

SuperPunch

3.50

Generated by Doxygen 1.8.10

Mon Apr 10 2017 10:38:04

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	BloodBehaviour Class Reference	5
3.1.1	Detailed Description	5
3.1.2	Member Data Documentation	5
3.1.2.1	timeOutRange	5
3.2	BloodMeter Class Reference	5
3.2.1	Detailed Description	6
3.2.2	Property Documentation	6
3.2.2.1	Measure	6
3.2.2.2	RelativeMeasure	6
3.3	BloodMeterModel Interface Reference	6
3.3.1	Detailed Description	7
3.4	GameBehaviour.endRound Class Reference	7
3.4.1	Detailed Description	7
3.5	GameBehaviour.fight Class Reference	8
3.5.1	Detailed Description	8
3.6	FighterController Class Reference	8
3.6.1	Detailed Description	11
3.6.2	Member Function Documentation	11
3.6.2.1	AnimSync()	11
3.6.2.2	ApplyDamage(HitData hitData)	11
3.6.2.3	GoToInitialPosition()	11
3.6.2.4	goToInitialPosition()	11
3.6.2.5	initializeVariables()	11
3.6.2.6	playBlockEffects()	11
3.6.2.7	playGroggyEffects()	11

3.6.2.8	playPunchEffects()	11
3.6.2.9	SetTransition(CombatTransitions t)	11
3.6.2.10	stopGroggyEffects()	12
3.6.2.11	updateFighterHealth()	12
3.6.2.12	updateFighterPower()	12
3.6.3	Property Documentation	12
3.6.3.1	activeInjured	12
3.6.3.2	Avatar	12
3.6.3.3	currentState	12
3.6.3.4	EffectivePower	12
3.6.3.5	Enable	12
3.6.3.6	EnableCountDown	12
3.6.3.7	Health	13
3.6.3.8	HealthBar	13
3.6.3.9	InjuredAvatar	13
3.6.3.10	InjuredDamage	13
3.6.3.11	Injuries	13
3.6.3.12	IsActive	13
3.6.3.13	isDead	13
3.6.3.14	IsHuman	13
3.6.3.15	isSuperPunchLoaded	13
3.6.3.16	KnockOutInfo	14
3.6.3.17	KnockOuts	14
3.6.3.18	MaxHealth	14
3.6.3.19	MaxSuperPunchHits	14
3.6.3.20	OnHitSucceed	14
3.6.3.21	Position	14
3.6.3.22	PowerBar	14
3.6.3.23	RelativeHealth	14
3.6.3.24	RelativePower	14
3.6.3.25	SucceedHits	15
3.6.3.26	SuperPunchHits	15
3.6.3.27	Tag	15
3.6.3.28	TotalSucceedHits	15
3.7	FighterInjuredBehaviour Class Reference	15
3.7.1	Detailed Description	16
3.7.2	Member Function Documentation	16
3.7.2.1	setInjuries(float damage)	16
3.7.3	Property Documentation	16
3.7.3.1	Avatar	16

3.7.3.2	Body	16
3.7.3.3	Damage	16
3.7.3.4	HitLimit	16
3.7.3.5	Injuries	16
3.8	FighterInjuryBehaviour Class Reference	17
3.8.1	Detailed Description	17
3.8.2	Member Function Documentation	17
3.8.2.1	OnEnable()	17
3.8.2.2	OnTouchBegan(Vector3 value)	17
3.8.3	Property Documentation	17
3.8.3.1	HitsLimit	17
3.9	FighterModel Interface Reference	18
3.9.1	Detailed Description	18
3.10	FingerEvent Class Reference	18
3.10.1	Detailed Description	19
3.10.2	Constructor & Destructor Documentation	19
3.10.2.1	FingerEvent(ITouchable it=null)	19
3.10.3	Member Function Documentation	19
3.10.3.1	Add(ITouchable obj)	19
3.10.3.2	clearList()	19
3.10.3.3	containsObject(ITouchable obj)	19
3.10.3.4	getLast()	20
3.10.3.5	remove(ITouchable obj)	20
3.11	FPSCounter Class Reference	20
3.11.1	Detailed Description	20
3.12	GameBehaviour Class Reference	20
3.12.1	Detailed Description	22
3.12.2	Member Function Documentation	22
3.12.2.1	playTheBell()	22
3.12.2.2	setCrowdBehaviour()	23
3.12.2.3	setGameOverStats()	23
3.12.2.4	SetTransition(GameTransitions t)	23
3.12.3	Property Documentation	23
3.12.3.1	AngrySpectators	23
3.12.3.2	Crowd	23
3.12.3.3	DynamicCrowd	23
3.12.3.4	ExcitedSpectators	23
3.12.3.5	GamePad	23
3.12.3.6	Lose	23
3.12.3.7	MAC	24

3.12.3.8	MaxRounds	24
3.12.3.9	PlayerONE	24
3.12.3.10	RelativeBloodMeasure	24
3.12.3.11	Round	24
3.12.3.12	RoundDuration	24
3.12.3.13	RoundGirl	24
3.12.3.14	RoundNumber	24
3.12.3.15	soundEffects	24
3.12.3.16	Versus	25
3.12.3.17	Win	25
3.13	GameBehaviour.GameOver Class Reference	25
3.13.1	Detailed Description	25
3.14	GamePadController Class Reference	25
3.14.1	Detailed Description	26
3.14.2	Member Function Documentation	26
3.14.2.1	OnEndButtonBehaivour()	26
3.14.2.2	OnStartButtonBehaivour()	26
3.14.2.3	OnTouchBegan(Vector3 value)	26
3.14.2.4	OnTouchEnded(Vector3 value)	26
3.14.2.5	Start()	26
3.15	headController Class Reference	27
3.15.1	Detailed Description	27
3.16	HitData Class Reference	27
3.17	GameBehaviour.InitCombat Class Reference	27
3.17.1	Detailed Description	28
3.18	GameBehaviour.limitRoundReached Class Reference	28
3.18.1	Detailed Description	28
3.19	MultiTouchController Class Reference	28
3.19.1	Detailed Description	29
3.20	ObjectBehaviour Class Reference	29
3.20.1	Detailed Description	29
3.21	ObjectPool Class Reference	29
3.21.1	Detailed Description	30
3.21.2	Member Function Documentation	30
3.21.2.1	activateObject()	30
3.22	GameBehaviour.PlayerLoses Class Reference	30
3.22.1	Detailed Description	30
3.23	GameBehaviour.PlayerWins Class Reference	31
3.23.1	Detailed Description	31
3.24	PlayHurtEffects Class Reference	31

3.24.1 Detailed Description	32
3.24.2 Member Function Documentation	32
3.24.2.1 ApplyDamage(HitData hitData)	32
3.25 PunchController Class Reference	32
3.25.1 Detailed Description	32
3.25.2 Property Documentation	32
3.25.2.1 defense	32
3.25.2.2 effectivePower	33
3.26 RoundGirlBehaviour Class Reference	33
3.26.1 Detailed Description	33
3.27 SelectBehaviour Class Reference	33
3.27.1 Member Function Documentation	34
3.27.1.1 action()	34
3.28 SpawnerBehaviour Class Reference	34
3.28.1 Detailed Description	34
3.29 GameBehaviour.startRound Class Reference	34
3.29.1 Detailed Description	35
3.30 testPunchBag Class Reference	35
Index	37

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

BloodMeterModel	6
BloodMeter	5
ButtonBehaviour	
SelectBehaviour	33
FighterModel	18
FighterController	8
FingerEvent	18
HitData	27
MonoBehaviour	
BloodBehaviour	5
BloodMeter	5
FighterController	8
FighterController	8
FighterInjuredBehaviour	15
FPSCounter	20
GameBehaviour	20
headController	27
MultiTouchController	28
ObjectBehaviour	29
ObjectPool	29
PlayHurtEffects	31
PunchController	32
RoundGirlBehaviour	33
SpawnerBehaviour	34
testPunchBag	35
State	
GameBehaviour.endRound	7
GameBehaviour.fight	8
GameBehaviour.GameOver	25
GameBehaviour.InitCombat	27
GameBehaviour.limitRoundReached	28
GameBehaviour.PlayerLoses	30
GameBehaviour.PlayerWins	31
GameBehaviour.startRound	34
TouchBehaviour	
FighterInjuryBehaviour	17
GamePadController	25

Chapter 2

Class Index

2.1 Class List

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

BloodBehaviour	Blood behaviour. Blood flesh is "instantiated" when fighter is punched on the face and this class sets the timer on when is enabled and when time is over disables the gameObject.	5
BloodMeter	Blood meter. GameObject used to detect the amount of blood spread over the ring.	5
BloodMeterModel	Blood meter model. Interface used to pass data through layers.	6
GameBehaviour.endRound	End round.	7
GameBehaviour.fight	Fight.	8
FighterController	Fighter controller.	8
FighterInjuredBehaviour	Fighter injured behaviour.	15
FighterInjuryBehaviour	Fighter injury behaviour.	17
FighterModel	Fighter model.	18
FingerEvent	Finger event.	18
FPSCounter	FPS counter.	20
GameBehaviour	Game behaviour.	20
GameBehaviour.GameOver	Game Over	25
GamePadController	Game pad controller.	25
headController	Head controller.	27
HitData	27
GameBehaviour.InitCombat	Init combat.	27
GameBehaviour.limitRoundReached	The limit Round is Reached	28
MultiTouchController	Multi touch controller.	28

ObjectBehaviour	
Object behaviour. Behaviour of the objects thrown by spectators.	29
ObjectPool	
Object pool.	29
GameBehaviour.PlayerLoses	
PPlayer Loses.	30
GameBehaviour.PlayerWins	
Player Wins.	31
PlayHurtEffects	
Play hurt effects.	31
PunchController	
Punch controller.	32
RoundGirlBehaviour	
Round girl behaviour.	33
SelectBehaviour	33
SpawnerBehaviour	
Spawner behaviour.	34
GameBehaviour.startRound	
Start round.	34
testPunchBag	35

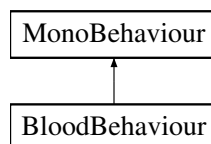
Chapter 3

Class Documentation

3.1 BloodBehaviour Class Reference

Blood behaviour. Blood flesh is "instantiated" when fighter is punched on the face and this class sets the timer on when is enabled and when time is over disables the gameObject.

Inheritance diagram for BloodBehaviour:



Public Attributes

- Vector2 `timeOutRange` =new Vector2(2,5)
The time out range.

3.1.1 Detailed Description

Blood behaviour. Blood flesh is "instantiated" when fighter is punched on the face and this class sets the timer on when is enabled and when time is over disables the gameObject.

3.1.2 Member Data Documentation

3.1.2.1 Vector2 BloodBehaviour.timeOutRange =new Vector2(2,5)

The time out range.

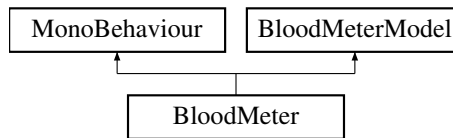
The documentation for this class was generated from the following file:

- behaviour/field/BloodBehaviour.cs

3.2 BloodMeter Class Reference

Blood meter. GameObject used to detect the amount of blood spread over the ring.

Inheritance diagram for BloodMeter:



Public Attributes

- float **maxBlood** = 25.0f
- float **bloodFactor** = 0.5f
- float **bloodDownFactor** = 0.25f
- float **totalBlood** = 0.0f

Properties

- float **MaxBlood** [get]
- float **Measure** [get, set]
Gets or sets the measure.
- float **RelativeMeasure** [get, set]
Gets the relative measure.

3.2.1 Detailed Description

Blood meter. GameObject used to detect the amount of blood spread over the ring.

3.2.2 Property Documentation

3.2.2.1 float BloodMeter.Measure [get], [set]

Gets or sets the measure.

The measure.

3.2.2.2 float BloodMeter.RelativeMeasure [get], [set]

Gets the relative measure.

The relative measure.

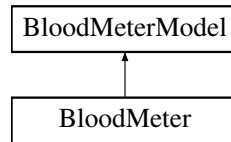
The documentation for this class was generated from the following file:

- behaviour/field/BloodMeter.cs

3.3 BloodMeterModel Interface Reference

Blood meter model. Interface used to pass data through layers.

Inheritance diagram for BloodMeterModel:



Properties

- float **MaxBlood** [get]
- float **Measure** [get, set]
- float **RelativeMeasure** [get, set]

3.3.1 Detailed Description

Blood meter model. Interface used to pass data through layers.

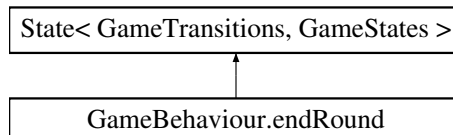
The documentation for this interface was generated from the following file:

- behaviour/field/BloodMeter.cs

3.4 GameBehaviour.endRound Class Reference

End round.

Inheritance diagram for GameBehaviour.endRound:



Public Member Functions

- **endRound** ([GameBehaviour](#) parent, float duration=0.1f)
- override void **Reason** ()
- override void **Act** ()
- override void **DoBeforeEntering** ()
- override void **DoBeforeLeaving** ()

3.4.1 Detailed Description

End round.

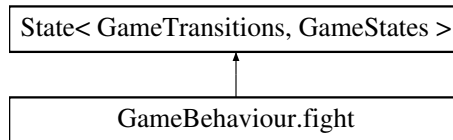
The documentation for this class was generated from the following file:

- behaviour/game/GameBehaviour.cs

3.5 GameBehaviour.fight Class Reference

Fight.

Inheritance diagram for GameBehaviour.fight:



Public Member Functions

- **fight** ([GameBehaviour](#) parent, float duration=0.1f)
- override void **Reason** ()
- override void **Act** ()
- override void **DoBeforeEntering** ()
- override void **DoBeforeLeaving** ()

3.5.1 Detailed Description

Fight.

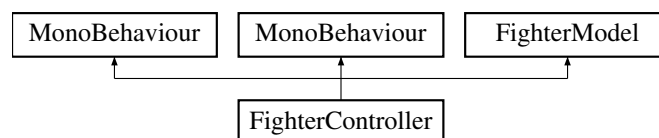
The documentation for this class was generated from the following file:

- `behaviour/game/GameBehaviour.cs`

3.6 FighterController Class Reference

Fighter controller.

Inheritance diagram for FighterController:



Public Member Functions

- bool [AnimSync](#) ()
Animation Sync. Plays the right animation based on player state.
- void [SetTransition](#) (CombatTransitions t)
Sets the transition.
- void [initializeVariables](#) ()
Initializes the variables.
- void [GoToInitialPosition](#) ()
Moves the fighter to the initial position.
- void [updateFighterHealth](#) ()
Updates the health bar status.

- void [updateFighterPower](#) ()
Updates the power bar status.
- void [ApplyDamage](#) (HitData hitData)
This function receives a class called hitData. It contains the hit damage and what's colliding.
- void [goToInitialPosition](#) ()
Moves the fighter to the initial position.
- void [playGroggyEffects](#) ()
Plaies the groggy effects.
- void [stopGroggyEffects](#) ()
Stops the groggy effects.
- void [playBlockEffects](#) ()
Plaies the block effects.
- void [playPunchEffects](#) ()
Plaies the punch effects.

Public Attributes

- Sprite[] **avatars**
- Sprite **largeAvatar**
- AudioClip[] **punchSFX**
- AudioClip[] **throwingPunchSFX**
- AudioClip[] **blockingPunchSFX**
- AudioClip[] **groggyEffectsSFX**
- int **knockOutsLimit** = 3
- int **hits4SuperPunch** = 5
- int **injuryHits** = 3
- int **punchRate** = 3
- int **superPunchHits** = 4
- float **health** = 0.0f
- float **punchPower** = 5.0f
- float **superPunchPower** = 10
- float **speed** = 35f
- float **precision** = 1.0f
- bool **superPunchFlag** = false
- bool **humanFlag** = false
- bool **superPunchAlwaysOn** = false

Protected Attributes

- FSMSys< CombatTransitions, CombatStates > **fsm**
- Vector3 **oldPosition** = Vector3.zero

Properties

- CombatStates [currentState](#) [get]
Gets the state of the current.
- float [MaxHealth](#) [get]
Gets the max health.
- int [MaxSuperPunchHits](#) [get]
Gets the max super punch hits.
- bool [OnHitSucceed](#) [get, set]

- Gets a value indicating whether this [FighterController](#) on hit succeed.*
- float [Health](#) [get]
Gets the health.
- float [RelativePower](#) [get]
Gets the relative power.
- float [RelativeHealth](#) [get]
Gets the relative health.
- int [Injuries](#) [get]
Gets the injuries.
- Sprite [Avatar](#) [get]
Gets the avatar.
- bool [IsHuman](#) [get, set]
Gets or sets a value indicating whether this instance is human.
- Vector3 [Position](#) [get, set]
Gets or sets the position.
- bool [isDead](#) [get, set]
Gets or sets a value indicating whether this [FighterController](#) is dead.
- bool [isSuperPunchLoaded](#) [get, set]
Gets a value indicating whether this [FighterController](#) is super punch loaded.
- bool [activeInjured](#) [set]
Sets a value indicating whether this [FighterController](#) active injured.
- float [InjuredDamage](#) [set]
Sets the injured damage.
- Sprite [InjuredAvatar](#) [set]
Sets the injured avatar.
- int [TotalSucceedHits](#) [get, set]
Gets or sets the total succeed hits.
- int [SucceedHits](#) [get, set]
Gets or sets the succeed hits.
- int [SuperPunchHits](#) [get, set]
Gets or sets the super punch hits.
- float [EffectivePower](#) [get, set]
Gets or sets the effective power.
- bool [EnableCountDown](#) [set]
Sets a value indicating whether this [FighterController](#) enable count down.
- bool [Enable](#) [get, set]
Gets or sets a value indicating whether this [FighterController](#) is enable.
- string [Tag](#) [get, set]
Gets or sets the tag.
- float [HealthBar](#) [set]
Sets the health bar.
- bool [KnockOutInfo](#) [set]
Sets a value indicating whether this [FighterController](#) knock out info.
- float [PowerBar](#) [set]
Sets the power bar.
- int [KnockOuts](#) [get, set]
Gets or sets the knock outs.
- bool [IsActive](#) [get, set]
Gets or sets a value indicating whether this instance is active.

3.6.1 Detailed Description

Fighter controller.

3.6.2 Member Function Documentation

3.6.2.1 bool FighterController.AnimSync ()

Animation Sync. Plays the right animation based on player state.

Returns

`true` `false` otherwise.

3.6.2.2 void FighterController.ApplyDamage (*HitData* *hitData*)

This function receives a class called *hitData*. It contains the hit damage and what's colliding.

Parameters

<i>hitData</i>	<i>hitData</i> .
----------------	------------------

3.6.2.3 void FighterController.GoToInitialPosition ()

Moves the fighter to the initial position.

Implements [FighterModel](#).

3.6.2.4 void FighterController.goToInitialPosition ()

Moves the fighter to the initial position.

3.6.2.5 void FighterController.initializeVariables ()

Initializes the variables.

3.6.2.6 void FighterController.playBlockEffects ()

Plays the block effects.

3.6.2.7 void FighterController.playGroggyEffects ()

Plays the groggy effects.

3.6.2.8 void FighterController.playPunchEffects ()

Plays the punch effects.

3.6.2.9 void FighterController.SetTransition (*CombatTransitions* *t*)

Sets the transition.

Parameters

t	T.
-----	----

Implements [FighterModel](#).

3.6.2.10 void FighterController.stopGroggyEffects ()

Stops the groggy effects.

3.6.2.11 void FighterController.updateFighterHealth ()

Updates the health bar status.

3.6.2.12 void FighterController.updateFighterPower ()

Updates the power bar status.

3.6.3 Property Documentation

3.6.3.1 bool FighterController.activeInjured [set]

Sets a value indicating whether this [FighterController](#) active injured.

true if active injured; otherwise, false.

3.6.3.2 Sprite FighterController.Avatar [get]

Gets the avatar.

The avatar.

3.6.3.3 CombatStates FighterController.currentState [get]

Gets the state of the current.

The state of the current.

3.6.3.4 float FighterController.EffectivePower [get], [set]

Gets or sets the effective power.

The effective power.

3.6.3.5 bool FighterController.Enable [get], [set]

Gets or sets a value indicating whether this [FighterController](#) is enable.

true if enable; otherwise, false.

3.6.3.6 bool FighterController.EnableCountDown [set]

Sets a value indicating whether this [FighterController](#) enable count down.

true if enable count down; otherwise, false.

3.6.3.7 float FighterController.Health [get]

Gets the health.

The health.

3.6.3.8 float FighterController.HealthBar [set]

Sets the health bar.

The health bar.

3.6.3.9 Sprite FighterController.InjuredAvatar [set]

Sets the injured avatar.

The injured avatar.

3.6.3.10 float FighterController.InjuredDamage [set]

Sets the injured damage.

The injured damage.

3.6.3.11 int FighterController.Injuries [get]

Gets the injuries.

The injuries.

3.6.3.12 bool FighterController.IsActive [get], [set]

Gets or sets a value indicating whether this instance is active.

true if this instance is active; otherwise, false.

3.6.3.13 bool FighterController.isDead [get], [set]

Gets or sets a value indicating whether this [FighterController](#) is dead.

true if is dead; otherwise, false.

3.6.3.14 bool FighterController.IsHuman [get], [set]

Gets or sets a value indicating whether this instance is human.

true if this instance is human; otherwise, false.

3.6.3.15 bool FighterController.isSuperPunchLoaded [get], [set]

Gets a value indicating whether this [FighterController](#) is super punch loaded.

true if is super punch loaded; otherwise, false.

3.6.3.16 `bool FighterController.KnockOutInfo` `[set]`

Sets a value indicating whether this [FighterController](#) knock out info.
`true` if knock out info; otherwise, `false`.

3.6.3.17 `int FighterController.KnockOuts` `[get], [set]`

Gets or sets the knock outs.
The knock outs.

3.6.3.18 `float FighterController.MaxHealth` `[get]`

Gets the max health.
The max health.

3.6.3.19 `int FighterController.MaxSuperPunchHits` `[get]`

Gets the max super punch hits.
The max super punch hits.

3.6.3.20 `bool FighterController.OnHitSucceed` `[get], [set]`

Gets a value indicating whether this [FighterController](#) on hit succeed.
`true` if on hit succeed; otherwise, `false`.

3.6.3.21 `Vector3 FighterController.Position` `[get], [set]`

Gets or sets the position.
The position.

3.6.3.22 `float FighterController.PowerBar` `[set]`

Sets the power bar.
The power bar.

3.6.3.23 `float FighterController.RelativeHealth` `[get]`

Gets the relative health.
The relative health.

3.6.3.24 `float FighterController.RelativePower` `[get]`

Gets the relative power.
The relative power.

3.6.3.25 `int FighterController.SucceedHits` `[get]`, `[set]`

Gets or sets the succeed hits.

The succeed hits.

3.6.3.26 `int FighterController.SuperPunchHits` `[get]`, `[set]`

Gets or sets the super punch hits.

The super punch hits.

3.6.3.27 `string FighterController.Tag` `[get]`, `[set]`

Gets or sets the tag.

The tag.

3.6.3.28 `int FighterController.TotalSucceedHits` `[get]`, `[set]`

Gets or sets the total succeed hits.

The total succeed hits.

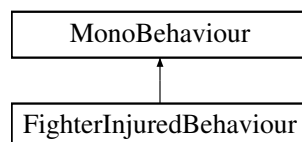
The documentation for this class was generated from the following file:

- `behaviour/fighter/FighterController.cs`

3.7 FighterInjuredBehaviour Class Reference

Fighter injured behaviour.

Inheritance diagram for FighterInjuredBehaviour:



Public Member Functions

- void `setInjuries` (float damage)
Enable all the injuries.

Properties

- Transform `Body` `[get]`
the fighter body.
- float `Damage` `[get, set]`
Gets or sets the damage amount.
- int `Injuries` `[get]`
Gets the injuries amount.
- int `HitLimit` `[get, set]`

Sets the hit limit.

- Sprite [Avatar](#) [set]

Sets the avatar.

3.7.1 Detailed Description

Fighter injured behaviour.

3.7.2 Member Function Documentation

3.7.2.1 void FighterInjuredBehaviour.setInjuries (float *damage*)

Enable all the injuries.

Parameters

<i>damage</i>	Damage.
---------------	---------

3.7.3 Property Documentation

3.7.3.1 Sprite FighterInjuredBehaviour.Avatar [set]

Sets the avatar.

The avatar.

3.7.3.2 Transform FighterInjuredBehaviour.Body [get]

the fighter body.

The body.

3.7.3.3 float FighterInjuredBehaviour.Damage [get], [set]

Gets or sets the damage amount.

The damage amount.

3.7.3.4 int FighterInjuredBehaviour.HitLimit [get], [set]

Sets the hit limit.

The hit limit.

3.7.3.5 int FighterInjuredBehaviour.Injuries [get]

Gets the injuries amount.

The injuries.

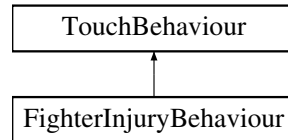
The documentation for this class was generated from the following file:

- behaviour/fighter/injured/FighterInjuredBehaviour.cs

3.8 FighterInjuryBehaviour Class Reference

Fighter injury behaviour.

Inheritance diagram for FighterInjuryBehaviour:



Public Member Functions

- override void [OnEnable](#) ()
Use this for initialization
- override void [OnTouchBegan](#) (Vector3 value)
Raises the touch down event.

Public Attributes

- AudioClip **onTouchSFX**
- AudioClip **onCuredSFX**

Properties

- int [HitsLimit](#) [get, set]
Gets or sets the hits limit.

3.8.1 Detailed Description

Fighter injury behaviour.

3.8.2 Member Function Documentation

3.8.2.1 override void FighterInjuryBehaviour.OnEnable ()

Use this for initialization

3.8.2.2 override void FighterInjuryBehaviour.OnTouchBegan (Vector3 value)

Raises the touch down event.

3.8.3 Property Documentation

3.8.3.1 int FighterInjuryBehaviour.HitsLimit [get], [set]

Gets or sets the hits limit.

The hits limit.

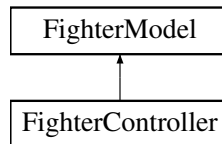
The documentation for this class was generated from the following file:

- behaviour/fighter/injured/FighterInjuryBehaviour.cs

3.9 FighterModel Interface Reference

Fighter model.

Inheritance diagram for FighterModel:



Public Member Functions

- void **SetTransition** (CombatTransitions t)
- void **GoToInitialPosition** ()

Properties

- float **Health** [get]
- bool **IsHuman** [get, set]
- bool **IsActive** [get, set]
- Vector3 **Position** [get, set]
- int **TotalSucceedHits** [get, set]
- int **KnockOuts** [get]
- bool **isDead** [get, set]
- CombatStates **currentState** [get]
- bool **Enable** [get, set]
- bool **OnHitSucceed** [get, set]
- string **Tag** [get, set]

3.9.1 Detailed Description

Fighter model.

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

- behaviour/fighter/FighterController.cs

3.10 FingerEvent Class Reference

Finger event.

Public Member Functions

- [FingerEvent](#) (ITouchable it=null)
Initializes a new instance of the [FingerEvent](#) class.
- bool [Add](#) (ITouchable obj)

- Add the specified obj.*
- `ITouchable` [getLast](#) ()
Gets the last GameObject added.
- `void` [clearList](#) ()
Clears the list.
- `bool` [containsObject](#) (`ITouchable obj`)
- `ITouchable` [remove](#) (`ITouchable obj`)
Remove the specified obj.

Public Attributes

- `List< ITouchable >` **touchedObjects** =new List<ITouchable>()

3.10.1 Detailed Description

Finger event.

3.10.2 Constructor & Destructor Documentation

3.10.2.1 `FingerEvent.FingerEvent (ITouchable it = null)`

Initializes a new instance of the [FingerEvent](#) class.

Parameters

<i>it</i>	It.
-----------	-----

3.10.3 Member Function Documentation

3.10.3.1 `bool FingerEvent.Add (ITouchable obj)`

Add the specified obj.

Parameters

<i>obj</i>	Object.
------------	---------

3.10.3.2 `void FingerEvent.clearList ()`

Clears the list.

3.10.3.3 `bool FingerEvent.containsObject (ITouchable obj)`

Checks if touched gameObject(obj) was registered previously touched.

Returns

`true`, if object was contained, `false` otherwise.

Parameters

<i>obj</i>	Object.
------------	---------

3.10.3.4 `ITouchable FingerEvent.getLast ()`

Gets the last GameObject added.

Returns

The last.

3.10.3.5 `ITouchable FingerEvent.remove (ITouchable obj)`

Remove the specified obj.

Parameters

<i>obj</i>	Object.
------------	---------

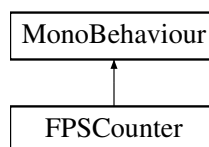
The documentation for this class was generated from the following file:

- behaviour/game/MultiTouchController.cs

3.11 FPSCounter Class Reference

FPS counter.

Inheritance diagram for FPSCounter:

**3.11.1 Detailed Description**

FPS counter.

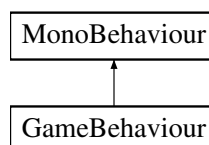
The documentation for this class was generated from the following file:

- test/FPSCounter.cs

3.12 GameBehaviour Class Reference

Game behaviour.

Inheritance diagram for GameBehaviour:



Classes

- class [endRound](#)
End round.
- class [fight](#)
Fight.
- class [GameOver](#)
Game Over
- class [InitCombat](#)
Init combat.
- class [limitRoundReached](#)
The limit Round is Reached
- class [PlayerLoses](#)
PLayer Loses.
- class [PlayerWins](#)
Player Wins.
- class [startRound](#)
Start round.

Public Member Functions

- void **showInterstitialAds** ()
- void [playTheBell](#) ()
Plaies the bell.
- void [setCrowdBehaviour](#) ()
Sets the crowd behaviour.
- void [setGameOverStats](#) ()
Sets the game over stats.
- void **loadCombatMusic** ()
- void [SetTransition](#) (GameTransitions t)
Sets the transition.

Public Attributes

- AudioClip[] **music**
- GameObject[] **fighters**
- bool **dynamicCrowd** = false
- Vector2 **angryCrowdRange** = new Vector2(0f,0.25f)
- Vector2 **excitedCrowdRange** = new Vector2(0.7f,1f)
- float **initialBloodMeasure** = 0.0f
- Vector3 **playerTwoPosition** = new Vector3(0,7.47f,0)
- Vector3 **playerOnePosition** = new Vector3(0,-5.49f,0)
- int **maxRounds** = 10
- float **roundDuration** = 20.0f
- GameObject **roundGirl**
- GameObject **spectatorsExcited**
- GameObject **spectatorsAngry**
- GameObject **bell**
- GameObject **versus**
- GameObject **win**
- GameObject **lose**
- GameObject **roundStats**

- GameObject **knockOutsoStats**
- GameObject **succeedHitsStats**
- GameObject **gamePad**
- GameObject **gameOver**

Properties

- int **Round** [get, set]
Gets or sets the round.
- bool **Versus** [set]
Sets a value indicating whether this [GameBehaviour](#) is versus.
- int **MaxRounds** [get]
Gets the max rounds.
- float **RoundDuration** [get]
Gets the duration of the round.
- **FighterModel PlayerONE** [get, set]
Gets or sets the player ON.
- **FighterModel MAC** [get, set]
Gets or sets the MA.
- bool **RoundGirl** [set]
Sets a value indicating whether this [GameBehaviour](#) round girl.
- bool **Win** [set]
Sets a value indicating whether this [GameBehaviour](#) is window.
- bool **Lose** [set]
Sets a value indicating whether this [GameBehaviour](#) is lose.
- string **RoundNumber** [set]
Sets the round number.
- bool **AngrySpectators** [set]
Sets a value indicating whether this [GameBehaviour](#) angry spectators.
- bool **ExcitedSpectators** [set]
Sets a value indicating whether this [GameBehaviour](#) excited spectators.
- bool **DynamicCrowd** [get]
Gets a value indicating whether this [GameBehaviour](#) dynamic crowd.
- float **RelativeBloodMeasure** [get, set]
Gets or sets the relative blood measure.
- bool **Crowd** [set]
Sets a value indicating whether this [GameBehaviour](#) is crowd.
- bool **GamePad** [get, set]
Sets a value indicating whether this [GameBehaviour](#) game pad.
- AudioSource **soundEffects** [get]
Gets the sound effects.

3.12.1 Detailed Description

Game behaviour.

3.12.2 Member Function Documentation

3.12.2.1 void GameBehaviour.playTheBell ()

Plaies the bell.

3.12.2.2 void GameBehaviour.setCrowdBehaviour ()

Sets the crowd behaviour.

3.12.2.3 void GameBehaviour.setGameOverStats ()

Sets the game over stats.

3.12.2.4 void GameBehaviour.SetTransition (GameTransitions *t*)

Sets the transition.

Parameters

<i>t</i>	T.
----------	----

3.12.3 Property Documentation

3.12.3.1 bool GameBehaviour.AngrySpectators [set]

Sets a value indicating whether this [GameBehaviour](#) angry spectators.

true if angry spectators; otherwise, false.

3.12.3.2 bool GameBehaviour.Crowd [set]

Sets a value indicating whether this [GameBehaviour](#) is crowd.

true if crowd; otherwise, false.

3.12.3.3 bool GameBehaviour.DynamicCrowd [get]

Gets a value indicating whether this [GameBehaviour](#) dynamic crowd.

true if dynamic crowd; otherwise, false.

3.12.3.4 bool GameBehaviour.ExcitedSpectators [set]

Sets a value indicating whether this [GameBehaviour](#) excited spectators.

true if excited spectators; otherwise, false.

3.12.3.5 bool GameBehaviour.GamePad [get],[set]

Sets a value indicating whether this [GameBehaviour](#) game pad.

true if game pad; otherwise, false.

3.12.3.6 bool GameBehaviour.Lose [set]

Sets a value indicating whether this [GameBehaviour](#) is lose.

true if lose; otherwise, false.

3.12.3.7 **FighterModel GameBehaviour.MAC** [get], [set]

Gets or sets the MA.

The MA.

3.12.3.8 **int GameBehaviour.MaxRounds** [get]

Gets the max rounds.

The max rounds.

3.12.3.9 **FighterModel GameBehaviour.PlayerONE** [get], [set]

Gets or sets the player ON.

The player ON.

3.12.3.10 **float GameBehaviour.RelativeBloodMeasure** [get], [set]

Gets or sets the relative blood measure.

The relative blood measure.

3.12.3.11 **int GameBehaviour.Round** [get], [set]

Gets or sets the round.

The round.

3.12.3.12 **float GameBehaviour.RoundDuration** [get]

Gets the duration of the round.

The duration of the round.

3.12.3.13 **bool GameBehaviour.RoundGirl** [set]

Sets a value indicating whether this [GameBehaviour](#) round girl.

true if round girl; otherwise, false.

3.12.3.14 **string GameBehaviour.RoundNumber** [set]

Sets the round number.

The round number.

3.12.3.15 **AudioSource GameBehaviour.soundEffects** [get]

Gets the sound effects.

The sound effects.

3.12.3.16 bool GameBehaviour.Versus [set]

Sets a value indicating whether this [GameBehaviour](#) is versus.

true if versus; otherwise, false.

3.12.3.17 bool GameBehaviour.Win [set]

Sets a value indicating whether this [GameBehaviour](#) is window.

true if window; otherwise, false.

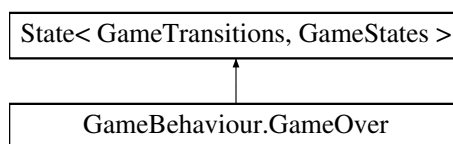
The documentation for this class was generated from the following file:

- behaviour/game/GameBehaviour.cs

3.13 GameBehaviour.GameOver Class Reference

Game Over

Inheritance diagram for GameBehaviour.GameOver:



Public Member Functions

- **GameOver** ([GameBehaviour](#) parent)
- override void **Reason** ()
- override void **Act** ()
- override void **DoBeforeEntering** ()
- override void **DoBeforeLeaving** ()

3.13.1 Detailed Description

Game Over

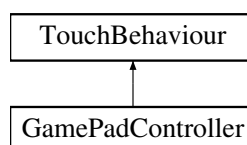
The documentation for this class was generated from the following file:

- behaviour/game/GameBehaviour.cs

3.14 GamePadController Class Reference

Game pad controller.

Inheritance diagram for GamePadController:



Public Member Functions

- override void [Start](#) ()
Start this instance.
- void [OnStartButtonBehaviour](#) ()
Raises the start button behaviour event.
- void [OnEndButtonBehaviour](#) ()
ON TOUCH END BEHAVIOUR
- override void [OnTouchBegan](#) (Vector3 value)
Raises the touch down event.
- override void [OnTouchEnded](#) (Vector3 value)
Raises the touch end event.

Public Attributes

- GamePadConfiguration **configuration**
- Sprite **buttonNormal**
- Sprite **buttonPushed**

3.14.1 Detailed Description

Game pad controller.

3.14.2 Member Function Documentation

3.14.2.1 void GamePadController.OnEndButtonBehaviour ()

ON TOUCH END BEHAVIOUR

3.14.2.2 void GamePadController.OnStartButtonBehaviour ()

Raises the start button behaviour event.

3.14.2.3 override void GamePadController.OnTouchBegan (Vector3 value)

Raises the touch down event.

3.14.2.4 override void GamePadController.OnTouchEnded (Vector3 value)

Raises the touch end event.

3.14.2.5 override void GamePadController.Start ()

Start this instance.

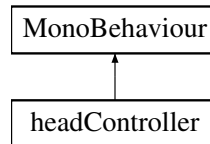
The documentation for this class was generated from the following file:

- behaviour/game/GamePadController.cs

3.15 headController Class Reference

Head controller.

Inheritance diagram for headController:



3.15.1 Detailed Description

Head controller.

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

- `behaviour/fighter/headController.cs`

3.16 HitData Class Reference

Public Attributes

- string **tag** = ""
- float **damage** = 0.0f

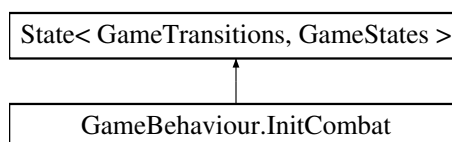
The documentation for this class was generated from the following file:

- `behaviour/fighter/PunchController.cs`

3.17 GameBehaviour.InitCombat Class Reference

Init combat.

Inheritance diagram for GameBehaviour.InitCombat:



Public Member Functions

- **InitCombat** ([GameBehaviour](#) parent, float duration=0.1f)
- override void **Reason** ()
- override void **Act** ()
- override void **DoBeforeEntering** ()
- override void **DoBeforeLeaving** ()

3.17.1 Detailed Description

Init combat.

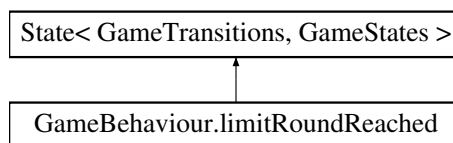
The documentation for this class was generated from the following file:

- `behaviour/game/GameBehaviour.cs`

3.18 GameBehaviour.limitRoundReached Class Reference

The limit Round is Reached

Inheritance diagram for GameBehaviour.limitRoundReached:



Public Member Functions

- **limitRoundReached** ([GameBehaviour](#) parent)
- override void **Reason** ()
- override void **Act** ()
- override void **DoBeforeEntering** ()
- override void **DoBeforeLeaving** ()

3.18.1 Detailed Description

The limit Round is Reached

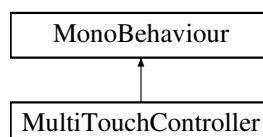
The documentation for this class was generated from the following file:

- `behaviour/game/GameBehaviour.cs`

3.19 MultiTouchController Class Reference

Multi touch controller.

Inheritance diagram for MultiTouchController:



Public Attributes

- LayerMask **touchInputMask**

3.19.1 Detailed Description

Multi touch controller.

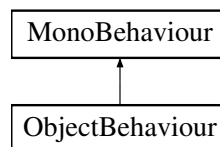
The documentation for this class was generated from the following file:

- behaviour/game/MultiTouchController.cs

3.20 ObjectBehaviour Class Reference

Object behaviour. Behaviour of the objects thrown by spectators.

Inheritance diagram for ObjectBehaviour:



Public Attributes

- float **blinkRate** = 6.0f
- float **rotationSpeed** = 500.0f
- Vector2 **timeOutRange** = new Vector2(2.0f,3.0f)
- bool **blink** = true
- float **damage** = 10.0f

3.20.1 Detailed Description

Object behaviour. Behaviour of the objects thrown by spectators.

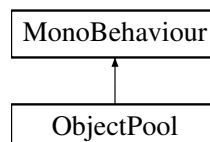
The documentation for this class was generated from the following file:

- behaviour/field/ObjectBehaviour.cs

3.21 ObjectPool Class Reference

Object pool.

Inheritance diagram for ObjectPool:



Public Member Functions

- GameObject **activateObject** ()
Activates the object.

Public Attributes

- `GameObject[] objectPool`
- `int numberOfObjects = 0`
- `GameObject prefab`

3.21.1 Detailed Description

Object pool.

3.21.2 Member Function Documentation

3.21.2.1 `GameObject ObjectPool.activateObject ()`

Activates the object.

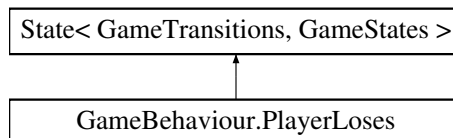
The documentation for this class was generated from the following file:

- `behaviour/game/ObjectPool.cs`

3.22 GameBehaviour.PlayerLoses Class Reference

PLayer Loses.

Inheritance diagram for `GameBehaviour.PlayerLoses`:



Public Member Functions

- **PlayerLoses** ([GameBehaviour](#) parent)
- override void **Reason** ()
- override void **Act** ()
- override void **DoBeforeEntering** ()
- override void **DoBeforeLeaving** ()

3.22.1 Detailed Description

PLayer Loses.

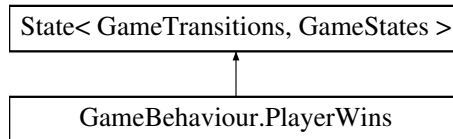
The documentation for this class was generated from the following file:

- `behaviour/game/GameBehaviour.cs`

3.23 GameBehaviour.PlayerWins Class Reference

Player Wins.

Inheritance diagram for GameBehaviour.PlayerWins:



Public Member Functions

- **PlayerWins** ([GameBehaviour](#) parent)
- override void **Reason** ()
- override void **Act** ()
- override void **DoBeforeEntering** ()
- override void **DoBeforeLeaving** ()

3.23.1 Detailed Description

Player Wins.

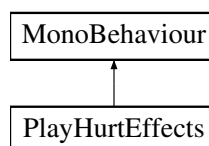
The documentation for this class was generated from the following file:

- `behaviour/game/GameBehaviour.cs`

3.24 PlayHurtEffects Class Reference

Play hurt effects.

Inheritance diagram for PlayHurtEffects:



Public Member Functions

- void [ApplyDamage](#) ([HitData](#) hitData)
Applies the damage.

Public Attributes

- AudioClip **hurtSFX**
- int **particles** = 3

3.24.1 Detailed Description

Play hurt effects.

3.24.2 Member Function Documentation

3.24.2.1 void PlayHurtEffects.ApplyDamage (HitData *hitData*)

Applies the damage.

Parameters

<i>hitData</i>	Hit data.
----------------	-----------

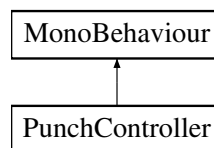
The documentation for this class was generated from the following file:

- behaviour/fighter/PlayHurtEffects.cs

3.25 PunchController Class Reference

Punch controller.

Inheritance diagram for PunchController:



Properties

- float **effectivePower** [get, set]
Gets or sets the effective power.
- float **defense** [get, set]
Gets or sets the defense. NOT IN USE.
- int **Hits** [get]

3.25.1 Detailed Description

Punch controller.

3.25.2 Property Documentation

3.25.2.1 float PunchController.defense [get], [set]

Gets or sets the defense. NOT IN USE.

The defense.

3.25.2.2 float PunchController.effectivePower [get],[set]

Gets or sets the effective power.

The effective power.

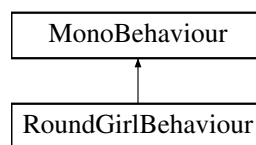
The documentation for this class was generated from the following file:

- behaviour/fighter/PunchController.cs

3.26 RoundGirlBehaviour Class Reference

Round girl behaviour.

Inheritance diagram for RoundGirlBehaviour:



3.26.1 Detailed Description

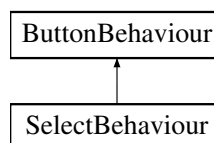
Round girl behaviour.

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

- behaviour/field/RoundGirlBehaviour.cs

3.27 SelectBehaviour Class Reference

Inheritance diagram for SelectBehaviour:



Public Member Functions

- void **enable** (bool value)

Public Attributes

- int **buttonID** = 0
- Sprite **buttonON**
- Sprite **buttonOFF**
- GameObject **associatedObject**

Protected Member Functions

- override void **OnEnable** ()
- override void **action** ()

Raises the mouse down event.

3.27.1 Member Function Documentation

3.27.1.1 override void SelectBehaviour.action () [protected]

Raises the mouse down event.

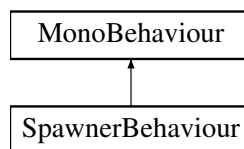
The documentation for this class was generated from the following file:

- behaviour/game/SelectBehaviour.cs

3.28 SpawnerBehaviour Class Reference

Spawner behaviour.

Inheritance diagram for SpawnerBehaviour:



Public Attributes

- Vector2 **timeRange** = new Vector2(5.0f,10.0f)
- Vector2 **force** = new Vector2(0,-200)

3.28.1 Detailed Description

Spawner behaviour.

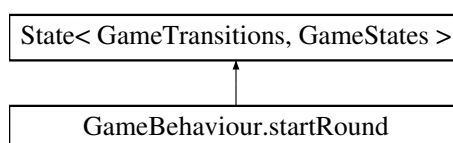
The documentation for this class was generated from the following file:

- behaviour/field/SpawnerBehaviour.cs

3.29 GameBehaviour.startRound Class Reference

Start round.

Inheritance diagram for GameBehaviour.startRound:



Public Member Functions

- **startRound** ([GameBehaviour](#) parent, float duration=0.1f)
- override void **Reason** ()
- override void **Act** ()
- override void **DoBeforeEntering** ()
- override void **DoBeforeLeaving** ()

3.29.1 Detailed Description

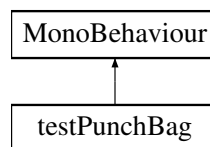
Start round.

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

- behaviour/game/GameBehaviour.cs

3.30 testPunchBag Class Reference

Inheritance diagram for testPunchBag:



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

- test/testPunchBag.cs

Index

- action
 - SelectBehaviour, [34](#)
- activateObject
 - ObjectPool, [30](#)
- activeInjured
 - FighterController, [12](#)
- Add
 - FingerEvent, [19](#)
- AngrySpectators
 - GameBehaviour, [23](#)
- AnimSync
 - FighterController, [11](#)
- ApplyDamage
 - FighterController, [11](#)
 - PlayHurtEffects, [32](#)
- Avatar
 - FighterController, [12](#)
 - FighterInjuredBehaviour, [16](#)
- BloodBehaviour, [5](#)
 - timeOutRange, [5](#)
- BloodMeter, [5](#)
 - Measure, [6](#)
 - RelativeMeasure, [6](#)
- BloodMeterModel, [6](#)
- Body
 - FighterInjuredBehaviour, [16](#)
- clearList
 - FingerEvent, [19](#)
- containsObject
 - FingerEvent, [19](#)
- Crowd
 - GameBehaviour, [23](#)
- currentState
 - FighterController, [12](#)
- Damage
 - FighterInjuredBehaviour, [16](#)
- defense
 - PunchController, [32](#)
- DynamicCrowd
 - GameBehaviour, [23](#)
- EffectivePower
 - FighterController, [12](#)
- effectivePower
 - PunchController, [32](#)
- Enable
 - FighterController, [12](#)
- EnableCountDown
 - FighterController, [12](#)
- ExcitedSpectators
 - GameBehaviour, [23](#)
- FPSCounter, [20](#)
- FighterController, [8](#)
 - activeInjured, [12](#)
 - AnimSync, [11](#)
 - ApplyDamage, [11](#)
 - Avatar, [12](#)
 - currentState, [12](#)
 - EffectivePower, [12](#)
 - Enable, [12](#)
 - EnableCountDown, [12](#)
 - GoToInitialPosition, [11](#)
 - goToInitialPosition, [11](#)
 - Health, [12](#)
 - HealthBar, [13](#)
 - initializeVariables, [11](#)
 - InjuredAvatar, [13](#)
 - InjuredDamage, [13](#)
 - Injuries, [13](#)
 - IsActive, [13](#)
 - isDead, [13](#)
 - IsHuman, [13](#)
 - isSuperPunchLoaded, [13](#)
 - KnockOutInfo, [13](#)
 - KnockOuts, [14](#)
 - MaxHealth, [14](#)
 - MaxSuperPunchHits, [14](#)
 - OnHitSucceed, [14](#)
 - playBlockEffects, [11](#)
 - playGroggyEffects, [11](#)
 - playPunchEffects, [11](#)
 - Position, [14](#)
 - PowerBar, [14](#)
 - RelativeHealth, [14](#)
 - RelativePower, [14](#)
 - SetTransition, [11](#)
 - stopGroggyEffects, [12](#)
 - SucceedHits, [14](#)
 - SuperPunchHits, [15](#)
 - Tag, [15](#)
 - TotalSucceedHits, [15](#)
 - updateFighterHealth, [12](#)
 - updateFighterPower, [12](#)
- FighterInjuredBehaviour, [15](#)
 - Avatar, [16](#)
 - Body, [16](#)

- Damage, 16
- HitLimit, 16
- Injuries, 16
- setInjuries, 16
- FighterInjuryBehaviour, 17
 - HitsLimit, 17
 - OnEnable, 17
 - OnTouchBegan, 17
- FighterModel, 18
- FingerEvent, 18
 - Add, 19
 - clearList, 19
 - containsObject, 19
 - FingerEvent, 19
 - getLast, 20
 - remove, 20
- GameBehaviour, 20
 - AngrySpectators, 23
 - Crowd, 23
 - DynamicCrowd, 23
 - ExcitedSpectators, 23
 - GamePad, 23
 - Lose, 23
 - MAC, 23
 - MaxRounds, 24
 - playTheBell, 22
 - PlayerONE, 24
 - RelativeBloodMeasure, 24
 - Round, 24
 - RoundDuration, 24
 - RoundGirl, 24
 - RoundNumber, 24
 - setCrowdBehaviour, 22
 - setGameOverStats, 23
 - SetTransition, 23
 - soundEffects, 24
 - Versus, 24
 - Win, 25
- GameBehaviour.endRound, 7
- GameBehaviour.fight, 8
- GameBehaviour.GameOver, 25
- GameBehaviour.InitCombat, 27
- GameBehaviour.limitRoundReached, 28
- GameBehaviour.PlayerLoses, 30
- GameBehaviour.PlayerWins, 31
- GameBehaviour.startRound, 34
- GamePad
 - GameBehaviour, 23
- GamePadController, 25
 - OnEndButtonBehaviour, 26
 - OnStartButtonBehaviour, 26
 - OnTouchBegan, 26
 - OnTouchEnded, 26
 - Start, 26
- getLast
 - FingerEvent, 20
- GoToInitialPosition
 - FighterController, 11
- goToInitialPosition
 - FighterController, 11
- headController, 27
- Health
 - FighterController, 12
- HealthBar
 - FighterController, 13
- HitData, 27
- HitLimit
 - FighterInjuredBehaviour, 16
- HitsLimit
 - FighterInjuryBehaviour, 17
- initializeVariables
 - FighterController, 11
- InjuredAvatar
 - FighterController, 13
- InjuredDamage
 - FighterController, 13
- Injuries
 - FighterController, 13
 - FighterInjuredBehaviour, 16
- IsActive
 - FighterController, 13
- isDead
 - FighterController, 13
- IsHuman
 - FighterController, 13
- isSuperPunchLoaded
 - FighterController, 13
- KnockOutInfo
 - FighterController, 13
- KnockOuts
 - FighterController, 14
- Lose
 - GameBehaviour, 23
- MAC
 - GameBehaviour, 23
- MaxHealth
 - FighterController, 14
- MaxRounds
 - GameBehaviour, 24
- MaxSuperPunchHits
 - FighterController, 14
- Measure
 - BloodMeter, 6
- MultiTouchController, 28
- ObjectBehaviour, 29
- ObjectPool, 29
 - activateObject, 30
- OnEnable
 - FighterInjuryBehaviour, 17
- OnEndButtonBehaviour
 - GamePadController, 26
- OnHitSucceed

- FighterController, 14
- OnStartButtonBehaviour
 - GamePadController, 26
- OnTouchBegan
 - FighterInjuryBehaviour, 17
 - GamePadController, 26
- OnTouchEnded
 - GamePadController, 26
- playBlockEffects
 - FighterController, 11
- playGroggyEffects
 - FighterController, 11
- PlayHurtEffects, 31
 - ApplyDamage, 32
- playPunchEffects
 - FighterController, 11
- playTheBell
 - GameBehaviour, 22
- PlayerONE
 - GameBehaviour, 24
- Position
 - FighterController, 14
- PowerBar
 - FighterController, 14
- PunchController, 32
 - defense, 32
 - effectivePower, 32
- RelativeBloodMeasure
 - GameBehaviour, 24
- RelativeHealth
 - FighterController, 14
- RelativeMeasure
 - BloodMeter, 6
- RelativePower
 - FighterController, 14
- remove
 - FingerEvent, 20
- Round
 - GameBehaviour, 24
- RoundDuration
 - GameBehaviour, 24
- RoundGirl
 - GameBehaviour, 24
- RoundGirlBehaviour, 33
- RoundNumber
 - GameBehaviour, 24
- SelectBehaviour, 33
 - action, 34
- setCrowdBehaviour
 - GameBehaviour, 22
- setGameOverStats
 - GameBehaviour, 23
- setInjuries
 - FighterInjuredBehaviour, 16
- SetTransition
 - FighterController, 11
- GameBehaviour, 23
- soundEffects
 - GameBehaviour, 24
- SpawnerBehaviour, 34
- Start
 - GamePadController, 26
- stopGroggyEffects
 - FighterController, 12
- SucceedHits
 - FighterController, 14
- SuperPunchHits
 - FighterController, 15
- Tag
 - FighterController, 15
- testPunchBag, 35
- timeOutRange
 - BloodBehaviour, 5
- TotalSucceedHits
 - FighterController, 15
- updateFighterHealth
 - FighterController, 12
- updateFighterPower
 - FighterController, 12
- Versus
 - GameBehaviour, 24
- Win
 - GameBehaviour, 25