProjectileReflect Class Reference

Inheritance diagram for ProjectileReflect:



Public Member Functions

- override void Initialize (Docking dock, Animator anim, int abId)

  Initialization that happens locally on every client.
- override void ButtonDown ()

  Callback for what this ability should do once its associated button has been pressed
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.
- override void SetModifier (bool state=false)

  Callback for what this ability is supposed to do depending on given state.

Public Attributes

- Transform **shieldTransform**
- float **fadeSpeed** = 5f
- float **fadeOutTimeOffset** = 0.5f
- ModifierInfo **buff**

Protected Member Functions

- override void Update ()

  Runs on every client, but only the local player has cooldown initialized.

Additional Inherited Members

3.90.1 Member Function Documentation

3.90.1.1 ButtonDown()

```
override void ProjectileReflect.ButtonDown ( )  [virtual]
```

Callback for what this ability should do once its associated button has been pressed

Implements Ability.

3.90.1.2 Initialize()

```
override void ProjectileReflect.Initialize (
             Docking dock,
             Animator anim,
             int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented from Ability.

3.90.1.3 SetActive()

```
override void ProjectileReflect.SetActive (
             bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

Implements Ability.

3.90.1.4 SetModifier()

```
override void ProjectileReflect.SetModifier (
            bool state = false )  [virtual]
```

Callback for what this ability is supposed to do depending on given state.

Parameters

state	Whether the ability is to be active or now

Reimplemented from Ability.

3.90.1.5 Update()

```
override void ProjectileReflect.Update ( )  [protected], [virtual]
```

Runs on every client, but only the local player has cooldown initialized.

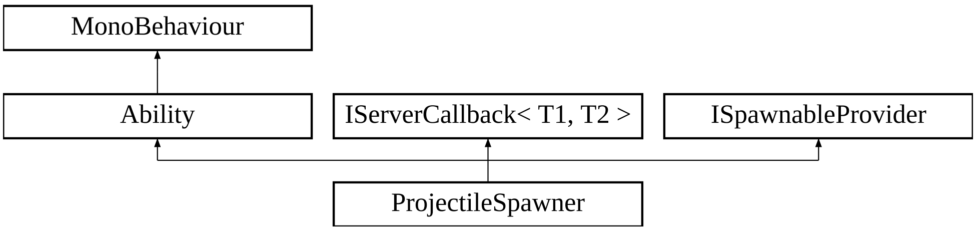
Reimplemented from Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BrawlerKit/ProjectileReflect.cs

3.91 ProjectileSpawner Class Reference

Inheritance diagram for ProjectileSpawner:



Public Member Functions

- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Public Attributes

- GameObject **projectilePrefab**
- float **spawnOffset**
- Stealth **stealthBuff**

Additional Inherited Members

3.91.1 Member Function Documentation

3.91.1.1 ButtonDown()

```
override void ProjectileSpawner.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.91.1.2 SetActive()

```
override void ProjectileSpawner.SetActive (
             bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

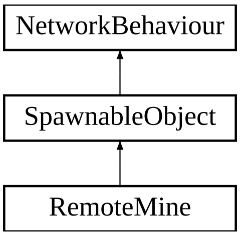
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/MarksmanKit/ProjectileSpawner.cs

3.92 RemoteMine Class Reference

Inheritance diagram for RemoteMine:



Public Member Functions

- void **Initialize** (GameObject owner)
- void **Explode** ()
- bool **IsActive** ()

Public Attributes

- float **baseDamage**
- float **explosionRadius**
- float **activationTime**
- ModifierInfo **stunInfo**

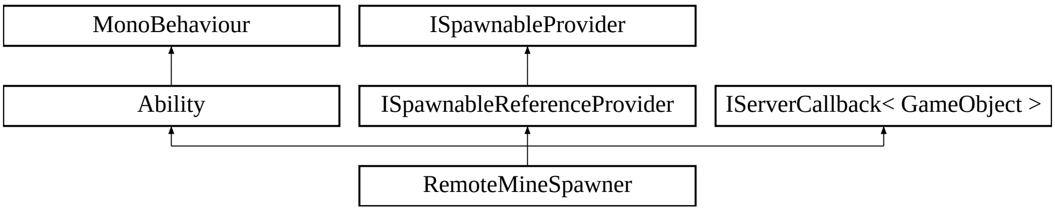
Additional Inherited Members

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BomberKit/RemoteMine.cs

3.93 RemoteMineSpawner Class Reference

Inheritance diagram for RemoteMineSpawner:



Public Member Functions

- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Public Attributes

- string **animatorTrigger**
- GameObject [ ] **minePrefab**
- GameObject **remoteMineReference**

Additional Inherited Members

3.93.1 Member Function Documentation

3.93.1.1 ButtonDown()

```
override void RemoteMineSpawner.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.93.1.2 SetActive()

```
override void RemoteMineSpawner.SetActive (
             bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

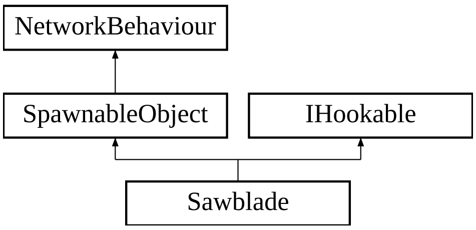
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BomberKit/RemoteMineSpawner.cs

3.94 Sawblade Class Reference

Inheritance diagram for Sawblade:



Public Member Functions

- void Hooked (GameObject playerObject, Transform hook)
  IHookable called when the sawblade has been hooked.

Public Attributes

- float **force** = 30f
- float **damage** = 20f
- float **lifetime** = 10f
- float **cooldownReduction** = 3f

Additional Inherited Members

3.94.1 Member Function Documentation

3.94.1.1 Hooked()

```
void Sawblade.Hooked (
            GameObject playerObject,
            Transform hook )
```

IHookable called when the sawblade has been hooked.

Parameters

playerObject	The hook's assosiated player object.
hook	The hook transform.

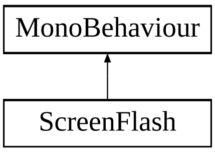
Implements IHookable.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TankKit/Sawblade.cs

3.95 ScreenFlash Class Reference

Inheritance diagram for ScreenFlash:



Public Attributes

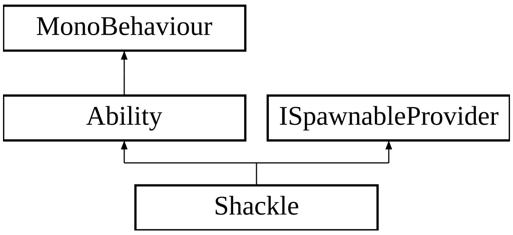
- float **fadeSpeed** = 5f

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/BrawlerKit/ScreenFlash.cs

3.96 Shackle Class Reference

Inheritance diagram for Shackle:



Public Member Functions

- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void CooldownReady ()

  Called from AbilityCooldown when the ability is ready. Setting active to true returns the bola to the docking kit visuals.
- override void SetActive (bool state)

  State is here the active of the bola visuals (Opposite of normal).

Public Attributes

- string **animatorTrigger**
- GameObject **spawnablePrefab**
- Transform **spawnPoint**

Additional Inherited Members

3.96.1 Member Function Documentation

3.96.1.1 ButtonDown()

```
override void Shackle.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.96.1.2 CooldownReady()

```
override void Shackle.CooldownReady ( )  [virtual]
```

Called from AbilityCooldown when the ability is ready. Setting active to true returns the bola to the docking kit visuals.

Reimplemented from Ability.

3.96.1.3 SetActive()

```
override void Shackle.SetActive (
             bool state )  [virtual]
```

State is here the active of the bola visuals (Opposite of normal).

Parameters

state	Visual state of the bola.

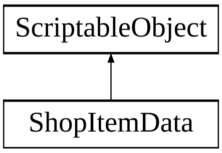
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/SniperKit/Shackle.cs

3.97 ShopItemData Class Reference

Inheritance diagram for ShopItemData:



Public Attributes

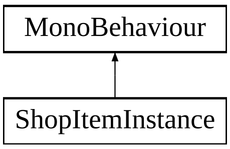
- string **itemName**
- Sprite **icon**
- GameObject **dockingKitPrefab**
- int **price**
- DockingKitId **dockingKitId**
- List< DockingKitDescriptions > **dockingKitDescriptions** = new List<DockingKitDescriptions>(5)

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/ShopItemData.cs

3.98 ShopItemInstance Class Reference

Inheritance diagram for ShopItemInstance:



Public Member Functions

- void **Initialize** (ShopItemData iData, IngameMenuHandler handler)
- void **OnSelectionChange** ()
- void **OnClick** ()

Public Attributes

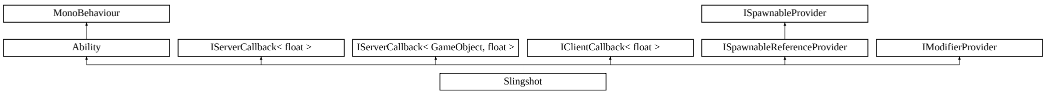
- ShopItemData **itemData**
- Image **uiIcon**
- Text **priceText**
- Image **unavailableOverlay**
- Text **isEquippedText**

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/ShopItemInstance.cs

3.99 Slingshot Class Reference

Inheritance diagram for Slingshot:



Public Member Functions

- override void InitializeLocalPlayer (AbilityUI abilityUI)

  Initialization that only happens for the local player (Player controlling this ability). Called after Initialize, so the references are already set up.
- override void ButtonDown ()

  Start the firing process if cooldown is ready.
- override void ButtonUp ()

  Fires the projectile if the ability is active.
- override void CancelAbility ()

  Cancel the firing process if active.
- override void SetActive (bool fire)

  Synchronizing states, either fires or resets.

Public Attributes

- GameObject **projectilePrefab**
- Transform **projectileSpawnPoint**
- Transform **leftFireIndicator**
- Transform **rightFireIndicator**
- ModifierInfo **snipingSlow**
- Transform **projectileVisuals**
- float **projectileMaxPrecisionY** = -1.5f
- LineRenderer **slingRenderer**
- float **startCurveModifier** = 0.5f
- float **holdCurveModifier** = 0.125f
- float **resetSpeed** = 2f
- AnimationCurve **startCurve**
- AnimationCurve **holdCurve**
- AnimationCurve **projectileFireAnimation**

Additional Inherited Members

3.99.1 Member Function Documentation

3.99.1.1 ButtonDown()

```
override void Slingshot.ButtonDown ( )  [virtual]
```

Start the firing process if cooldown is ready.

Implements Ability.

3.99.1.2 ButtonUp()

```
override void Slingshot.ButtonUp ( )  [virtual]
```

Fires the projectile if the ability is active.

Reimplemented from Ability.

3.99.1.3 CancelAbility()

```
override void Slingshot.CancelAbility ( )  [virtual]
```

Cancel the firing process if active.

Reimplemented from Ability.

3.99.1.4 InitializeLocalPlayer()

```
override void Slingshot.InitializeLocalPlayer (
             AbilityUI abilityUI )  [virtual]
```

Initialization that only happens for the local player (Player controlling this ability). Called after Initialize, so the references are already set up.

Reimplemented from Ability.

3.99.1.5 SetActive()

```
override void Slingshot.SetActive (
             bool fire )  [virtual]
```

Synchronizing states, either fires or resets.

Parameters

fire	If true fire, otherwise reset.

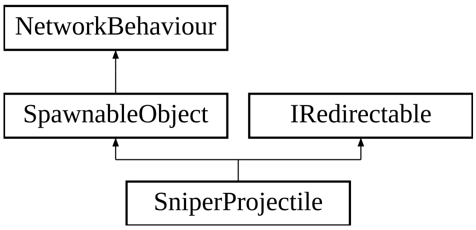
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/SniperKit/Slingshot.cs

3.100 SniperProjectile Class Reference

Inheritance diagram for SniperProjectile:



Public Member Functions

- void Initialize (float forceModifier)

  Server call for initializing the projectile based on the forceModifier.
- void RpcInitialize (float forceModifier)

  ClientRpc for synchronizing the forceModifier.

Public Attributes

- float **moveSpeed** = 60f
- float **damage** = 50f
- float **lifetime** = 8f

Additional Inherited Members

3.100.1 Member Function Documentation

3.100.1.1 Initialize()

```
void SniperProjectile.Initialize (
             float forceModifier )
```

Server call for initializing the projectile based on the forceModifier.

Parameters

forceModifier	Modifier in the 0-1 range which affects the stats.

3.100.1.2 RpcInitialize()

```
void SniperProjectile.RpcInitialize (
             float forceModifier )
```

ClientRpc for synchronizing the forceModifier.

Parameters

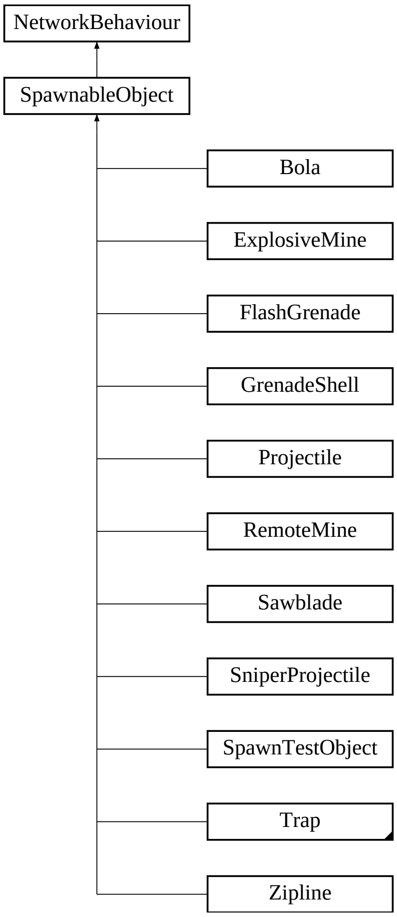
forceModifier	Modifier in the 0-1 range which affects the stats.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/SniperKit/SniperProjectile.cs

3.101 SpawnableObject Class Reference

Inheritance diagram for SpawnableObject:



Public Member Functions

- uint **GetOwnerPlayerId** ()
- int **GetOwnerTeamId** ()
- void **SetOwner** (uint player, int team)

Protected Attributes

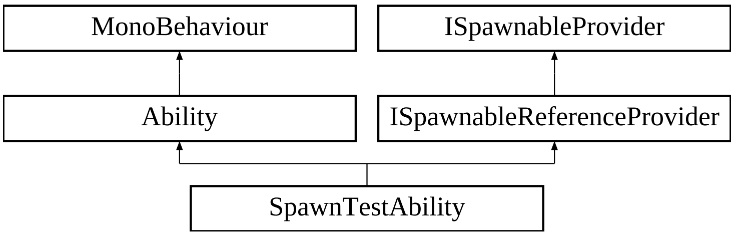
- uint **playerId**
- int **teamId**

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/SpawnableObject.cs

3.102 SpawnTestAbility Class Reference

Inheritance diagram for SpawnTestAbility:



Public Member Functions

- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Public Attributes

- GameObject **spawnTestPrefab**
- string **animatorTrigger**
- int **maxObjects** = 5
- List< GameObject > **spawnedObjects**

Additional Inherited Members

3.102.1 Member Function Documentation

3.102.1.1 ButtonDown()

```
override void SpawnTestAbility.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.102.1.2 SetActive()

```
override void SpawnTestAbility.SetActive (
             bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

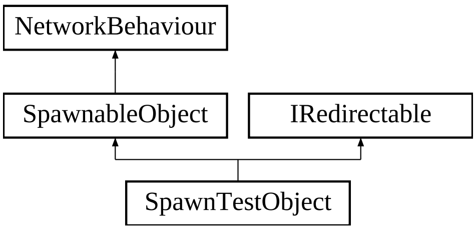
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TestAbilities/SpawnTestAbility.cs

3.103 SpawnTestObject Class Reference

Inheritance diagram for SpawnTestObject:



Public Member Functions

- void RedirectDirection (Vector3 newDirection, int newPlayerId=-1)

  Redirects direction of the spawnable.

Public Attributes

- float **moveSpeed**
- float **damage**

Additional Inherited Members

3.103.1 Member Function Documentation

3.103.1.1 RedirectDirection()

```
void SpawnTestObject.RedirectDirection (
             Vector3 newDirection,
             int newPlayerId = -1 )
```

Redirects direction of the spawnable.

Parameters

newDirection	The new direction.
newPlayerId	The player id of the new owner, -1 if current owner is kept.

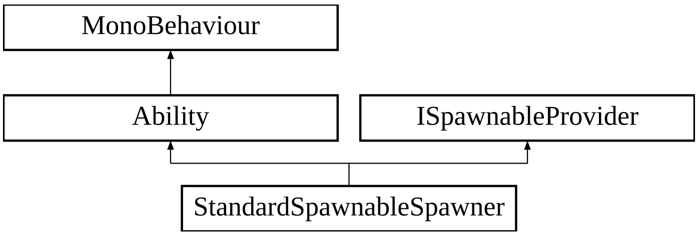
Implements IRedirectable.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TestAbilities/SpawnTestObject.cs

3.104 StandardSpawnableSpawner Class Reference

Inheritance diagram for StandardSpawnableSpawner:



Public Member Functions

- override void ButtonDown ()

  Callback for what this ability should do once its associated button has been pressed
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.
- GameObject GetSpawnablePrefab (int spawnableId)

  Used by the Docking to get the correct prefab to spawn from the abilities. Parameter only used if the ability has a list of prefabs.

Public Attributes

- string **animatorTrigger**
- GameObject **spawnablePrefab**
- Transform **spawnPoint**

Additional Inherited Members

3.104.1 Member Function Documentation

3.104.1.1 ButtonDown()

```
override void StandardSpawnableSpawner.ButtonDown ( )  [virtual]
```

Callback for what this ability should do once its associated button has been pressed

Implements Ability.

3.104.1.2 GetSpawnablePrefab()

```
GameObject StandardSpawnableSpawner.GetSpawnablePrefab (
             int spawnableId )
```

Used by the Docking to get the correct prefab to spawn from the abilities. Parameter only used if the ability has a list of prefabs.

Parameters

spawnable↵ Id	The Id of the spawnable object.

Returns

Reference to the prefab GameObject.

Implements ISpawnableProvider.

3.104.1.3 SetActive()

```
override void StandardSpawnableSpawner.SetActive (
             bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

Implements Ability.

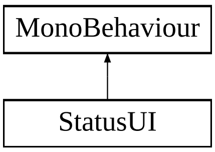
The documentation for this class was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/StandardSpawnableSpawner.cs

3.105 StatusUI Class Reference

Class for UI status modifiers.

Inheritance diagram for StatusUI:



Public Member Functions

- void Initialize (PlayerUIHandler playerUI, Sprite statusIcon, StatusType statusType, float startDuration)

  Initializes the UI element.
- void SetNewDuration (float newDuration)

  Sets the duration text of the UI element to the parameter.
- void Remove ()

  Remove and destroy this UI element.

Public Attributes

- Color **buffColor**
- Color **debuffColor**
- Image **frame**
- Image **darkMask**
- Text **durationText**
- Image **icon**

3.105.1 Detailed Description

Class for UI status modifiers.

3.105.2 Member Function Documentation

3.105.2.1 Initialize()

```
void StatusUI.Initialize (
            PlayerUIHandler playerUI,
            Sprite statusIcon,
            StatusType statusType,
            float startDuration )
```

Initializes the UI element.

Parameters

playerUI	Reference to the PlayerUIHandler.
statusIcon	The sprite that will be displayed in the UI element
statusType	Status type, buff or debuff.
startDuration	The start duration of the status effect.

3.105.2.2 Remove()

```
void StatusUI.Remove ( )
```

Remove and destroy this UI element.

3.105.2.3 SetNewDuration()

```
void StatusUI.SetNewDuration (
             float newDuration )
```

Sets the duration text of the UI element to the parameter.

Parameters

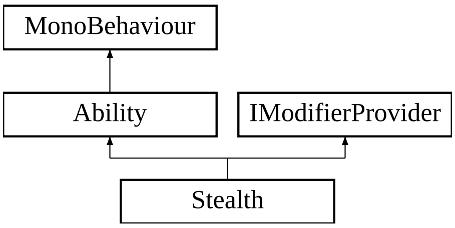
newDuration	The new duration we want to update with.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/UI/StatusUI.cs

3.106 Stealth Class Reference

Inheritance diagram for Stealth:



Public Member Functions

- override void Initialize (Docking dock, Animator anim, int abId)

  Initialization that happens locally on every client.
- override void ButtonDown ()

Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.
- override void SetModifier (bool state=false)

  Called by the Modifier. Appropriate place for doing local changes.
- void FindPlayerSpriteRenderers (List< string > names)

  Function to find the sprite rendererers relevant to fading into stealth
- bool **IsStealthed** ()
- int **GetAbilityId** ()
- int **GetBuffId** ()

Public Attributes

- float **stealthDamageBonus**
- List< string > **namesOfVisuals**
- List< SpriteRenderer > **visuals**
- ModifierInfo **buffInfo**
- ModifierInfo [ ] **modifierInfos**
- float **fadeTime**

Additional Inherited Members

3.106.1 Member Function Documentation

3.106.1.1 ButtonDown()

```
override void Stealth.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.106.1.2 FindPlayerSpriteRenderers()

```
void Stealth.FindPlayerSpriteRenderers (
            List< string > names )
```

Function to find the sprite rendererers relevant to fading into stealth

Parameters

name	The name of the parent

Returns

3.106.1.3 Initialize()

```
override void Stealth.Initialize (
             Docking dock,
             Animator anim,
             int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented from Ability.

3.106.1.4 SetActive()

```
override void Stealth.SetActive (
             bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

Implements Ability.

3.106.1.5 SetModifier()

```
override void Stealth.SetModifier (
             bool state = false )  [virtual]
```

Called by the Modifier. Appropriate place for doing local changes.

Parameters

state	If the modifier should be activated or deactivated.

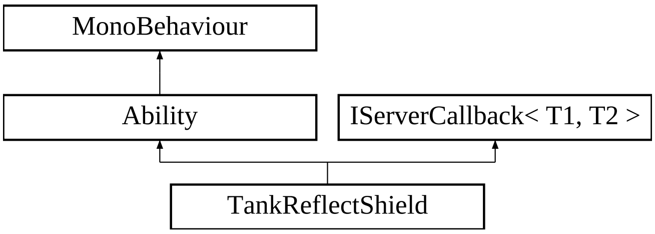
Reimplemented from Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/MarksmanKit/Stealth.cs

3.107 TankReflectShield Class Reference

Inheritance diagram for TankReflectShield:



Public Member Functions

- override void Initialize (Docking dock, Animator anim, int abId)

  Initialization that happens locally on every client.
- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Public Attributes

- float **duration**
- string **animatorBool**
- GameObject **shieldCollider**

Additional Inherited Members

3.107.1 Member Function Documentation

3.107.1.1 ButtonDown()

```
override void TankReflectShield.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.107.1.2 Initialize()

```
override void TankReflectShield.Initialize (
             Docking dock,
             Animator anim,
             int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented from Ability.

3.107.1.3 SetActive()

```
override void TankReflectShield.SetActive (
            bool state )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

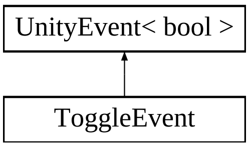
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TankKit/TankReflectShield.cs

3.108 ToggleEvent Class Reference

Inheritance diagram for ToggleEvent:

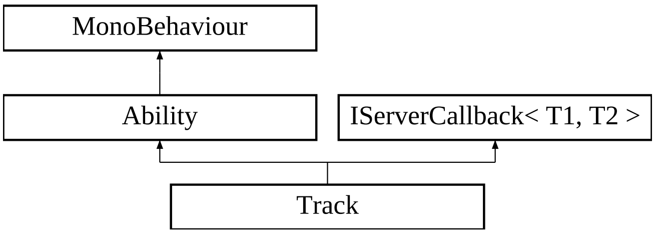


The documentation for this class was generated from the following file:

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Player/Player.cs

3.109 Track Class Reference

Inheritance diagram for Track:



Public Member Functions

- override void Initialize (Docking dock, Animator anim, int abId)

  Initialization that happens locally on every client.
- override void ButtonDown ()

  Called when the associated ability button is pressed. Must be overriden.
- override void SetActive (bool state=false)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Public Attributes

- float **castRange**
- LayerMask **layerMask**
- ModifierInfo **trackInfo**

Additional Inherited Members

3.109.1 Member Function Documentation

3.109.1.1 ButtonDown()

```
override void Track.ButtonDown ( )  [virtual]
```

Called when the associated ability button is pressed. Must be overriden.

Implements Ability.

3.109.1.2 Initialize()

```
override void Track.Initialize (
             Docking dock,
             Animator anim,
             int abId )  [virtual]
```

Initialization that happens locally on every client.

Parameters

dock	Reference to the associated Docking.
anim	Reference to the DockingKit animator.
abId	The ability's id in DockingKit abilities list.

Reimplemented from Ability.

3.109.1.3 SetActive()

```
override void Track.SetActive (
             bool state = false )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

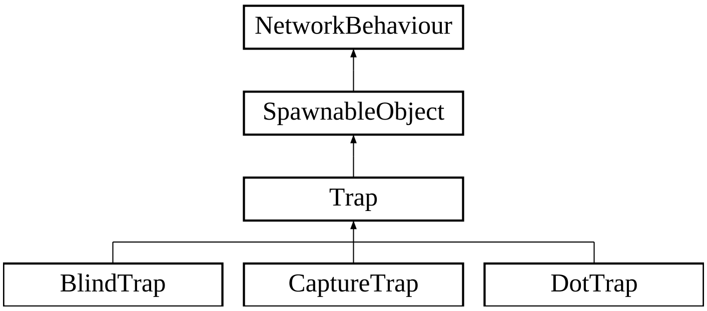
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/MarksmanKit/Track.cs

3.110 Trap Class Reference

Inheritance diagram for Trap:



Public Member Functions

- void Initialize (TrapSpawner owner)

  An initialisation function for caching the script reference to this trap's owner
- void SetVisualState (bool state)

  Sets the visual state of this trap
- virtual void HandleTrigger (PlayerStatus playerStatus)

  A virtual function that allows children of this class to handle what they want to do when a trap is triggered.

Public Attributes

- GameObject **visuals**
- GameObject **extraVisuals**
- float **timeAfterTriggerDestroy** = 1
- string **animatorTrigger**
- Animator **animator**

Protected Member Functions

- void RpcSetExtraVisualsState (bool state)

  ClientRpc used for synchronising the visual state of the trap

Protected Attributes

- List< Player > **appliedToList** = new List<Player>()
- List< Rigidbody > **appliedToListRbodies** = new List<Rigidbody>()

3.110.1 Member Function Documentation

3.110.1.1 HandleTrigger()

```
virtual void Trap.HandleTrigger (
             PlayerStatus playerStatus )  [virtual]
```

A virtual function that allows children of this class to handle what they want to do when a trap is triggered.

Parameters

playerStatus	The PlayerStatus component of the triggered player

Reimplemented in CaptureTrap, DotTrap, and BlindTrap.

3.110.1.2 Initialize()

```
void Trap.Initialize (
             TrapSpawner owner )
```

An initialisation function for caching the script reference to this trap's owner

Parameters

owner	

3.110.1.3 RpcSetExtraVisualsState()

```
void Trap.RpcSetExtraVisualsState (
             bool state )  [protected]
```

ClientRpc used for synchronising the visual state of the trap

Parameters

state	The state of the visuals

3.110.1.4 SetVisualState()

```
void Trap.SetVisualState (
             bool state )
```

Sets the visual state of this trap

Parameters

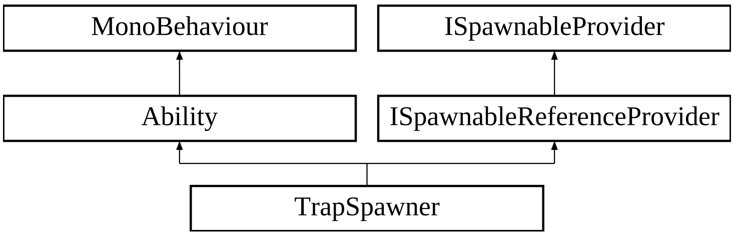
state	The visual state of this trap

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TrapperKit/Trap.cs

3.111 TrapSpawner Class Reference

Inheritance diagram for TrapSpawner:



Public Member Functions

- override void ButtonDown ()

  Callback for what the local client is supposed to do when this ability's button is pressed
- override void SetActive (bool state=false)

  Callback for synchronising visual state based on the given parameter
- void DisplayTrapState (bool state)

  A public function that allows traps to update the visual state of the docking kit's placed trap indicator

Public Attributes

- GameObject **trapPrefab**
- float **trapActiveAlpha** = 0.2f
- List< SpriteRenderer > **trapActiveSprites** = new List<SpriteRenderer>()
- float **lerpSpeed** = 5f

Additional Inherited Members

3.111.1 Member Function Documentation

3.111.1.1 ButtonDown()

```
override void TrapSpawner.ButtonDown ( )  [virtual]
```

Callback for what the local client is supposed to do when this ability's button is pressed

Implements Ability.

3.111.1.2 DisplayTrapState()

```
void TrapSpawner.DisplayTrapState (
             bool state )
```

A public function that allows traps to update the visual state of the docking kit's placed trap indicator

Parameters

state	The display state

3.111.1.3 SetActive()

```
override void TrapSpawner.SetActive (
             bool state = false )  [virtual]
```

Callback for synchronising visual state based on the given parameter

Parameters

state	The state of the ability.

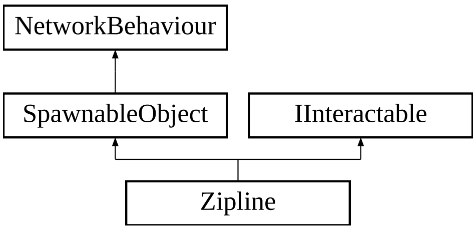
Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/TrapperKit/TrapSpawner.cs

3.112 Zipline Class Reference

Inheritance diagram for Zipline:



Public Member Functions

- bool FirePoint (GameObject player, Vector3 position, Vector3 direction)

  Server call from ZiplineGun whenever a point is fired.

Public Attributes

- Transform **wallEndPoint**
- Transform **lineStartPoint**
- Transform **lineEndPoint**
- SphereCollider **sphereCollider**
- LineRenderer **lineRenderer**
- Transform **handles**

- Transform **radiusTransform**
- LayerMask **interruptionLayerMask**
- float **maxFireRange** = 10f
- float **maxLineDistance** = 20f
- float **hookPointFireSpeed** = 40f
- float **normalRotationSpeed** = 10f
- float **playerMoveSpeed** = 20f
- int **uses** = 3

Additional Inherited Members

3.112.1 Member Function Documentation

3.112.1.1 FirePoint()

```
bool Zipline.FirePoint (
             GameObject player,
             Vector3 position,
             Vector3 direction )
```

Server call from ZiplineGun whenever a point is fired.

Parameters

player	The player firing.
position	Fired from this position.
direction	Fired in this direction.

Returns

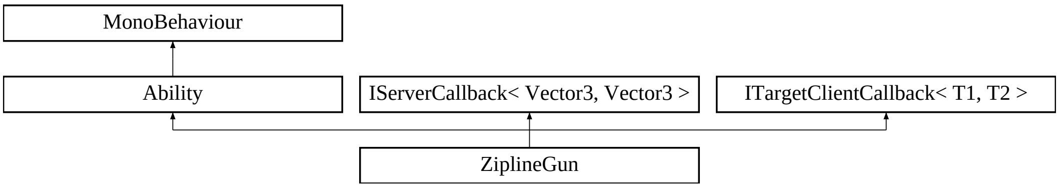
If the shot was successful.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/SniperKit/Zipline.cs

3.113 ZiplineGun Class Reference

Inheritance diagram for ZiplineGun:


MonoBehaviour
Ability
IServerCallback< Vector3, Vector3 >
ITargetClientCallback< T1, T2 >
ZiplineGun

Public Member Functions

- override void InitializeLocalPlayer (AbilityUI abilityUI)

  Initialization that only happens for the local player (Player controlling this ability). Called after Initialize, so the references are already set up.
- override void ButtonDown ()

  Activate the radius indicator if cooldown is ready.
- override void ButtonUp ()

  Fire the zipline if the radiusObject is active, this means ButtonDown was called when the cooldown was ready.
- override void SetActive (bool state)

  Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Public Attributes

- GameObject **ziplinePrefab**
- Transform **spawnPoint**
- GameObject **radiusObject**

Additional Inherited Members

3.113.1 Member Function Documentation

3.113.1.1 ButtonDown()

```
override void ZiplineGun.ButtonDown ( )  [virtual]
```

Activate the radius indicator if cooldown is ready.

Implements Ability.

3.113.1.2 ButtonUp()

```
override void ZiplineGun.ButtonUp ( )  [virtual]
```

Fire the zipline if the radiusObject is active, this means ButtonDown was called when the cooldown was ready.

Reimplemented from Ability.

3.113.1.3 InitializeLocalPlayer()

```
override void ZiplineGun.InitializeLocalPlayer (
             AbilityUI abilityUI )  [virtual]
```

Initialization that only happens for the local player (Player controlling this ability). Called after Initialize, so the references are already set up.

Reimplemented from Ability.

3.113.1.4 SetActive()

```
override void ZiplineGun.SetActive (
             bool state )  [virtual]
```

Synchronized state call between clients. Appropriate place for starting local animations and sounds.

Parameters

state	If the ability should be activated or deactivated.

Implements Ability.

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

- C:/Users/Andreas/Git Repos/dockitleague/Assets/Scripts/Abilities/SniperKit/ZiplineGun.cs

Index