SuperPunch

3.50

Generated by Doxygen 1.8.10

Mon Apr 10 2017 10:38:04

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:

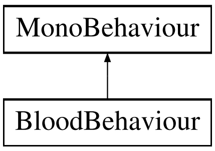
Chapter 3

Class Documentation

3.1 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.

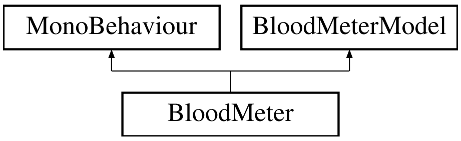
The documentation for this class was generated from the following file:

- 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.

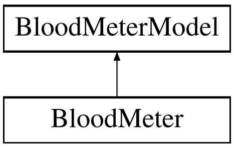
The documentation for this class was generated from the following file:

- 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.

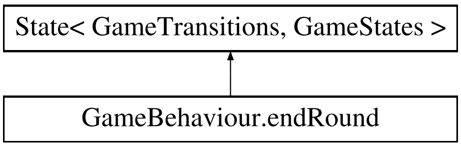
The documentation for this interface was generated from the following file:

- 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.

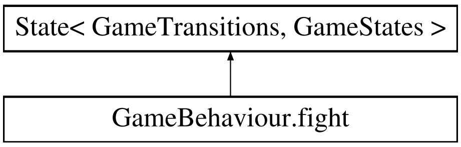
The documentation for this class was generated from the following file:

- 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.

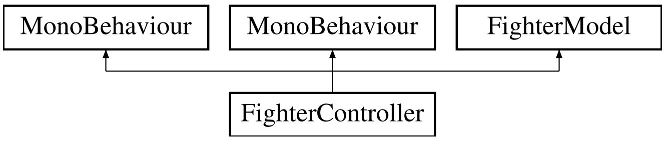
The documentation for this class was generated from the following file:

- 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

hitData	hitData.

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 ( )

Plaies the block effects.

3.6.2.7 void FighterController.playGroggyEffects ( )

Plaies the groggy effects.

3.6.2.8 void FighterController.playPunchEffects ( )

Plaies 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.

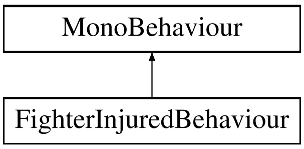
The documentation for this class was generated from the following file:

- 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

damage	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.

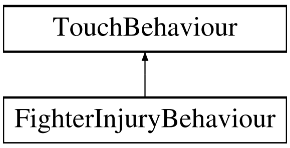
The documentation for this class was generated from the following file:

- 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.

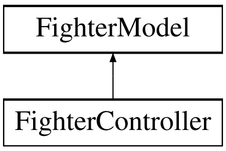
The documentation for this class was generated from the following file:

- 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

it	It.

3.10.3 Member Function Documentation

3.10.3.1 bool FingerEvent.Add (ITouchable *obj*)

Add the specified obj.

Parameters

obj	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 containsed, `false` otherwise.

Parameters

obj	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

obj	Object.

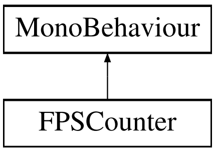
The documentation for this class was generated from the following file:

- behaviour/game/MultiTouchController.cs

3.11 FPSCounter Class Reference

FPS counter.

Inheritance diagram for FPSCounter:



3.11.1 Detailed Description

FPS counter.

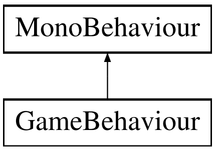
The documentation for this class was generated from the following file:

- 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

t	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`.

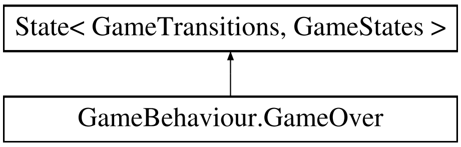
The documentation for this class was generated from the following file:

- 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

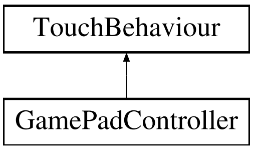
The documentation for this class was generated from the following file:

- 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 OnStartButtonBehaivour ()

  Raises the start button behaivour event.
- void OnEndButtonBehaivour ()

  ON TOUCH END BEHAIVOUR
- 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.OnEndButtonBehaivour ( )

ON TOUCH END BEHAIVOUR

3.14.2.2 void GamePadController.OnStartButtonBehaivour ( )

Raises the start button behaivour 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.

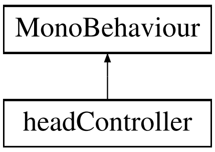
The documentation for this class was generated from the following file:

- 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

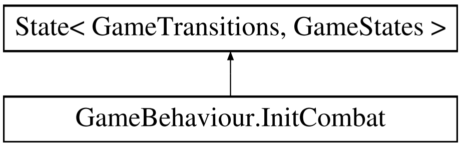
The documentation for this class was generated from the following file:

- 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.

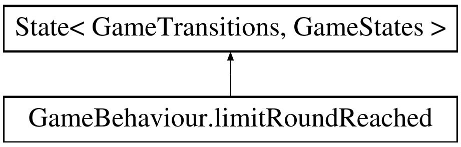
The documentation for this class was generated from the following file:

- behaviour/game/GameBehaviour.cs

3.18 GameBehaviour.limitRoundReached Class Reference

The limit Round is Reached

Inheritance diagram for GameBehaviour.limitRoundReached:


State< GameTransitions, GameStates >
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

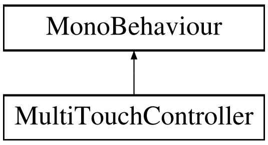
The documentation for this class was generated from the following file:

- 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.

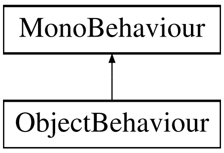
The documentation for this class was generated from the following file:

- 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.

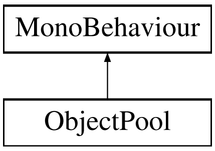
The documentation for this class was generated from the following file:

- 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.

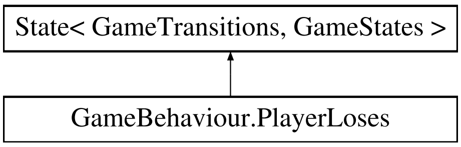
The documentation for this class was generated from the following file:

- behaviour/game/ObjectPool.cs

3.22 GameBehaviour.PlayerLoses Class Reference

PLayer Loses.

Inheritance diagram for GameBehaviour.PlayerLoses:


State< GameTransitions, GameStates >
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.

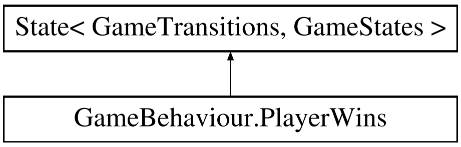
The documentation for this class was generated from the following file:

- 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.

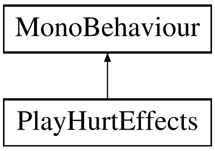
The documentation for this class was generated from the following file:

- 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

hitData	Hit data.

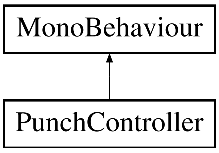
The documentation for this class was generated from the following file:

- 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.

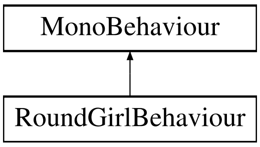
The documentation for this class was generated from the following file:

- behaviour/fighter/PunchController.cs

3.26 RoundGirlBehaviour Class Reference

Round girl behaviour.

Inheritance diagram for RoundGirlBehaviour:



3.26.1 Detailed Description

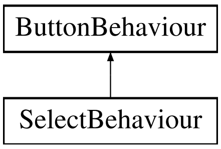
Round girl behaviour.

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.

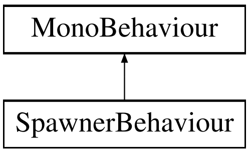
The documentation for this class was generated from the following file:

- 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.

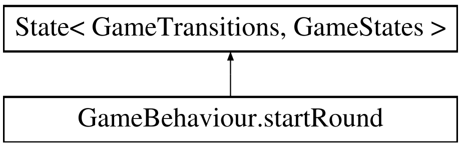
The documentation for this class was generated from the following file:

- 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

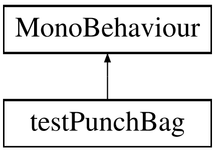
Start round.

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