

Team Ball Game Template

Generated by Doxygen 1.8.14

Contents

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

TeamBallGame	??
TeamBallGame.Gameplay	??
TeamBallGame.Mechanics	??
TeamBallGame.Model	??
TeamBallGame.UI	??
TeamBallGame.Visual	??
TMPro	??
TMPro.Examples	??

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

TeamBallGame.Mechanics.BallGameConfig	??
TeamBallGame.Model.BallGameModel	??
TeamBallGame.ComponentPool< T >	??
TeamBallGame.Simulation.Event	??
TeamBallGame.Gameplay.BallContest	??
TeamBallGame.Gameplay.BallInterception	??
TeamBallGame.Gameplay.BarricadeBallCollision	??
TeamBallGame.Gameplay.PassBall	??
TeamBallGame.Gameplay.ResolveBallContest	??
TeamBallGame.Gameplay.SuspendPlay	??
TeamBallGame.Simulation.Event< T >	??
TeamBallGame.Simulation.Event< BallBounce >	??
TeamBallGame.Gameplay.BallBounce	??
TeamBallGame.Simulation.Event< BallIsLaunched >	??
TeamBallGame.Gameplay.BallIsLaunched	??
TeamBallGame.Simulation.Event< BallUp >	??
TeamBallGame.Gameplay.BallUp	??
TeamBallGame.Simulation.Event< DisableUserInput >	??
TeamBallGame.Gameplay.DisableUserInput	??
TeamBallGame.Simulation.Event< EnableUserInput >	??
TeamBallGame.Gameplay.EnableUserInput	??
TeamBallGame.Simulation.Event< HeadBallCollision >	??
TeamBallGame.Gameplay.HeadBallCollision	??
TeamBallGame.Simulation.Event< LaunchBall >	??
TeamBallGame.Gameplay.LaunchBall	??
TeamBallGame.Simulation.Event< LookDirectionChanged >	??
TeamBallGame.Gameplay.LookDirectionChanged	??
TeamBallGame.Simulation.Event< PlayerBallCollision >	??
TeamBallGame.Gameplay.PlayerBallCollision	??
TeamBallGame.Simulation.Event< PlayerHasBeenTackled >	??
TeamBallGame.Gameplay.PlayerHasBeenTackled	??
TeamBallGame.Simulation.Event< PlayerJump >	??

TeamBallGame.Gameplay.PlayerJump	??
TeamBallGame.Simulation.Event< PlayerMovement >	??
TeamBallGame.Gameplay.PlayerMovement	??
TeamBallGame.Simulation.Event< PlayerRecoversFromTackle >	??
TeamBallGame.Gameplay.PlayerRecoversFromTackle	??
TeamBallGame.Simulation.Event< PlayerTackle >	??
TeamBallGame.Gameplay.PlayerTackle	??
TeamBallGame.Simulation.Event< PrepareToLaunchBall >	??
TeamBallGame.Gameplay.PrepareToLaunchBall	??
TeamBallGame.Simulation.Event< PrepareToPassBall >	??
TeamBallGame.Gameplay.PrepareToPassBall	??
TeamBallGame.Simulation.Event< ReceiveBall >	??
TeamBallGame.Gameplay.ReceiveBall	??
TeamBallGame.Simulation.Event< RepositionArrowIndicator >	??
TeamBallGame.Gameplay.RepositionArrowIndicator	??
TeamBallGame.Simulation.Event< ResetGamePlay >	??
TeamBallGame.Gameplay.ResetGamePlay	??
TeamBallGame.Simulation.Event< Score >	??
TeamBallGame.Gameplay.Score	??
TeamBallGame.Simulation.Event< StartGameplay >	??
TeamBallGame.Gameplay.StartGameplay	??
TeamBallGame.HeapQueue< T >	??
TeamBallGame.HeapQueue< PendingReturn >	??
ICollectionable	
TeamBallGame.Simulation.Event< T >	??
MonoBehaviour	
TeamBallGame.Barricade	??
TeamBallGame.Mechanics.CrowdController	??
TeamBallGame.Mechanics.DirectionIndicator	??
TeamBallGame.Mechanics.GameConfiguration	??
TeamBallGame.Mechanics.GameController	??
TeamBallGame.Mechanics.ImpactAudio	??
TeamBallGame.Mechanics.LookIndicator	??
TeamBallGame.Mechanics.MetaGameController	??
TeamBallGame.Mechanics.MovementController	??
TeamBallGame.Mechanics.Scoreboard	??
TeamBallGame.Mechanics.Umpire	??
TeamBallGame.Mechanics.UserInput	??
TeamBallGame.Model.Ball	??
TeamBallGame.Model.FieldPosition	??
TeamBallGame.Model.Goal	??
TeamBallGame.Model.Player	??
TeamBallGame.Model.Team	??
TeamBallGame.UI.MainUIController	??
TMPro.Examples.VertexZoom	??
TeamBallGame.PlayingField	??
TeamBallGame.Oval	??
TeamBallGame.Quadrangle	??
StateMachineBehaviour	
TeamBallGame.Visual.IdleIndexSetter	??

Chapter 3

Class Index

3.1 Class List

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

TeamBallGame.Model.Ball	This behaviour fires ball events and provides a Ball API to the designer.	??
TeamBallGame.Gameplay.BallBounce	This event occurs whenever the ball bounces on the field.	??
TeamBallGame.Gameplay.BallContest	This event is scheduled when players come into contact with the ball or the player in possession of the ball. The actual contest (with all participants) is resolved later.	??
TeamBallGame.Mechanics.BallGameConfig	This class is designed to be accessed via <code>Simulation.GetModel</code> . It contains scene specific configuration items which are not directly related to gameplay. Gameplay specific data is stored in the <code>BallGameModel</code> class.	??
TeamBallGame.Model.BallGameModel	This is a general model containing all data required for simulation of a ball game. It needs to be initialised at the start of a game by the <code>GameController</code> . It also contains methods for getting useful information from the model.	??
TeamBallGame.Gameplay.BallInterception		??
TeamBallGame.Gameplay.BallIsLaunched	This event is fired when the ball has been successfully launched by a player.	??
TeamBallGame.Gameplay.BallUp	This event occurs when the ball is stationary, or thrown in / bounced by an umpire and the two nearest players will compete for possession.	??
TeamBallGame.Barricade		??
TeamBallGame.Gameplay.BarricadeBallCollision	This event occurs when the ball hits a barricade. Logic for scheduling out-of-bounds events would happen here.	??
TeamBallGame.ComponentPool< T >		??
TeamBallGame.Mechanics.CrowdController		??
TeamBallGame.Mechanics.DirectionIndicator	Sets the location of a transform and looks at another transform. Used for providing visual feedback about directions to the user.	??
TeamBallGame.Gameplay.DisableUserInput	This event is fired when user input has been disabled for gameplay reasons (eg referee suspends play, goal is scored)	??
TeamBallGame.Gameplay.EnableUserInput	This event is fired when user input has been reenabled for gameplay reasons (eg Ball up contest started)	??

TeamBallGame.Simulation.Event	??
TeamBallGame.Simulation.Event< T >	??
TeamBallGame.Model.FieldPosition	
A named position on a playing field, for example "Goal Keeper".	??
TeamBallGame.Mechanics.GameConfiguration	
This behaviour holds configuration information for gameplay, and stores it in the Simulation Ball↔ GameModel instance when initialised.	??
TeamBallGame.Mechanics.GameController	??
TeamBallGame.Model.Goal	
This behaviour is attached to goal trigger volumes, and will fire a Score event when a Ball com- ponent enters the trigger.	??
TeamBallGame.Gameplay.HeadBallCollision	
This event is fired when the head collider of a player makes contact with the ball.	??
TeamBallGame.HeapQueue< T >	??
TeamBallGame.Visual.IdleIndexSetter	??
TeamBallGame.Mechanics.ImpactAudio	
Plays impact audio, taking into account distance from audio listener and the speed of sound in air at 20C.	??
TeamBallGame.Gameplay.LaunchBall	
This event occurs when a player launches the ball. It is usually followed by a BallIsLaunched event.	??
TeamBallGame.Gameplay.LookDirectionChanged	
This event is fired when the user input has changed the look direction of the active player.	??
TeamBallGame.Mechanics.LookIndicator	
A specialised Directional Indicator used to indicate the current pass direction, and if the pass direction will intercept a team mate.	??
TeamBallGame.UI.MainUIController	??
TeamBallGame.Mechanics.MetaGameController	
The MetaGameController is responsible for switching control between the high level contexts of the application, eg the Main Menu and Gameplay systems.	??
TeamBallGame.Mechanics.MovementController	??
TeamBallGame.Oval	??
TeamBallGame.Gameplay.PassBall	??
TeamBallGame.Model.Player	??
TeamBallGame.Gameplay.PlayerBallCollision	
This event is fired when a player collides with the ball. Generally, this would start a BallContest .	??
TeamBallGame.Gameplay.PlayerHasBeenTackled	
This event is fired when a player has been tackled.	??
TeamBallGame.Gameplay.PlayerJump	
This event is fired when the jump input is activated.	??
TeamBallGame.Gameplay.PlayerMovement	??
TeamBallGame.Gameplay.PlayerRecoversFromTackle	??
TeamBallGame.Gameplay.PlayerTackle	
This event is triggered when the tackle input is received.	??
TeamBallGame.PlayingField	??
TeamBallGame.Gameplay.PrepareToLaunchBall	
This event is fired when a player is launching the ball at a target position.	??
TeamBallGame.Gameplay.PrepareToPassBall	
This event is fired when a player is passing the ball.	??
TeamBallGame.Quadrangle	??
TeamBallGame.Gameplay.ReceiveBall	
This event is fired when a player receives control of the ball.	??
TeamBallGame.Gameplay.RepositionArrowIndicator	??
TeamBallGame.Gameplay.ResetGamePlay	
Reset game play to a ball up or kick off state. Eg, move ball to center and all players to field positions.	??

[TeamBallGame.Gameplay.ResolveBallContest](#)

This event is continuously scheduled so that pending ball contests can be resolved and ball possession can be changed accordingly. ??

[TeamBallGame.Gameplay.Score](#)

This event is fired when the ball enters a Goal trigger. ??

[TeamBallGame.Mechanics.Scoreboard](#)

[Scoreboard](#) waits for ScoreEvents and updates text widgets with the new score values. ??

[TeamBallGame.Gameplay.StartGameplay](#)

This event is triggered when gameplay should start, eg at the start of a game period or after the ball is returned to the center. ??

[TeamBallGame.Gameplay.SuspendPlay](#)

Suspend play occurs when players must stop and allow a penalty, ball up or some other event to occur before resuming. ??

[TeamBallGame.Model.Team](#)

??

[TeamBallGame.Mechanics.Umpire](#)

??

[TeamBallGame.Mechanics.UserInput](#)

??

[TMPro.Examples.VertexZoom](#)

??

Chapter 4

Namespace Documentation

4.1 TeamBallGame Namespace Reference

Namespaces

Classes

- class [Barricade](#)
- class [ComponentPool](#)
- class **Fuzzy**
- class [HeapQueue](#)
- class [Oval](#)
- class [PlayingField](#)
- class [Quadrangle](#)
- class **Simulation**

The Simulation class implements the discrete event simulator pattern. Events are pooled.

4.2 TeamBallGame.Gameplay Namespace Reference

Classes

- class [BallBounce](#)
This event occurs whenever the ball bounces on the field.
- class [BallContest](#)
This event is scheduled when players come into contact with the ball or the player in possession of the ball. The actual contest (with all participants) is resolved later.
- class [BallInterception](#)
- class [BallsLaunched](#)
This event is fired when the ball has been successfully launched by a player.
- class [BallUp](#)
This event occurs when the ball is stationary, or thrown in / bounced by an umpire and the two nearest players will compete for possession.
- class [BarricadeBallCollision](#)
This event occurs when the ball hits a barricade. Logic for scheduling out-of-bounds events would happen here.

- class [DisableUserInput](#)
This event is fired when user input has been disabled for gameplay reasons (eg referee suspends play, goal is scored)
- class [EnableUserInput](#)
This event is fired when user input has been reenabled for gameplay reasons (eg Ball up contest started)
- class [HeadBallCollision](#)
This event is fired when the head collider of a player makes contact with the ball.
- class [LaunchBall](#)
This event occurs when a player launches the ball. It is usually followed by a [BallIsLaunched](#) event.
- class [LookDirectionChanged](#)
This event is fired when the user input has changed the look direction of the active player.
- class [PassBall](#)
- class [PlayerBallCollision](#)
This event is fired when a player collides with the ball. Generally, this would start a [BallContest](#).
- class [PlayerHasBeenTackled](#)
This event is fired when a player has been tackled.
- class [PlayerJump](#)
This event is fired when the jump input is activated.
- class [PlayerMovement](#)
- class [PlayerRecoversFromTackle](#)
- class [PlayerTackle](#)
This event is triggered when the tackle input is received.
- class [PrepareToLaunchBall](#)
This event is fired when a player is launching the ball at a target position.
- class [PrepareToPassBall](#)
This event is fired when a player is passing the ball.
- class [ReceiveBall](#)
This event is fired when a player receives control of the ball.
- class [RepositionArrowIndicator](#)
- class [ResetGamePlay](#)
Reset game play to a ball up or kick off state. Eg, move ball to center and all players to field positions.
- class [ResolveBallContest](#)
This event is continuously scheduled so that pending ball contests can be resolved and ball possession can be changed accordingly.
- class [Score](#)
This event is fired when the ball enters a Goal trigger.
- class [StartGameplay](#)
This event is triggered when gameplay should start, eg at the start of a game period or after the ball is returned to the center.
- class [SuspendPlay](#)
Suspend play occurs when players must stop and allow a penalty, ball up or some other event to occur before resuming.

4.3 TeamBallGame.Mechanics Namespace Reference

Classes

- class [BallGameConfig](#)
This class is designed to be accessed via `Simulation.GetModel`. It contains scene specific configuration items which are not directly related to gameplay. [Gameplay](#) specific data is stored in the `BallGameModel` class.
- class [CrowdController](#)

- class [DirectionIndicator](#)
Sets the location of a transform and looks at another transform. Used for providing visual feedback about directions to the user.
- class [GameConfiguration](#)
This behaviour holds configuration information for gameplay, and stores it in the Simulation BallGameModel instance when initialised.
- class [GameController](#)
- class [ImpactAudio](#)
Plays impact audio, taking into account distance from audio listener and the speed of sound in air at 20C.
- class [LookIndicator](#)
A specialised Directional Indicator used to indicate the current pass direction, and if the pass direction will intercept a team mate.
- class [MetaGameController](#)
The [MetaGameController](#) is responsible for switching control between the high level contexts of the application, eg the Main Menu and [Gameplay](#) systems.
- class [MovementController](#)
- class [Scoreboard](#)
[Scoreboard](#) waits for ScoreEvents and updates text widgets with the new score values.
- class [Umpire](#)
- class [UserInput](#)

4.4 TeamBallGame.Model Namespace Reference

Classes

- class [Ball](#)
This behaviour fires ball events and provides a [Ball](#) API to the designer.
- class [BallGameModel](#)
This is a general model containing all data required for simulation of a ball game. It needs to be initialised at the start of a game by the GameController. It also contains methods for getting useful information from the model.
- class [FieldPosition](#)
A named position on a playing field, for example "Goal Keeper".
- class [Goal](#)
This behaviour is attached to goal trigger volumes, and will fire a Score event when a [Ball](#) component enters the trigger.
- class [Player](#)
- class [Team](#)

Enumerations

- enum **TeamType** { Home, Away }

4.5 TeamBallGame.UI Namespace Reference

Classes

- class [MainUIController](#)

4.6 TeamBallGame.Visual Namespace Reference

Classes

- class [IdleIndexSetter](#)

4.7 TPro Namespace Reference

Namespaces

4.8 TPro.Examples Namespace Reference

Classes

- class [VertexZoom](#)

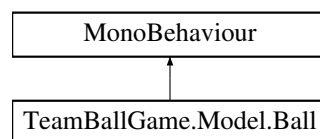
Chapter 5

Class Documentation

5.1 TeamBallGame.Model.Ball Class Reference

This behaviour fires ball events and provides a [Ball](#) API to the designer.

Inheritance diagram for TeamBallGame.Model.Ball:



Public Attributes

- new Rigidbody **rigidbody**
- [ImpactAudio](#) **impactAudio**
- bool **IsPossessedByHomeTeam** => ballGame.playerInPossession == null ? false : ballGame.playerInPossession.team.teamType == TeamType.Home
- bool **IsPossessedByAwayTeam** => ballGame.playerInPossession == null ? false : ballGame.playerInPossession.team.teamType == TeamType.Away
- bool **IsPossessed** => ballGame.playerInPossession != null
- float **Height** => transform.position.y

Properties

- bool **IsInPlay** [get, set]

5.1.1 Detailed Description

This behaviour fires ball events and provides a [Ball](#) API to the designer.

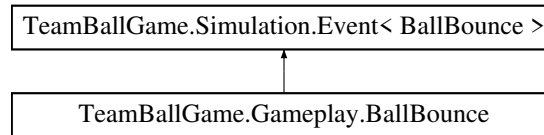
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Model/Ball.cs

5.2 TeamBallGame.Gameplay.BallBounce Class Reference

This event occurs whenever the ball bounces on the field.

Inheritance diagram for TeamBallGame.Gameplay.BallBounce:



Public Member Functions

- override void **Execute** ()

Public Attributes

- Collision **collision**

5.2.1 Detailed Description

This event occurs whenever the ball bounces on the field.

Template Parameters

<i>BallBounce</i>	
-----------------------------------	--

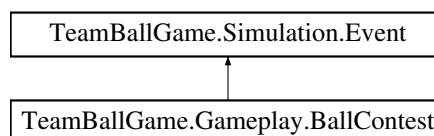
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Gameplay/BallBounce.cs

5.3 TeamBallGame.Gameplay.BallContest Class Reference

This event is scheduled when players come into contact with the ball or the player in possession of the ball. The actual contest (with all participants) is resolved later.

Inheritance diagram for TeamBallGame.Gameplay.BallContest:



Public Member Functions

- override void **Execute** ()

Public Attributes

- [Player](#) **player**

5.3.1 Detailed Description

This event is scheduled when players come into contact with the ball or the player in possession of the ball. The actual contest (with all participants) is resolved later.

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

- Assets/TeamBallGameTemplate/Scripts/Gameplay/BallContest.cs

5.4 TeamBallGame.Mechanics.BallGameConfig Class Reference

This class is designed to be accessed via `Simulation.GetModel`. It contains scene specific configuration items which are not directly related to gameplay. [Gameplay](#) specific data is stored in the `BallGameModel` class.

Public Attributes

- Sprite **aiControlIcon**
- Transform **reticle**
- [LookIndicator](#) **arrowIndicator**
- [DirectionIndicator](#) **activeGoalDirectionIndicator**
- ParticleSystem **ballBounceParticles**
- ParticleSystem **goalScoreParticles**
- GameObject [] **enableOnGoal**
- AudioClip **ballBounceAudio**
- AudioClip **headBallCollisionAudio**
- AudioClip **ballKickAudio**
- AudioClip **tackleAudio**
- AudioClip **crowdAudio**
- AudioClip **tackledAudio**
- AudioClip **interceptionAudio**

5.4.1 Detailed Description

This class is designed to be accessed via `Simulation.GetModel`. It contains scene specific configuration items which are not directly related to gameplay. [Gameplay](#) specific data is stored in the `BallGameModel` class.

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

- Assets/TeamBallGameTemplate/Scripts/Mechanics/BallGameConfig.cs

5.5 TeamBallGame.Model.BallGameModel Class Reference

This is a general model containing all data required for simulation of a ball game. It needs to be initialised at the start of a game by the GameController. It also contains methods for getting useful information from the model.

Public Member Functions

- bool [IsClosest](#) ([Player](#) player)
Query if the player is one of the closest two players to the ball.
- void [AddToContest](#) ([Player](#) player)
Add a player to the active contest, which will be resolved later by an external system.
- [Player](#) [GetClosestPlayer](#) ([Player](#)[] [players](#), [Vector3](#) position)

Public Attributes

- [Ball](#) [ball](#)
The ball.
- [Team](#) [homeTeam](#)
- [Player](#) [homePlayerPrefab](#)
- float [maxSpeed](#) = 5
- float [backwardsSpeedPenalty](#) = 0.7f
- float [maxKickDistance](#) = 30
- float [maxTurnSpeed](#) = 360
- [Player](#) [playerInPossession](#)
The player currently in possession of the ball.
- [Player](#) [] [players](#)
The list of all players from both teams.
- List< [Player](#) > [activeContest](#) = new List<[Player](#)>()
The list of players involved in an active contest.
- bool [IsBallUnderUserControl](#) => [playerInPossession](#)?.team.teamType == TeamType.Home
Is the ball currently possessed by a player from the home team?
- int [homeScore](#) = 0
- int [awayScore](#) = 0
- [Goal](#) [homeGoal](#)
Goal components for each team.
- float [durationFromGoalToBallup](#) = 4
- float [timeBetweenTackles](#) = 2
- float [tackleRecoveryTime](#) = 1.3f

5.5.1 Detailed Description

This is a general model containing all data required for simulation of a ball game. It needs to be initialised at the start of a game by the GameController. It also contains methods for getting useful information from the model.

5.5.2 Member Function Documentation

5.5.2.1 AddToContest()

```
void TeamBallGame.Model.BallGameModel.AddToContest (
    Player player )
```

Add a player to the active contest, which will be resolved later by an external system.

Parameters

<i>player</i>	
---------------	--

5.5.2.2 IsClosest()

```
bool TeamBallGame.Model.BallGameModel.IsClosest (
    Player player )
```

Query if the player is one of the closest two players to the ball.

Parameters

<i>player</i>	
---------------	--

Returns

5.5.3 Member Data Documentation

5.5.3.1 activeContest

```
List<Player> TeamBallGame.Model.BallGameModel.activeContest = new List<Player>()
```

The list of players involved in an active contest.

5.5.3.2 ball

```
Ball TeamBallGame.Model.BallGameModel.ball
```

The ball.

5.5.3.3 homeGoal

```
Goal TeamBallGame.Model.BallGameModel.homeGoal
```

Goal components for each team.

5.5.3.4 IsBallUnderUserControl

```
bool TeamBallGame.Model.BallGameModel.IsBallUnderUserControl => playerInPossession?.team.←
teamType == TeamType.Home
```

Is the ball currently possessed by a player from the home team?

5.5.3.5 playerInPossession

```
Player TeamBallGame.Model.BallGameModel.playerInPossession
```

The player currently in possession of the ball.

5.5.3.6 players

```
Player [ ] TeamBallGame.Model.BallGameModel.players
```

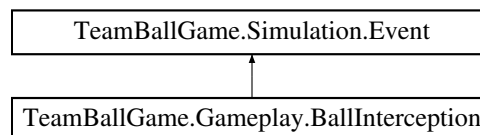
The list of all players from both teams.

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

- Assets/TeamBallGameTemplate/Scripts/Model/BallGameModel.cs

5.6 TeamBallGame.Gameplay.BallInterception Class Reference

Inheritance diagram for TeamBallGame.Gameplay.BallInterception:



Public Member Functions

- override void **Execute** ()

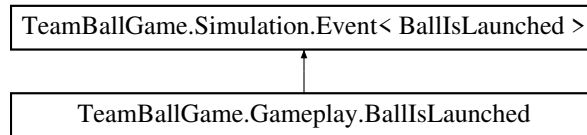
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Gameplay/BallInterception.cs

5.7 TeamBallGame.Gameplay.BallsLaunched Class Reference

This event is fired when the ball has been successfully launched by a player.

Inheritance diagram for TeamBallGame.Gameplay.BallsLaunched:



Public Member Functions

- override void **Execute** ()

Public Attributes

- float **flightDuration**
- [Player](#) **playerThatLaunchedBall**
- Vector3 **targetPosition**
- Vector3 **velocity**

5.7.1 Detailed Description

This event is fired when the ball has been successfully launched by a player.

Template Parameters

BallsLaunched	
-------------------------------	--

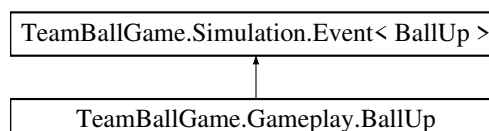
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Gameplay/BallsLaunched.cs

5.8 TeamBallGame.Gameplay.BallUp Class Reference

This event occurs when the ball is stationary, or thrown in / bounced by an umpire and the two nearest players will compete for possession.

Inheritance diagram for TeamBallGame.Gameplay.BallUp:



Public Member Functions

- override void **Execute** ()

Public Attributes

- Vector3 **position**

5.8.1 Detailed Description

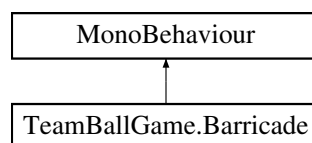
This event occurs when the ball is stationary, or thrown in / bounced by an umpire and the two nearest players will compete for possession.

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

- Assets/TeamBallGameTemplate/Scripts/Gameplay/BallUp.cs

5.9 TeamBallGame.Barricade Class Reference

Inheritance diagram for TeamBallGame.Barricade:



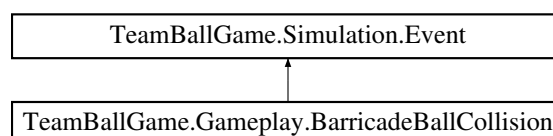
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Model/Barricade.cs

5.10 TeamBallGame.Gameplay.BarricadeBallCollision Class Reference

This event occurs when the ball hits a barricade. Logic for scheduling out-of-bounds events would happen here.

Inheritance diagram for TeamBallGame.Gameplay.BarricadeBallCollision:



Public Member Functions

- override void **Execute** ()

Public Attributes

- Collision **collision**
- [Barricade](#) **barricade**

5.10.1 Detailed Description

This event occurs when the ball hits a barricade. Logic for scheduling out-of-bounds events would happen here.

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

- Assets/TeamBallGameTemplate/Scripts/Gameplay/BarricadeBallCollision.cs

5.11 TeamBallGame.ComponentPool< T > Class Template Reference

Static Public Member Functions

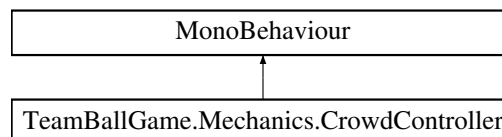
- static void **Prewarm** (T prefab, int count)
- static T **Take** (T prefab)
- static void **Return** (T instance, float when)
- static void **Return** (T instance)

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

- Assets/TeamBallGameTemplate/Scripts/Simulation/ComponentPool.cs

5.12 TeamBallGame.Mechanics.CrowdController Class Reference

Inheritance diagram for TeamBallGame.Mechanics.CrowdController:



Public Attributes

- AnimationCurve **intensity**
- Transform **homeGoal**

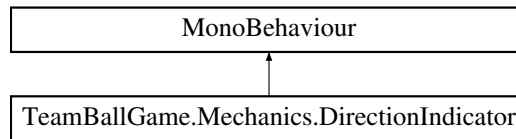
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Mechanics/CrowdController.cs

5.13 TeamBallGame.Mechanics.DirectionIndicator Class Reference

Sets the location of a transform and looks at another transform. Used for providing visual feedback about directions to the user.

Inheritance diagram for TeamBallGame.Mechanics.DirectionIndicator:



Public Attributes

- Transform **source**
- Vector3 **offset**

5.13.1 Detailed Description

Sets the location of a transform and looks at another transform. Used for providing visual feedback about directions to the user.

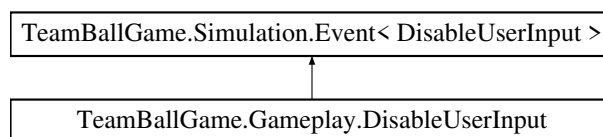
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Mechanics/DirectionIndicator.cs

5.14 TeamBallGame.Gameplay.DisableUserInput Class Reference

This event is fired when user input has been disabled for gameplay reasons (eg referee suspends play, goal is scored)

Inheritance diagram for TeamBallGame.Gameplay.DisableUserInput:



Public Member Functions

- override void **Execute** ()

5.14.1 Detailed Description

This event is fired when user input has been disabled for gameplay reasons (eg referee suspends play, goal is scored)

Template Parameters

[DisableUserInput](#)

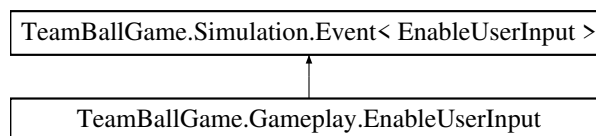
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Gameplay/DisableUserInput.cs

5.15 TeamBallGame.Gameplay.EnableUserInput Class Reference

This event is fired when user input has been reenabled for gameplay reasons (eg Ball up contest started)

Inheritance diagram for TeamBallGame.Gameplay.EnableUserInput:



Public Member Functions

- override void **Execute** ()

5.15.1 Detailed Description

This event is fired when user input has been reenabled for gameplay reasons (eg Ball up contest started)

Template Parameters

[DisableUserInput](#)

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

- Assets/TeamBallGameTemplate/Scripts/Gameplay/EnableUserInput.cs

5.16 TeamBallGame.Simulation.Event Class Reference

Inheritance diagram for TeamBallGame.Simulation.Event:

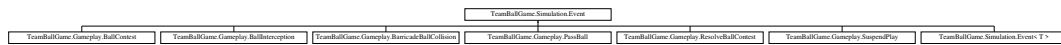


The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Simulation/Simulation.Event.cs

5.17 TeamBallGame.Simulation.Event Class Reference

Inheritance diagram for TeamBallGame.Simulation.Event:



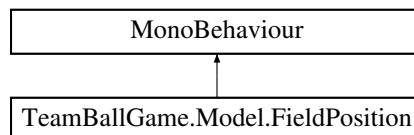
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Simulation/Simulation.Event.cs

5.18 TeamBallGame.Model.FieldPosition Class Reference

A named position on a playing field, for example "Goal Keeper".

Inheritance diagram for TeamBallGame.Model.FieldPosition:



Public Attributes

- float **radius** = 2

5.18.1 Detailed Description

A named position on a playing field, for example "Goal Keeper".

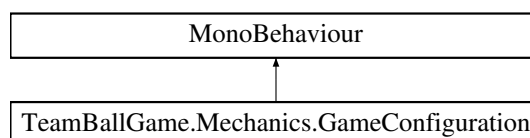
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Model/FieldPosition.cs

5.19 TeamBallGame.Mechanics.GameConfiguration Class Reference

This behaviour holds configuration information for gameplay, and stores it in the Simulation BallGameModel instance when initialised.

Inheritance diagram for TeamBallGame.Mechanics.GameConfiguration:



Public Attributes

- [BallGameModel](#) **model**
- [BallGameConfig](#) **config**

5.19.1 Detailed Description

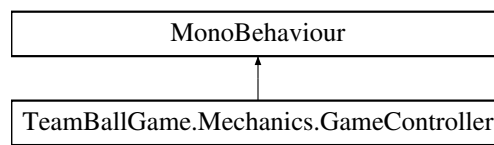
This behaviour holds configuration information for gameplay, and stores it in the Simulation BallGameModel instance when initialised.

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

- Assets/TeamBallGameTemplate/Scripts/Mechanics/GameConfiguration.cs

5.20 TeamBallGame.Mechanics.GameController Class Reference

Inheritance diagram for TeamBallGame.Mechanics.GameController:



Public Attributes

- [UserInput](#) **homeUserInput**

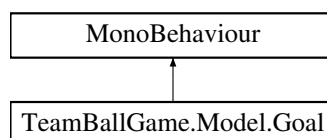
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Mechanics/GameController.cs

5.21 TeamBallGame.Model.Goal Class Reference

This behaviour is attached to goal trigger volumes, and will fire a Score event when a [Ball](#) component enters the trigger.

Inheritance diagram for TeamBallGame.Model.Goal:



Public Attributes

- TeamType **teamType**
- int **scoreValue** = 1

5.21.1 Detailed Description

This behaviour is attached to goal trigger volumes, and will fire a Score event when a [Ball](#) component enters the trigger.

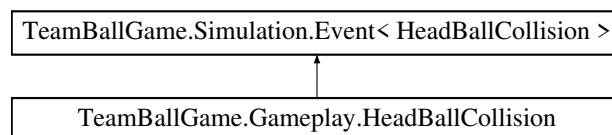
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Model/Goal.cs

5.22 TeamBallGame.Gameplay.HeadBallCollision Class Reference

This event is fired when the head collider of a player makes contact with the ball.

Inheritance diagram for TeamBallGame.Gameplay.HeadBallCollision:



Public Member Functions

- override void **Execute** ()

Public Attributes

- [Player](#) **player**
- Vector3 **deltaToBall**

5.22.1 Detailed Description

This event is fired when the head collider of a player makes contact with the ball.

Template Parameters

HeadBallCollision	
-----------------------------------	--

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

- Assets/TeamBallGameTemplate/Scripts/Gameplay/HeadBallCollision.cs

5.23 TeamBallGame.HeapQueue< T > Class Template Reference

Public Member Functions

- void **Clear** ()
- bool **Contains** (T item)
- void **Remove** (T item)
- T **Peek** ()
- void **Push** (T item)
- T **Pop** ()

Properties

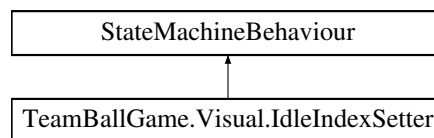
- int **Count** [get]
- bool **IsEmpty** [get]
- T **First** [get]

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

- Assets/TeamBallGameTemplate/Scripts/Simulation/HeapQueue.cs

5.24 TeamBallGame.Visual.IdleIndexSetter Class Reference

Inheritance diagram for TeamBallGame.Visual.IdleIndexSetter:



Public Member Functions

- override void **OnStateEnter** (Animator animator, AnimatorStateInfo stateInfo, int layerIndex)

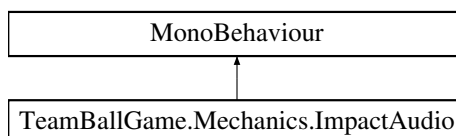
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Visual/IdleIndexSetter.cs

5.25 TeamBallGame.Mechanics.ImpactAudio Class Reference

Plays impact audio, taking into account distance from audio listener and the speed of sound in air at 20C.

Inheritance diagram for TeamBallGame.Mechanics.ImpactAudio:



Public Member Functions

- void **Play** (float magnitude, AudioClip clip)
- void **Play** (Collision collision, AudioClip clip)

Public Attributes

- float **estimatedMaxImpactVelocity** = 16
- new AudioSource **audio**

5.25.1 Detailed Description

Plays impact audio, taking into account distance from audio listener and the speed of sound in air at 20C.

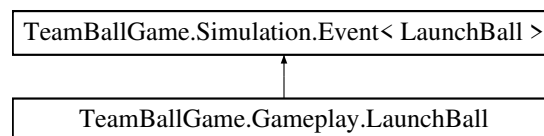
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Mechanics/ImpactAudio.cs

5.26 TeamBallGame.Gameplay.LaunchBall Class Reference

This event occurs when a player launches the ball. It is usually followed by a [BallsLaunched](#) event.

Inheritance diagram for TeamBallGame.Gameplay.LaunchBall:



Public Member Functions

- override void **Execute** ()
- void **CalculateVelocity** (Vector3 target, out Vector3 velocity, out float duration)

Public Attributes

- Vector3 **target**
- [Player](#) **player**

5.26.1 Detailed Description

This event occurs when a player launches the ball. It is usually followed by a [BallsLaunched](#) event.

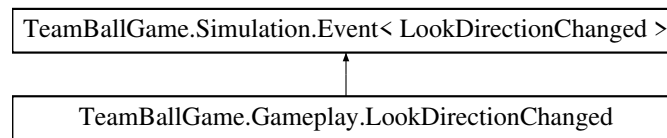
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Gameplay/LaunchBall.cs

5.27 TeamBallGame.Gameplay.LookDirectionChanged Class Reference

This event is fired when the user input has changed the look direction of the active player.

Inheritance diagram for TeamBallGame.Gameplay.LookDirectionChanged:



Public Member Functions

- override void **Execute** ()

Public Attributes

- [Player](#) **player**
- `Vector3` **direction**

5.27.1 Detailed Description

This event is fired when the user input has changed the look direction of the active player.

Template Parameters

LookDirectionChanged	
--------------------------------------	--

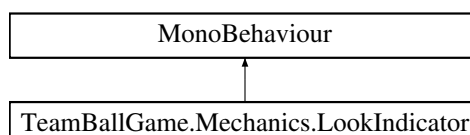
The documentation for this class was generated from the following file:

- `Assets/TeamBallGameTemplate/Scripts/Gameplay/LookDirectionChanged.cs`

5.28 TeamBallGame.Mechanics.LookIndicator Class Reference

A specialised Directional Indicator used to indicate the current pass direction, and if the pass direction will intercept a team mate.

Inheritance diagram for TeamBallGame.Mechanics.LookIndicator:



Public Member Functions

- void **SetPassIndicator** (bool passable)
- void **SetPositions** (Vector3 start, Vector3 end)

Public Attributes

- LineRenderer **lineRenderer**
- Material **normalMaterial**
- Material **canPassMaterial**

5.28.1 Detailed Description

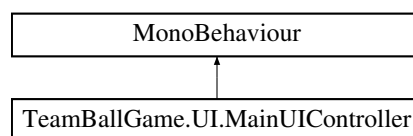
A specialised Directional Indicator used to indicate the current pass direction, and if the pass direction will intercept a team mate.

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

- Assets/TeamBallGameTemplate/Scripts/Mechanics/LookIndicator.cs

5.29 TeamBallGame.UI.MainUIController Class Reference

Inheritance diagram for TeamBallGame.UI.MainUIController:



Public Member Functions

- void **SetActivePanel** (int index)

Public Attributes

- GameObject [] **panels**

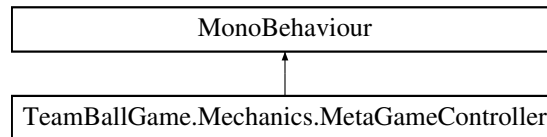
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/UI/MainUIController.cs

5.30 TeamBallGame.Mechanics.MetaGameController Class Reference

The [MetaGameController](#) is responsible for switching control between the high level contexts of the application, eg the Main Menu and [Gameplay](#) systems.

Inheritance diagram for TeamBallGame.Mechanics.MetaGameController:



Public Member Functions

- void **ToggleMainMenu** (bool show)

Public Attributes

- [MainUIController](#) **mainMenu**
- Canvas [] **gamePlayCanvasii**
- [GameController](#) **gameController**

5.30.1 Detailed Description

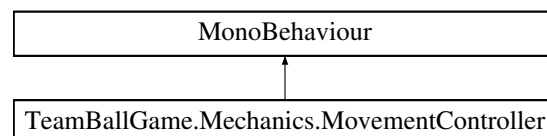
The [MetaGameController](#) is responsible for switching control between the high level contexts of the application, eg the Main Menu and [Gameplay](#) systems.

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

- Assets/TeamBallGameTemplate/Scripts/Mechanics/MetaGameController.cs

5.31 TeamBallGame.Mechanics.MovementController Class Reference

Inheritance diagram for TeamBallGame.Mechanics.MovementController:



Public Member Functions

- void **SetMovement** (bool enabled)
- void **To** (Vector3 position)
Set the destinate position for the controller.
- void **LookAt** (Vector3 position)
Set the desired look at position for the controller.
- void **LookDirection** (Vector3 direction)
Set the desired look direction for the controller.
- void **BumpTowards** (Vector3 position, float duration=1)
Momentarily push the controller in a direction for a specified period. Overrides any destination during this time.

Public Attributes

- float **maxSpeed** = 5
- float **maxTurnSpeed** = 360
- float **backwardsSpeedPenalty** = 0.4f

5.31.1 Member Function Documentation

5.31.1.1 BumpTowards()

```
void TeamBallGame.Mechanics.MovementController.BumpTowards (
    Vector3 position,
    float duration = 1 )
```

Momentarily push the controller in a direction for a specified period. Overrides any destination during this time.

Parameters

<i>position</i>	
<i>duration</i>	

5.31.1.2 LookAt()

```
void TeamBallGame.Mechanics.MovementController.LookAt (
    Vector3 position )
```

Set the desired look at position for the controller.

Parameters

<i>position</i>	
-----------------	--

5.31.1.3 LookDirection()

```
void TeamBallGame.Mechanics.MovementController.LookDirection (
    Vector3 direction )
```

Set the desired look direction for the controller.

Parameters

<i>position</i>	
-----------------	--

5.31.1.4 To()

```
void TeamBallGame.Mechanics.MovementController.To (
    Vector3 position )
```

Set the destinate position for the controller.

Parameters

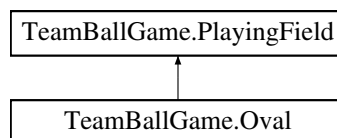
<i>position</i>	
-----------------	--

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

- Assets/TeamBallGameTemplate/Scripts/Mechanics/MovementController.cs

5.32 TeamBallGame.Oval Class Reference

Inheritance diagram for TeamBallGame.Oval:



Public Member Functions

- override bool **Contains** (Vector3 position)

Public Attributes

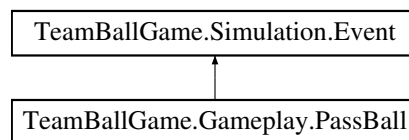
- Vector3 **size**

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

- Assets/TeamBallGameTemplate/Scripts/Model/Oval.cs

5.33 TeamBallGame.Gameplay.PassBall Class Reference

Inheritance diagram for TeamBallGame.Gameplay.PassBall:



Public Member Functions

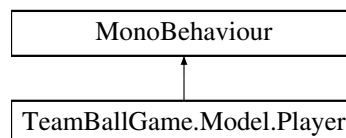
- override void **Execute** ()

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

- Assets/TeamBallGameTemplate/Scripts/Gameplay/PassBall.cs

5.34 TeamBallGame.Model.Player Class Reference

Inheritance diagram for TeamBallGame.Model.Player:



Public Types

- enum **State** {
 UserControl, **ReturnToPosition**, **AIControl**, **Waiting**,
 Tackled }

Public Member Functions

- void **OnTackle** ()
- void **OnPlayerJump** ()
- void **OnPrepareToLaunchBall** (Vector3 target)
- void **OnSuccessfulInterception** ()
- void **OnBallWillLandNearMe** (Vector3 targetPosition)
- void **OnLaunchBall** (Vector3 targetPosition)
- void **OnOtherPlayerLaunchedBall** ([Player](#) playerThatLaunchedBall, Vector3 targetPosition)
- void **OnBallBounceNearMe** (Vector3 position)
- void **OnUserInput** (Vector3 moveDirection, Vector3 lookDirection)
- void **SetMovement** (bool enabled)

Public Attributes

- Vector3 **DeltaToBall** => ballGame.ball.transform.position - transform.position
- Vector3 **BallPosition** => transform.TransformPoint(possessionOffset)
- Vector3 **ReticlePosition** => transform.TransformPoint(reticleOffset)
- Vector3 **HeadPosition** => transform.TransformPoint(headOffset)
- bool **IsBallOwner** => ballGame.playerInPossession == this
- bool **IsHomeTeam** => team.teamType == TeamType.Home
- bool **IsAI** => state == State.AIControl
- Animator **animator**
- SpriteRenderer **icon**
- Vector3 **possessionOffset** = new Vector3(0, -1, -1)
- Vector3 **reticleOffset** = new Vector3(0, -1, 0)
- Vector3 **headOffset** = new Vector3(0, 1, 0)
- float **headSize** = 1

Properties

- Vector3 **DeltaToGoal** [get]

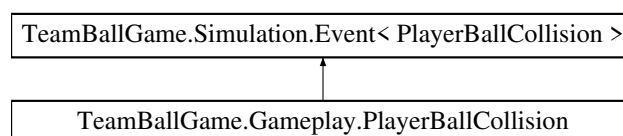
The documentation for this class was generated from the following files:

- Assets/TeamBallGameTemplate/Scripts/Model/Player.AI.cs
- Assets/TeamBallGameTemplate/Scripts/Model/Player.API.cs
- Assets/TeamBallGameTemplate/Scripts/Model/Player.cs

5.35 TeamBallGame.Gameplay.PlayerBallCollision Class Reference

This event is fired when a player collides with the ball. Generally, this would start a [BallContest](#).

Inheritance diagram for TeamBallGame.Gameplay.PlayerBallCollision:



Public Member Functions

- override void **Execute** ()

Public Attributes

- [Player](#) **player**
- Collision **collision**

5.35.1 Detailed Description

This event is fired when a player collides with the ball. Generally, this would start a [BallContest](#).

Template Parameters

PlayerBallCollision	
-------------------------------------	--

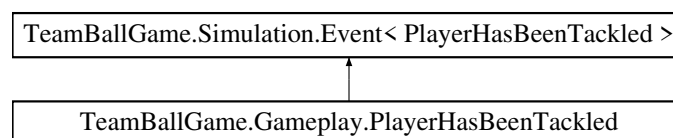
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Gameplay/PlayerBallCollision.cs

5.36 TeamBallGame.Gameplay.PlayerHasBeenTackled Class Reference

This event is fired when a player has been tackled.

Inheritance diagram for TeamBallGame.Gameplay.PlayerHasBeenTackled:



Public Member Functions

- override void **Execute** ()

Public Attributes

- [Player](#) **player**
- [Player](#) **tackler**
- Vector3 **direction**
- float **tacklePower** = 5

5.36.1 Detailed Description

This event is fired when a player has been tackled.

Template Parameters

PlayerHasBeenTackled	
--------------------------------------	--

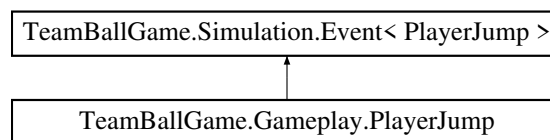
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Gameplay/PlayerHasBeenTackled.cs

5.37 TeamBallGame.Gameplay.PlayerJump Class Reference

This event is fired when the jump input is activated.

Inheritance diagram for TeamBallGame.Gameplay.PlayerJump:



Public Member Functions

- override void **Execute** ()

Public Attributes

- [Player](#) **player**

5.37.1 Detailed Description

This event is fired when the jump input is activated.

Template Parameters

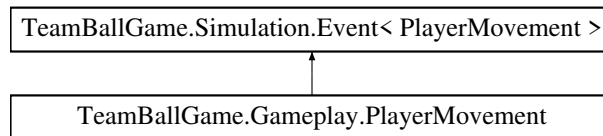
PlayerJump	
----------------------------	--

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

- Assets/TeamBallGameTemplate/Scripts/Gameplay/PlayerJump.cs

5.38 TeamBallGame.Gameplay.PlayerMovement Class Reference

Inheritance diagram for TeamBallGame.Gameplay.PlayerMovement:



Public Member Functions

- override void **Execute** ()

Public Attributes

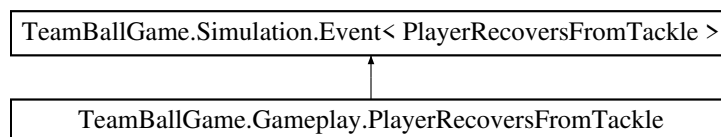
- [Player](#) **player**

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

- `Assets/TeamBallGameTemplate/Scripts/Gameplay/PlayerMovement.cs`

5.39 TeamBallGame.Gameplay.PlayerRecoversFromTackle Class Reference

Inheritance diagram for `TeamBallGame.Gameplay.PlayerRecoversFromTackle`:



Public Member Functions

- override void **Execute** ()

Public Attributes

- [Player](#) **player**

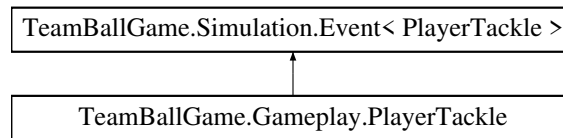
The documentation for this class was generated from the following file:

- `Assets/TeamBallGameTemplate/Scripts/Gameplay/PlayerRecoversFromTackle.cs`

5.40 TeamBallGame.Gameplay.PlayerTackle Class Reference

This event is triggered when the tackle input is received.

Inheritance diagram for TeamBallGame.Gameplay.PlayerTackle:



Public Member Functions

- override void **Execute** ()

Public Attributes

- [Player](#) **player**

5.40.1 Detailed Description

This event is triggered when the tackle input is received.

Template Parameters

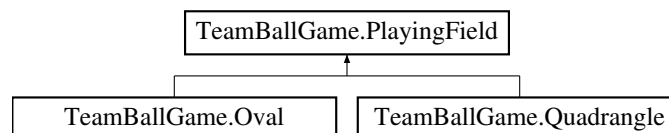
PlayerTackle	
------------------------------	--

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

- Assets/TeamBallGameTemplate/Scripts/Gameplay/PlayerTackle.cs

5.41 TeamBallGame.PlayingField Class Reference

Inheritance diagram for TeamBallGame.PlayingField:



Public Member Functions

- abstract bool **Contains** (Vector3 position)

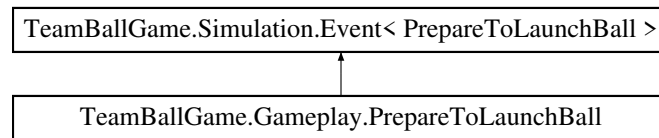
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Model/PlayingField.cs

5.42 TeamBallGame.Gameplay.PrepareToLaunchBall Class Reference

This event is fired when a player is launching the ball at a target position.

Inheritance diagram for TeamBallGame.Gameplay.PrepareToLaunchBall:



Public Member Functions

- override void **Execute** ()

Public Attributes

- Vector3 **target**
- float **delay** = 0.25f
- [Player](#) **player**

5.42.1 Detailed Description

This event is fired when a player is launching the ball at a target position.

Template Parameters

PrepareToLaunchBall	
-------------------------------------	--

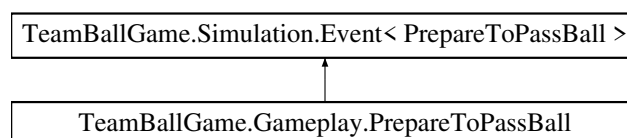
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Gameplay/PrepareToLaunchBall.cs

5.43 TeamBallGame.Gameplay.PrepareToPassBall Class Reference

This event is fired when a player is passing the ball.

Inheritance diagram for TeamBallGame.Gameplay.PrepareToPassBall:



Public Member Functions

- override void **Execute** ()

Public Attributes

- Vector3 **target**
- float **delay** = 0.125f
- [Player](#) **player**

5.43.1 Detailed Description

This event is fired when a player is passing the ball.

Template Parameters

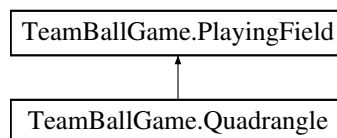
PrepareToPassBall	
-----------------------------------	--

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

- Assets/TeamBallGameTemplate/Scripts/Gameplay/PrepareToPassBall.cs

5.44 TeamBallGame.Quadrangle Class Reference

Inheritance diagram for TeamBallGame.Quadrangle:



Public Member Functions

- override bool **Contains** (Vector3 position)

Public Attributes

- Vector3 **size**

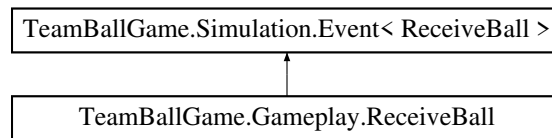
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Model/Quadrangle.cs

5.45 TeamBallGame.Gameplay.ReceiveBall Class Reference

This event is fired when a player receives control of the ball.

Inheritance diagram for TeamBallGame.Gameplay.ReceiveBall:



Public Member Functions

- override void **Execute** ()

Public Attributes

- **Player** player

5.45.1 Detailed Description

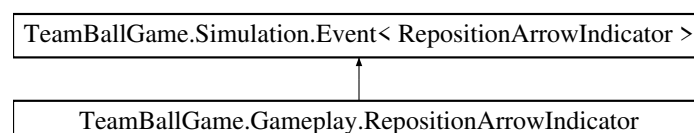
This event is fired when a player receives control of the ball.

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

- Assets/TeamBallGameTemplate/Scripts/Gameplay/ReceiveBall.cs

5.46 TeamBallGame.Gameplay.RepositionArrowIndicator Class Reference

Inheritance diagram for TeamBallGame.Gameplay.RepositionArrowIndicator:



Public Member Functions

- override void **Execute** ()

Public Attributes

- Vector3 **position**
- Vector3 **direction**

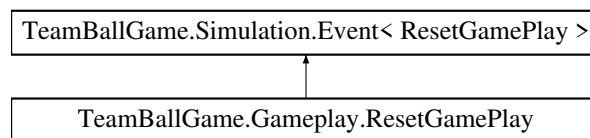
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Gameplay/RepositionArrowIndicator.cs

5.47 TeamBallGame.Gameplay.ResetGamePlay Class Reference

Reset game play to a ball up or kick off state. Eg, move ball to center and all players to field positions.

Inheritance diagram for TeamBallGame.Gameplay.ResetGamePlay:



Public Member Functions

- override void **Execute** ()

5.47.1 Detailed Description

Reset game play to a ball up or kick off state. Eg, move ball to center and all players to field positions.

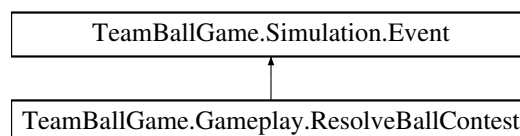
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Gameplay/ResetGamePlay.cs

5.48 TeamBallGame.Gameplay.ResolveBallContest Class Reference

This event is continuously scheduled so that pending ball contests can be resolved and ball possession can be changed accordingly.

Inheritance diagram for TeamBallGame.Gameplay.ResolveBallContest:



Public Member Functions

- override void **Execute** ()

5.48.1 Detailed Description

This event is continuously scheduled so that pending ball contests can be resolved and ball possession can be changed accordingly.

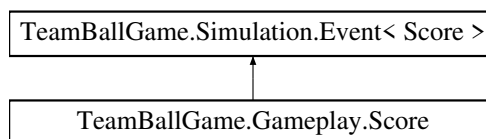
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Gameplay/ResolveBallContest.cs

5.49 TeamBallGame.Gameplay.Score Class Reference

This event is fired when the ball enters a Goal trigger.

Inheritance diagram for TeamBallGame.Gameplay.Score:



Public Member Functions

- override void **Execute** ()

Public Attributes

- TeamType **teamType**
- [Goal](#) **goal**

5.49.1 Detailed Description

This event is fired when the ball enters a Goal trigger.

Template Parameters

Score	
-----------------------	--

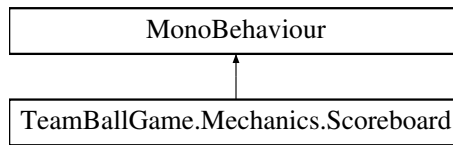
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Gameplay/Score.cs

5.50 TeamBallGame.Mechanics.Scoreboard Class Reference

[Scoreboard](#) waits for ScoreEvents and updates text widgets with the new score values.

Inheritance diagram for TeamBallGame.Mechanics.Scoreboard:



Public Attributes

- `TMPPro.TMP_Text` **home**

5.50.1 Detailed Description

[Scoreboard](#) waits for ScoreEvents and updates text widgets with the new score values.

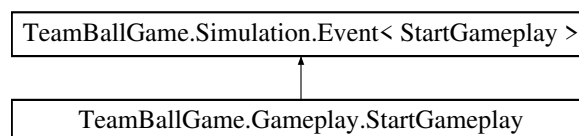
The documentation for this class was generated from the following file:

- `Assets/TeamBallGameTemplate/Scripts/Mechanics/Scoreboard.cs`

5.51 TeamBallGame.Gameplay.StartGameplay Class Reference

This event is triggered when gameplay should start, eg at the start of a game period or after the ball is returned to the center.

Inheritance diagram for TeamBallGame.Gameplay.StartGameplay:



Public Member Functions

- override void **Execute** ()

5.51.1 Detailed Description

This event is triggered when gameplay should start, eg at the start of a game period or after the ball is returned to the center.

Template Parameters

StartGameplay

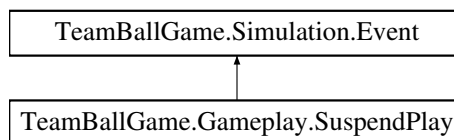
The documentation for this class was generated from the following file:

- Assets/TeamBallGameTemplate/Scripts/Gameplay/StartGameplay.cs

5.52 TeamBallGame.Gameplay.SuspendPlay Class Reference

Suspend play occurs when players must stop and allow a penalty, ball up or some other event to occur before resuming.

Inheritance diagram for TeamBallGame.Gameplay.SuspendPlay:



Public Member Functions

- override void **Execute** ()

5.52.1 Detailed Description

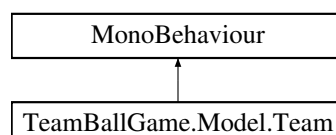
Suspend play occurs when players must stop and allow a penalty, ball up or some other event to occur before resuming.

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

- Assets/TeamBallGameTemplate/Scripts/Gameplay/SuspendPlay.cs

5.53 TeamBallGame.Model.Team Class Reference

Inheritance diagram for TeamBallGame.Model.Team:



Public Member Functions

- void **InstantiatePlayers** ()

Public Attributes

- TeamType **teamType**
- LayerMask **layer**
- Goal **goal**
- Team **opposingTeam**
- FieldPosition [] **positions**
- Player [] **players**
- Material **teamMaterial**

Properties

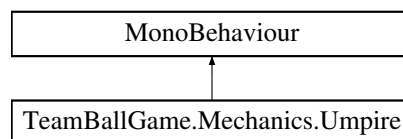
- [UserInput](#) **UserInput** [get, set]

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

- Assets/TeamBallGameTemplate/Scripts/Model/Team.cs

5.54 TeamBallGame.Mechanics.Umpire Class Reference

Inheritance diagram for TeamBallGame.Mechanics.Umpire:



Properties

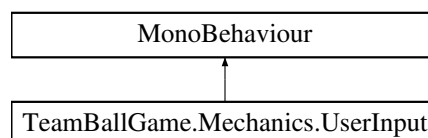
- [MovementController](#) **Move** [get]

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

- Assets/TeamBallGameTemplate/Scripts/Mechanics/Umpire.cs

5.55 TeamBallGame.Mechanics.UserInput Class Reference

Inheritance diagram for TeamBallGame.Mechanics.UserInput:



Public Member Functions

- void **UpdateActivePlayer** ()

Public Attributes

- Camera **mainCamera**

Properties

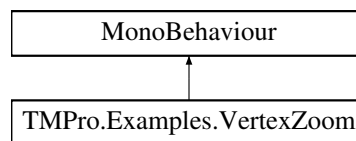
- [Player](#) **ActivePlayer** [get, set]

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

- Assets/TeamBallGameTemplate/Scripts/Mechanics/UserInput.cs

5.56 TPro.Examples.VertexZoom Class Reference

Inheritance diagram for TPro.Examples.VertexZoom:



Public Attributes

- float **scaleMin** = 0.5f
- float **scaleMax** = 1.0f
- float **scaleSpeed** = 0.1f

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

- Assets/TeamBallGameTemplate/Scripts/Visual/VertexZoom.cs