

Round Spawn System

1.0

Generated by Doxygen 1.8.10

Tue Sep 1 2015 13:39:13

Contents

1	Namespace Index	1
1.1	Packages	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Namespace Documentation	9
4.1	RoundManager Namespace Reference	9
4.2	RoundManager.Events Namespace Reference	11
5	Class Documentation	13
5.1	RoundManager.AudioOnRoundEnd Class Reference	13
5.1.1	Detailed Description	13
5.2	RoundManager.AudioOnRoundStart Class Reference	13
5.2.1	Detailed Description	14
5.3	RoundManager.BackgroundAudio Class Reference	14
5.3.1	Detailed Description	14
5.4	RoundManager.BackgroundAudioChangeOnNewRound Class Reference	14
5.4.1	Detailed Description	15
5.4.2	Member Data Documentation	15
5.4.2.1	Audio	15
5.5	RoundManager.BossCountdownBegun Class Reference	15
5.5.1	Detailed Description	15
5.5.2	Member Function Documentation	15
5.5.2.1	OnBossCountdownBegun(BossCountdownBegunEvent e)	15
5.6	RoundManager.Events.BossCountdownBegunEvent Class Reference	16
5.6.1	Detailed Description	16
5.6.2	Property Documentation	16
5.6.2.1	CurrentRound	16
5.7	RoundManager.BossCountdownText Class Reference	16

5.7.1	Detailed Description	17
5.8	RoundManager.BossKilled Class Reference	17
5.8.1	Detailed Description	17
5.8.2	Member Function Documentation	17
5.8.2.1	OnBossKilled(BossKilledEvent e)	17
5.9	RoundManager.Events.BossKilledEvent Class Reference	18
5.9.1	Detailed Description	18
5.9.2	Property Documentation	18
5.9.2.1	CurrentRound	18
5.10	RoundManager.BossSpawned Class Reference	18
5.10.1	Detailed Description	19
5.10.2	Member Function Documentation	19
5.10.2.1	OnBossSpawned(BossSpawnedEvent e)	19
5.11	RoundManager.Events.BossSpawnedEvent Class Reference	19
5.11.1	Detailed Description	20
5.11.2	Property Documentation	20
5.11.2.1	Boss	20
5.12	RoundManager.BossSpawnRequest Class Reference	20
5.12.1	Detailed Description	20
5.13	RoundManager.Events.BossSpawnRequestEvent Class Reference	20
5.13.1	Detailed Description	21
5.14	RoundManager.CheckpointEnd Class Reference	21
5.14.1	Detailed Description	21
5.14.2	Member Function Documentation	22
5.14.2.1	OnCheckpointEnd(CheckpointEndEvent e)	22
5.15	RoundManager.Events.CheckpointEndEvent Class Reference	23
5.15.1	Detailed Description	23
5.15.2	Property Documentation	23
5.15.2.1	CurrentRound	23
5.15.2.2	HasNextCheckpoint	23
5.16	RoundManager.CheckpointStart Class Reference	24
5.16.1	Detailed Description	24
5.16.2	Member Function Documentation	24
5.16.2.1	OnCheckpointStart(CheckpointStartEvent e)	24
5.17	RoundManager.Events.CheckpointStartEvent Class Reference	24
5.17.1	Detailed Description	25
5.17.2	Property Documentation	25
5.17.2.1	CurrentRound	25
5.18	RoundManager.DestroyCurrentEnemiesRequest Class Reference	25
5.18.1	Detailed Description	25

5.18.2	Member Function Documentation	25
5.18.2.1	OnDestroyCurrentEnemiesRequest(DestroyCurrentEnemiesRequestEvent e)	25
5.19	RoundManager.Events.DestroyCurrentEnemiesRequestEvent Class Reference	26
5.19.1	Detailed Description	26
5.19.2	Property Documentation	26
5.19.2.1	CurrentRound	26
5.20	RoundManager.Director Class Reference	26
5.20.1	Detailed Description	27
5.21	RoundManager.EnemyHealth Class Reference	27
5.21.1	Detailed Description	27
5.22	RoundManager.EnemySpawner Class Reference	28
5.22.1	Detailed Description	28
5.23	RoundManager.EnemySpawnRequest Class Reference	28
5.23.1	Detailed Description	28
5.24	RoundManager.Events.EnemySpawnRequestEvent Class Reference	29
5.24.1	Detailed Description	29
5.25	RoundManager.Environment Class Reference	29
5.25.1	Detailed Description	30
5.25.2	Member Function Documentation	30
5.25.2.1	GenerateFloor()	30
5.25.3	Member Data Documentation	30
5.25.3.1	EnemySpawnArea	30
5.26	RoundManager.FinishedRounds Class Reference	30
5.26.1	Detailed Description	31
5.26.2	Member Function Documentation	31
5.26.2.1	OnFinishedRounds(FinishedRoundsEvent e)	31
5.27	RoundManager.Events.FinishedRoundsEvent Class Reference	31
5.27.1	Detailed Description	31
5.27.2	Property Documentation	31
5.27.2.1	LastRound	31
5.28	RoundManager.FirstRound Class Reference	32
5.28.1	Detailed Description	32
5.28.2	Member Function Documentation	32
5.28.2.1	OnFirstRound(FirstRoundEvent e)	32
5.29	RoundManager.Events.FirstRoundEvent Class Reference	32
5.29.1	Detailed Description	33
5.29.2	Property Documentation	33
5.29.2.1	CurrentRound	33
5.30	RoundManager.LastRound Class Reference	33
5.30.1	Detailed Description	33

5.31 RoundManager.Events.LastRoundEvent Class Reference	33
5.31.1 Detailed Description	34
5.31.2 Property Documentation	34
5.31.2.1 CurrentRound	34
5.32 RoundManager.LerpOverTime Class Reference	34
5.32.1 Detailed Description	34
5.32.2 Property Documentation	35
5.32.2.1 Value	35
5.33 RoundManager.LightingChangeOnNewRound Class Reference	35
5.33.1 Detailed Description	35
5.33.2 Member Data Documentation	35
5.33.2.1 Lighting	35
5.34 RoundManager.ObjectSpawner Class Reference	35
5.34.1 Detailed Description	36
5.35 RoundManager.ObjectSpawnRequest Class Reference	36
5.35.1 Detailed Description	36
5.35.2 Member Function Documentation	36
5.35.2.1 OnObjectSpawnRequest(ObjectSpawnRequestEvent e)	36
5.36 RoundManager.Events.ObjectSpawnRequestEvent Class Reference	37
5.36.1 Detailed Description	37
5.36.2 Property Documentation	37
5.36.2.1 CurrentRound	37
5.36.2.2 ObjectPrefab	38
5.37 RoundManager.PreperationCountdownText Class Reference	38
5.37.1 Detailed Description	38
5.38 RoundManager.PreperationEnd Class Reference	38
5.38.1 Detailed Description	39
5.38.2 Member Function Documentation	39
5.38.2.1 OnPreperationEnd(PreperationEndEvent e)	39
5.39 RoundManager.Events.PreperationEndEvent Class Reference	39
5.39.1 Detailed Description	39
5.39.2 Property Documentation	40
5.39.2.1 CurrentRound	40
5.40 RoundManager.PreperationStart Class Reference	40
5.40.1 Detailed Description	40
5.40.2 Member Function Documentation	40
5.40.2.1 OnPreperationStart(PreperationStartEvent e)	40
5.41 RoundManager.Events.PreperationStartEvent Class Reference	40
5.41.1 Detailed Description	41
5.41.2 Property Documentation	41

5.41.2.1	CurrentRound	41
5.42	RoundManager.Round Class Reference	41
5.42.1	Detailed Description	43
5.42.2	Member Function Documentation	43
5.42.2.1	BossKilled()	43
5.42.2.2	Enter(RoundManager manager)	43
5.42.2.3	Execute()	44
5.42.2.4	Exit()	44
5.42.2.5	OnPreparationOver()	44
5.42.2.6	OnPreparationStart()	44
5.42.2.7	RegisterEnemyKilled()	44
5.42.2.8	RegisterEnemySpawned()	44
5.42.2.9	SpawnBoss()	44
5.42.2.10	TriggerRoundEnd()	44
5.42.3	Member Data Documentation	45
5.42.3.1	BossCountdown	45
5.42.3.2	Checkpoints	45
5.42.3.3	DestroyEnemiesOnRoundEnd	45
5.42.3.4	DestroyOtherEnemiesWhenBossCountdownBegins	45
5.42.3.5	DestroyOtherEnemiesWhenBossSpawns	45
5.42.3.6	DestroyPreviousRoundEnemiesOnRoundStart	45
5.42.3.7	HasPreperationTime	45
5.42.3.8	NumberOfObjectsToSpawn	45
5.42.3.9	OnlySpawnBossWhenAllEnemiesKilled	45
5.42.3.10	PreperationTime	45
5.42.3.11	PreperationTimeObjects	45
5.42.3.12	RoundBossPrefab	46
5.42.3.13	RoundHasBoss	46
5.42.3.14	RoundProgressType	46
5.42.3.15	SpawnObjectsDuringPreperaionTime	46
5.42.4	Property Documentation	46
5.42.4.1	CurrentCheckpoint	46
5.42.4.2	InPreperation	46
5.42.4.3	RoundTime	46
5.42.4.4	RoundTimeInMinutes	46
5.42.4.5	RoundTimeInMinutesSeconds	46
5.42.4.6	RoundTimeInSeconds	47
5.42.4.7	RoundTotalTime	47
5.43	RoundManager.RoundBoss Class Reference	47
5.43.1	Detailed Description	47

5.44 RoundManager.RoundCheckpoint Class Reference	47
5.44.1 Detailed Description	48
5.44.2 Member Function Documentation	48
5.44.2.1 Enter()	48
5.44.2.2 Execute()	48
5.44.3 Member Data Documentation	48
5.44.3.1 CheckpointTime	48
5.44.3.2 EnemySpawnChance	49
5.44.3.3 LimitEnemyCount	49
5.44.3.4 MaxEnemies	49
5.44.3.5 RoundEnemies	49
5.44.3.6 TimeBetweenEnemySpawns	49
5.44.4 Property Documentation	49
5.44.4.1 CheckpointTimeInMinutes	49
5.44.4.2 CheckpointTimeInMinutesSeconds	49
5.44.4.3 CheckpointTimeInSeconds	49
5.45 RoundManager.RoundCountdownAudio Class Reference	49
5.45.1 Detailed Description	50
5.46 RoundManager.RoundCountdownText Class Reference	50
5.46.1 Detailed Description	50
5.47 RoundManager.RoundEnd Class Reference	50
5.47.1 Detailed Description	51
5.47.2 Member Function Documentation	51
5.47.2.1 OnRoundEnd(RoundEndEvent e)	51
5.48 RoundManager.Events.RoundEndEvent Class Reference	51
5.48.1 Detailed Description	52
5.48.2 Property Documentation	52
5.48.2.1 CurrentRound	52
5.48.2.2 HasNextRound	52
5.49 RoundManager.RoundEnemy Class Reference	52
5.49.1 Detailed Description	52
5.50 RoundManager.Events.RoundEvent Class Reference	52
5.50.1 Detailed Description	53
5.51 RoundManager.Events.RoundEvents Class Reference	53
5.51.1 Detailed Description	54
5.51.2 Member Function Documentation	54
5.51.2.1 AddListener< T >(EventDelegate< T > del)	54
5.51.2.2 Raise(RoundEvent e)	54
5.51.2.3 RemoveListener< T >(EventDelegate< T > del)	54
5.52 RoundManager.RoundLighting Class Reference	55

5.52.1 Detailed Description	55
5.52.2 Member Data Documentation	55
5.52.2.1 RoundNumber	55
5.53 RoundManager.RoundManager Class Reference	55
5.53.1 Detailed Description	56
5.53.2 Member Function Documentation	56
5.53.2.1 Begin()	56
5.53.2.2 CoroutineCallback(Routine callBack)	56
5.53.2.3 LastRound()	56
5.53.2.4 LogRoundMessage(object message)	56
5.53.3 Member Data Documentation	57
5.53.3.1 Rounds	57
5.53.3.2 ShowDebugMessages	57
5.53.4 Property Documentation	57
5.53.4.1 CurrentRound	57
5.53.4.2 CurrentRoundIndex	57
5.53.4.3 Instance	57
5.54 RoundObjectSpawner Class Reference	57
5.54.1 Detailed Description	58
5.54.2 Member Data Documentation	58
5.54.2.1 Prefab	58
5.54.2.2 Weight	58
5.55 RoundManager.RoundsFinishedText Class Reference	58
5.55.1 Detailed Description	58
5.56 RoundManager.RoundStart Class Reference	58
5.56.1 Detailed Description	59
5.56.2 Member Function Documentation	59
5.56.2.1 OnRoundStart(RoundStartEvent e)	59
5.57 RoundManager.Events.RoundStartEvent Class Reference	59
5.57.1 Detailed Description	60
5.57.2 Property Documentation	60
5.57.2.1 Round	60
5.58 RoundManager.ShowTextOnRoundStart Class Reference	60
5.58.1 Detailed Description	60
5.59 RoundManager.Spawner Class Reference	60
5.59.1 Detailed Description	61
5.60 RoundManager.TriggerRoundEnd Class Reference	61
5.60.1 Detailed Description	61
5.61 RoundManager.Vector2i Struct Reference	61

Index	63
-----------------------	----

Chapter 1

Namespace Index

1.1 Packages

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

RoundManager	9
RoundManager.Events	11

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

RoundManager.BackgroundAudio	14
RoundManager.LerpOverTime	34
MonoBehaviour	
RoundManager.AudioOnRoundEnd	13
RoundManager.AudioOnRoundStart	13
RoundManager.BackgroundAudioChangeOnNewRound	14
RoundManager.BossCountdownBegun	15
RoundManager.BossCountdownText	16
RoundManager.BossKilled	17
RoundManager.BossSpawned	18
RoundManager.BossSpawnRequest	20
RoundManager.CheckpointEnd	21
RoundManager.CheckpointStart	24
RoundManager.DestroyCurrentEnemiesRequest	25
RoundManager.Director	26
RoundManager.EnemyHealth	27
RoundManager.EnemySpawnRequest	28
RoundManager.Environment	29
RoundManager.FinishedRounds	30
RoundManager.FirstRound	32
RoundManager.LastRound	33
RoundManager.LightingChangeOnNewRound	35
RoundManager.ObjectSpawnRequest	36
RoundManager.PreperationCountdownText	38
RoundManager.PreperationEnd	38
RoundManager.PreperationStart	40
RoundManager.RoundBoss	47
RoundManager.RoundCountdownAudio	49
RoundManager.RoundCountdownText	50
RoundManager.RoundEnd	50
RoundManager.RoundEnemy	52
RoundManager.RoundManager	55
RoundManager.RoundsFinishedText	58
RoundManager.RoundStart	58
RoundManager.ShowTextOnRoundStart	60
RoundManager.Spawner	60
RoundManager.EnemySpawner	28
RoundManager.ObjectSpawner	35

RoundManager.TriggerRoundEnd	61
RoundManager.Round	41
RoundManager.RoundCheckpoint	47
RoundManager.Events.RoundEvent	52
RoundManager.Events.BossCountdownBegunEvent	16
RoundManager.Events.BossKilledEvent	18
RoundManager.Events.BossSpawnedEvent	19
RoundManager.Events.CheckpointEndEvent	23
RoundManager.Events.CheckpointStartEvent	24
RoundManager.Events.DestroyCurrentEnemiesRequestEvent	26
RoundManager.Events.FinishedRoundsEvent	31
RoundManager.Events.FirstRoundEvent	32
RoundManager.Events.LastRoundEvent	33
RoundManager.Events.ObjectSpawnRequestEvent	37
RoundManager.Events.EnemySpawnRequestEvent	29
RoundManager.Events.BossSpawnRequestEvent	20
RoundManager.Events.PreperationEndEvent	39
RoundManager.Events.PreperationStartEvent	40
RoundManager.Events.RoundEndEvent	51
RoundManager.Events.RoundStartEvent	59
RoundManager.Events.RoundEvents	53
RoundManager.RoundLighting	55
RoundObjectSpawner	57
RoundManager.Vector2i	61

Chapter 3

Class Index

3.1 Class List

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

RoundManager.AudioOnRoundEnd	Example class reacting to round event RoundEndEvent . Plays an audio clip when a round ends.	13
RoundManager.AudioOnRoundStart	Example class reacting to round event RoundStartEvent . Plays an audio clip when a round starts.	13
RoundManager.BackgroundAudio	Used by BackgroundAudioChangeOnNewRound . Defines audio clips to play and their associated round number defining when to start the clip.	14
RoundManager.BackgroundAudioChangeOnNewRound	Example class reacting to round event RoundStartEvent . Changes background audio when a specific round is reached. Fades between previous audio clip and new clip.	14
RoundManager.BossCountdownBegun	Template class. Use this as a template to react to a BossCountdownBegunEvent . This event is raised whenever a boss countdown begins. Place your logic in the OnBossCountdownBegun function.	15
RoundManager.Events.BossCountdownBegunEvent	Raised when the boss countdown begins. See BossCountdownBegun template and BossCountdownText for a concrete example of how to react to the event.	16
RoundManager.BossCountdownText	Example class reacting to round event BossCountdownBegunEvent . Shows boss countdown text on screen when BossCountdownBegunEvent is raised.	16
RoundManager.BossKilled	Template class. Use this as a template to react to a BossKilledEvent . This event is raised whenever a boss is killed. Place your logic in the OnBossKilled function.	17
RoundManager.Events.BossKilledEvent	Raised when the round boss is killed. See BossKilled template for a template of how to react to the event.	18
RoundManager.BossSpawned	Template class. Use this as a template to react to a BossSpawnedEvent . This event is raised when a rounds boss is spawned. Place your logic in the OnBossSpawned function.	18
RoundManager.Events.BossSpawnedEvent	Raised when a boss is spawned. See BossSpawned for a template of how to react to this event. The event is currently raised in EnemySpawner . If you want to react to this event, and you are using a custom boss spawner then ensure that the event is raised when a boss is spawned. . .	19

RoundManager.BossSpawnRequest	20
Template class. Use this as a template to react to a BossSpawnRequestEvent . This event is raised when a boss should be spawned. Place your logic in the <code>#OnBossSpawnRequest</code> function.	
RoundManager.Events.BossSpawnRequestEvent	20
Raised when the a boss should be spawned. See BossSpawnRequest template and EnemySpawner for a concrete example of how to react to the event.	
RoundManager.CheckpointEnd	21
Template class. Use this as a template to react to a CheckpointEndEvent . This event is raised whenever a checkpoint finishes. Place your logic in the <code>OnCheckpointEnd</code> function.	
RoundManager.Events.CheckpointEndEvent	23
Raised when the a checkpoint ends. See CheckpointEnd template for an example of how to react to the event.	
RoundManager.CheckpointStart	24
Template class. Use this as a template to react to a CheckpointStartEvent . This event is raised whenever a checkpoint starts. Place your logic in the <code>OnCheckpointStart</code> function.	
RoundManager.Events.CheckpointStartEvent	24
Raised when the a checkpoint starts. See CheckpointStart template for an example of how to react to the event.	
RoundManager.DestroyCurrentEnemiesRequest	25
Template class. Use this as a template to react to a DestroyCurrentEnemiesRequestEvent . This event is raised when all currently spawned enemies should be killed. Place your logic in the <code>OnDestroyCurrentEnemiesRequest</code> function.	
RoundManager.Events.DestroyCurrentEnemiesRequestEvent	26
Raised to signify that all currently spawned enemies should be destroyed. See DestroyCurrentEnemiesRequest template and EnemySpawner for a concrete example of how to react to this event.	
RoundManager.Director	26
Entry point for the demo scene.	
RoundManager.EnemyHealth	27
Example enemy health script.	
RoundManager.EnemySpawner	28
Example class reacting to spawn request events. Spawns enemy when EnemySpawnRequestEvent is raised. Spawns boss when BossSpawnRequestEvent is raised. Destroys all spawned enemies when DestroyCurrentEnemiesRequestEvent is raised.	
RoundManager.EnemySpawnRequest	28
Template class. Use this as a template to react to a EnemySpawnRequestEvent . This event is raised when an enemy should be spawned. Place your logic in the <code>#OnEnemySpawnRequest</code> function.	
RoundManager.Events.EnemySpawnRequestEvent	29
Raised when the an enemy should be spawned. See EnemySpawnRequest template and EnemySpawner for a concrete example of how to react to the event.	
RoundManager.Environment	29
Generates the demo scene environment.	
RoundManager.FinishedRounds	30
Template class. Use this as a template to react to a FinishedRoundsEvent . This event is raised when all rounds are complete. Place your logic in the <code>OnFinishedRounds</code> function.	
RoundManager.Events.FinishedRoundsEvent	31
Raised to signify that all rounds have finished. See FinishedRounds for a template of how to react to this event and RoundsFinishedText for a concrete example.	
RoundManager.FirstRound	32
Template class. Use this as a template to react to a FirstRoundEvent . This event is raised at the beginning of the first round. Place your logic in the <code>OnFirstRound</code> function.	
RoundManager.Events.FirstRoundEvent	32
Raised to signify that the first round has started. See FirstRound for a template of how to react to this event.	

RoundManager.LastRound	Template class. Use this as a template to react to a LastRoundEvent . This event is raised at the beginning of the last round. Place your logic in the <code>#OnLastRound</code> function.	33
RoundManager.Events.LastRoundEvent	Raised to signify that the last round has started. See LastRound for a template of how to react to this event.	33
RoundManager.LerpOverTime	Lerp from one float to another over time.	34
RoundManager.LightingChangeOnNewRound	Example class reacting to round event RoundStartEvent . Changes scene lighting colour and intensity when a specific round is reached. Lerps between previous and new colour/intensity.	35
RoundManager.ObjectSpawner	Example class reacting to round event ObjectSpawnRequestEvent . Spawns preperation object when ObjectSpawnRequestEvent is raised.	35
RoundManager.ObjectSpawnRequest	Template class. Use this as a template to react to a ObjectSpawnRequestEvent . This event is raised when a object should be spawned. This is raised during the preperation stage of a round if the round has Round::PreperationTimeObjects . Place your logic in the <code>OnObjectSpawnRequest</code> function.	36
RoundManager.Events.ObjectSpawnRequestEvent	Raised when the an object should be spawned. See ObjectSpawnRequest template and ObjectSpawner for a concrete example of how to react to the event.	37
RoundManager.PreperationCountdownText	Example class reacting to round event PreperationStartEvent . Shows round preperation countdown text on screen when PreperationStartEvent is raised. . .	38
RoundManager.PreperationEnd	Template class. Use this as a template to react to a PreperationEndEvent . This event is raised when a rounds preperation stage has finished. Place your logic in the <code>OnPreperationEnd</code> function.	38
RoundManager.Events.PreperationEndEvent	Raised to signify that the current rounds preperation period has finished. See PreperationEnd for a template of how to react to this event.	39
RoundManager.PreperationStart	Template class. Use this as a template to react to a PreperationStartEvent . This event is raised when a rounds preperation stage starts. Place your logic in the <code>OnPreperationStart</code> function. .	40
RoundManager.Events.PreperationStartEvent	Raised to signify that the current rounds preperation period has started. See PreperationStart for a template of how to react to this event.	40
RoundManager.Round	Responsible for storing, managing, and updating an array of Checkpoint. Handles round preperation time (if HasPreperationTime), preperation object spawning, and boss spawning (if Round←HasBoss).	41
RoundManager.RoundBoss	Attach to any gameobject that is classed as a rounds boss. When the object attached to this script is disabled the current round is notified.	47
RoundManager.RoundCheckpoint	Responsible for raising EnemySpawnRequestEvent	47
RoundManager.RoundCountdownAudio	Plays Audio Clip when Round has {SecondsLeftToBeginAudio} until it will begin.	49
RoundManager.RoundCountdownText	Example class reacting to round event PreperationEndEvent . Shows round countdown text on screen when PreperationEndEvent is raised and the round starts.	50
RoundManager.RoundEnd	Template class. Use this as a template to react to a RoundEndEvent . This event is raised at the end of each round. Place your logic in the <code>OnRoundEnd</code> function.	50

RoundManager.Events.RoundEndEvent	
Raised everytime a round ends. See RoundEnd for a template of how to react to this event and AudioOnRoundEnd for a concrete example.	51
RoundManager.RoundEnemy	
Attach to any gameobject that is classed as a round enemy. It registers with the current round when an enemy is spawned and killed. Used for when you want the player to have killed all enemies before the round boss is spawned.	52
RoundManager.Events.RoundEvent	
Base Event class. All events should derive from this class.	52
RoundManager.Events.RoundEvents	
Round events.	53
RoundManager.RoundLighting	
Used by LightingChangeOnNewRound . Defines new light colour and intensity as well as the associated round number (defining when to change the lighting).	55
RoundManager.RoundManager	
Main round manager class. Responsible for storing and updating rounds, and transitioning from one round to the next.	55
RoundObjectSpawner	
Round object spawner. Includes an object prefab and an associated spawn weight. Used by RoundCheckpoint::RoundEnemies and Round::PreperationTimeObjects	57
RoundManager.RoundsFinishedText	
Example class reacting to round event FinishedRoundsEvent . Shows #TextToShow on screen when FinishedRoundsEvent is raised i.e. all rounds have finished.	58
RoundManager.RoundStart	
Template class. Use this as a template to react to a RoundStartEvent . This event is raised at the start of each round. Place your logic in the OnRoundStart function.	58
RoundManager.Events.RoundStartEvent	
Raised everytime a round starts. See RoundStart for a template of how to react to this event and AudioOnRoundStart for a concrete example.	59
RoundManager.ShowTextOnRoundStart	
Example class reacting to round event RoundStartEvent . Shows round number on screen when RoundStartEvent	60
RoundManager.Spawner	
Base Class for ObjectSpawner and EnemySpawner . Provides functionality to instantiate object.	60
RoundManager.TriggerRoundEnd	
Example class triggering the end of a round. Invokes Round::TriggerRoundEnd() when an object with tag 'Player' enters trigger. Can be used to end a round when player reaches specific point.	61
RoundManager.Vector2i	61

Chapter 4

Namespace Documentation

4.1 RoundManager Namespace Reference

Namespaces

- namespace [Events](#)

Classes

- class [AudioOnRoundEnd](#)
*Example class reacting to round event [RoundEndEvent](#).
Plays an audio clip when a round ends.*
- class [AudioOnRoundStart](#)
*Example class reacting to round event [RoundStartEvent](#).
Plays an audio clip when a round starts.*
- class [BackgroundAudio](#)
Used by [BackgroundAudioChangeOnNewRound](#). Defines audio clips to play and their associated round number defining when to start the clip.
- class [BackgroundAudioChangeOnNewRound](#)
*Example class reacting to round event [RoundStartEvent](#).
Changes background audio when a specific round is reached. Fades between previous audio clip and new clip.*
- class [BossCountdownBegun](#)
Template class. Use this as a template to react to a [BossCountdownBegunEvent](#). This event is raised whenever a boss countdown begins. Place your logic in the [OnBossCountdownBegun](#) function.
- class [BossCountdownText](#)
*Example class reacting to round event [BossCountdownBegunEvent](#).
Shows boss countdown text on screen when [BossCountdownBegunEvent](#) is raised.*
- class [BossKilled](#)
Template class. Use this as a template to react to a [BossKilledEvent](#). This event is raised whenever a boss is killed. Place your logic in the [OnBossKilled](#) function.
- class [BossSpawned](#)
Template class. Use this as a template to react to a [BossSpawnedEvent](#). This event is raised when a rounds boss is spawned. Place your logic in the [OnBossSpawned](#) function.
- class [BossSpawnRequest](#)
Template class. Use this as a template to react to a [BossSpawnRequestEvent](#). This event is raised when a boss should be spawned. Place your logic in the [#OnBossSpawnRequest](#) function.
- class [CheckpointEnd](#)
Template class. Use this as a template to react to a [CheckpointEndEvent](#). This event is raised whenever a checkpoint finishes. Place your logic in the [OnCheckpointEnd](#) function.

- class [CheckpointStart](#)
 Template class. Use this as a template to react to a [CheckpointStartEvent](#). This event is raised whenever a checkpoint starts. Place your logic in the [OnCheckpointStart](#) function.
- class [DestroyCurrentEnemiesRequest](#)
 Template class. Use this as a template to react to a [DestroyCurrentEnemiesRequestEvent](#). This event is raised when all currently spawned enemies should be killed. Place your logic in the [OnDestroyCurrentEnemiesRequest](#) function.
- class [Director](#)
 Entry point for the demo scene.
- class [EnemyHealth](#)
 Example enemy health script.
- class [EnemySpawner](#)
 Example class reacting to spawn request events.
 Spawns enemy when [EnemySpawnRequestEvent](#) is raised.
 Spawns boss when [BossSpawnRequestEvent](#) is raised.
 Destroys all spawned enemies when [DestroyCurrentEnemiesRequestEvent](#) is raised.
- class [EnemySpawnRequest](#)
 Template class. Use this as a template to react to a [EnemySpawnRequestEvent](#). This event is raised when an enemy should be spawned. Place your logic in the [#OnEnemySpawnRequest](#) function.
- class [Environment](#)
 Generates the demo scene environment.
- class [FinishedRounds](#)
 Template class. Use this as a template to react to a [FinishedRoundsEvent](#). This event is raised when all rounds are complete. Place your logic in the [OnFinishedRounds](#) function.
- class [FirstRound](#)
 Template class. Use this as a template to react to a [FirstRoundEvent](#). This event is raised at the beginning of the first round. Place your logic in the [OnFirstRound](#) function.
- class [LastRound](#)
 Template class. Use this as a template to react to a [LastRoundEvent](#). This event is raised at the beginning of the last round. Place your logic in the [#OnLastRound](#) function.
- class [LerpOverTime](#)
 Lerp from one float to another over time.
- class [LightingChangeOnNewRound](#)
 Example class reacting to round event [RoundStartEvent](#).
 Changes scene lighting colour and intensity when a specific round is reached. Lerps between previous and new colour/intensity.
- class [ObjectSpawner](#)
 Example class reacting to round event [ObjectSpawnRequestEvent](#).
 Spawns preperation object when [ObjectSpawnRequestEvent](#) is raised.
- class [ObjectSpawnRequest](#)
 Template class. Use this as a template to react to a [ObjectSpawnRequestEvent](#). This event is raised when a object should be spawned. This is raised during the preperation stage of a round if the round has [Round::Preperation↔TimeObjects](#). Place your logic in the [OnObjectSpawnRequest](#) function.
- class [PreperationCountdownText](#)
 Example class reacting to round event [PreperationStartEvent](#).
 Shows round preperation countdown text on screen when [PreperationStartEvent](#) is raised.
- class [PreperationEnd](#)
 Template class. Use this as a template to react to a [PreperationEndEvent](#). This event is raised when a rounds preperation stage has finished. Place your logic in the [OnPreperationEnd](#) function.
- class [PreperationStart](#)
 Template class. Use this as a template to react to a [PreperationStartEvent](#). This event is raised when a rounds preperation stage starts. Place your logic in the [OnPreperationStart](#) function.
- class [Round](#)
 Responsible for storing, managing, and updating an array of Checkpoint. Handles round preperation time (if [Has↔PreperationTime](#)), preperation object spawning, and boss spawning (if [RoundHasBoss](#)).

- class [RoundBoss](#)
Attach to any gameobject that is classed as a rounds boss. When the object attached to this script is disabled the current round is notified.
- class [RoundCheckpoint](#)
Responsible for raising [EnemySpawnRequestEvent](#).
- class [RoundCountdownAudio](#)
Plays Audio Clip when [Round](#) has {SecondsLeftToBeginAudio} until it will begin.
- class [RoundCountdownText](#)
*Example class reacting to round event [PreperationEndEvent](#).
Shows round countdown text on screen when [PreperationEndEvent](#) is raised and the round starts.*
- class [RoundEnd](#)
*Template class. Use this as a template to react to a [RoundEndEvent](#). This event is raised at the end of each round.
Place your logic in the [OnRoundEnd](#) function.*
- class [RoundEnemy](#)
Attach to any gameobject that is classed as a round enemy. It registers with the current round when an enemy is spawned and killed. Used for when you want he player to have killed all enemies before the round boss is spawned.
- class [RoundLighting](#)
Used by [LightingChangeOnNewRound](#). Defines new light colour and intensity as well as the associated round number (defining when to change the lighting).
- class [RoundManager](#)
Main round manager class. Responsible for storing and updating rounds, and transitioning from one round to the next.
- class [RoundsFinishedText](#)
*Example class reacting to round event [FinishedRoundsEvent](#).
Shows #TextToShow on screen when [FinishedRoundsEvent](#) is raised i.e. all rounds have finished.*
- class [RoundStart](#)
*Template class. Use this as a template to react to a [RoundStartEvent](#). This event is raised at the start of each round.
Place your logic in the [OnRoundStart](#) function.*
- class [ShowTextOnRoundStart](#)
*Example class reacting to round event [RoundStartEvent](#).
Shows round number on screen when [RoundStartEvent](#).*
- class [Spawner](#)
Base Class for [ObjectSpawner](#) and [EnemySpawner](#). Provides functionality to instantiate object.
- class [TriggerRoundEnd](#)
*Example class triggering the end of a round.
Invokes [Round::TriggerRoundEnd\(\)](#) when an object with tag 'Player' enters trigger. Can be used to end a round when player reaches specific point.*
- struct [Vector2i](#)

4.2 RoundManager.Events Namespace Reference

Classes

- class [BossCountdownBegunEvent](#)
Raised when the boss countdown begins. See [BossCountdownBegun](#) template and [BossCountdownText](#) for a concrete example of how to react to the event.
- class [BossKilledEvent](#)
Raised when the round boss is killed. See [BossKilled](#) template for a template of how to react to the event.
- class [BossSpawnedEvent](#)
Raised when a boss is spawned. See [BossSpawned](#) for a template of how to react to this event. The event is currently raised in [EnemySpawner](#). If you want to react to this event, and you are using a custom boss spawner then ensure that the event is raised when a boss is spawned.
- class [BossSpawnRequestEvent](#)

Raised when the a boss should be spawned. See [BossSpawnRequest](#) template and [EnemySpawner](#) for a concrete example of how to react to the event.

- class [CheckpointEndEvent](#)

Raised when the a checkpoint ends. See [CheckPointEnd](#) template for an example of how to react to the event.

- class [CheckpointStartEvent](#)

Raised when the a checkpoint starts. See [CheckPointStart](#) template for an example of how to react to the event.

- class [DestroyCurrentEnemiesRequestEvent](#)

Raised to signify that all currently spawned enemies should be destroyed. See [DestroyCurrentEnemiesRequest](#) template and [EnemySpawner](#) for a concrete example of how to react to this event.

- class [EnemySpawnRequestEvent](#)

Raised when the an enemy should be spawned. See [EnemySpawnRequest](#) template and [EnemySpawner](#) for a concrete example of how to react to the event.

- class [FinishedRoundsEvent](#)

Raised to signify that all rounds have finished. See [FinishedRounds](#) for a template of how to react to this event and [RoundsFinishedText](#) for a concrete example.

- class [FirstRoundEvent](#)

Raised to signify that the first round has started. See [FirstRound](#) for a template of how to react to this event.

- class [LastRoundEvent](#)

Raised to signify that the last round has started. See [LastRound](#) for a template of how to react to this event.

- class [ObjectSpawnRequestEvent](#)

Raised when the an object should be spawned. See [ObjectSpawnRequest](#) template and [ObjectSpawner](#) for a concrete example of how to react to the event.

- class [PreperationEndEvent](#)

Raised to signify that the current rounds preperation period has finished. See [PreperationEnd](#) for a template of how to react to this event.

- class [PreperationStartEvent](#)

Raised to signify that the current rounds preperation period has started. See [PreperationStart](#) for a template of how to react to this event.

- class [RoundEndEvent](#)

Raised everytime a round ends. See [RoundEnd](#) for a template of how to react to this event and [AudioOnRoundEnd](#) for a concrete example.

- class [RoundEvent](#)

Base Event class. All events should derive from this class.

- class [RoundEvents](#)

[Round](#) events.

- class [RoundStartEvent](#)

Raised everytime a round starts. See [RoundStart](#) for a template of how to react to this event and [AudioOnRoundStart](#) for a concrete example.

Chapter 5

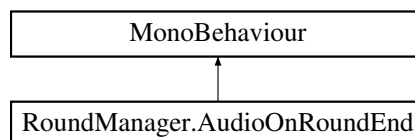
Class Documentation

5.1 RoundManager.AudioOnRoundEnd Class Reference

Example class reacting to round event [RoundEndEvent](#).

Plays an audio clip when a round ends.

Inheritance diagram for RoundManager.AudioOnRoundEnd:



Public Attributes

- AudioClip **Audio**
- float **Volume** = 1f

5.1.1 Detailed Description

Example class reacting to round event [RoundEndEvent](#).

Plays an audio clip when a round ends.

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

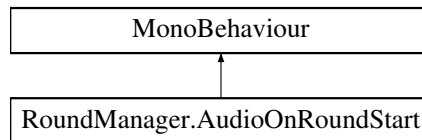
- Round System/Scripts/Demo Scripts/Event Examples/Audio/AudioOnRoundEnd.cs

5.2 RoundManager.AudioOnRoundStart Class Reference

Example class reacting to round event [RoundStartEvent](#).

Plays an audio clip when a round starts.

Inheritance diagram for RoundManager.AudioOnRoundStart:



Public Attributes

- AudioClip **Audio**
- float **Volume** = 1f

5.2.1 Detailed Description

Example class reacting to round event [RoundStartEvent](#).

Plays an audio clip when a round starts.

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

- Round System/Scripts/Demo Scripts/Event Examples/Audio/AudioOnRoundStart.cs

5.3 RoundManager.BackgroundAudio Class Reference

Used by [BackgroundAudioChangeOnNewRound](#). Defines audio clips to play and their associated round number defining when to start the clip.

Public Attributes

- int **RoundNumber**
- AudioClip **Audio**

5.3.1 Detailed Description

Used by [BackgroundAudioChangeOnNewRound](#). Defines audio clips to play and their associated round number defining when to start the clip.

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

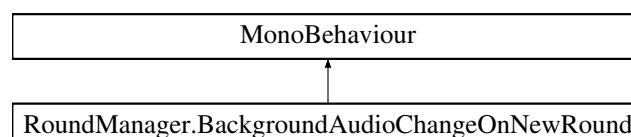
- Round System/Scripts/Demo Scripts/Event Examples/Audio/Background Audio/BackgroundAudio.cs

5.4 RoundManager.BackgroundAudioChangeOnNewRound Class Reference

Example class reacting to round event [RoundStartEvent](#).

Changes background audio when a specific round is reached. Fades between previous audio clip and new clip.

Inheritance diagram for RoundManager.BackgroundAudioChangeOnNewRound:



Public Attributes

- [BackgroundAudio\[\]](#) [Audio](#)

Defines audio clips to play and their associated round number (defining when to start the clip).

5.4.1 Detailed Description

Example class reacting to round event [RoundStartEvent](#).

Changes background audio when a specific round is reached. Fades between previous audio clip and new clip.

5.4.2 Member Data Documentation

5.4.2.1 [BackgroundAudio \[\]](#) [RoundManager.BackgroundAudioChangeOnNewRound.Audio](#)

Defines audio clips to play and their associated round number (defining when to start the clip).

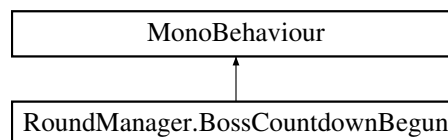
The documentation for this class was generated from the following file:

- Round System/Scripts/Demo Scripts/Event Examples/Audio/Background Audio/BackgroundAudioChange↔
OnNewRound.cs

5.5 RoundManager.BossCountdownBegun Class Reference

Template class. Use this as a template to react to a [BossCountdownBegunEvent](#). This event is raised whenever a boss countdown begins. Place your logic in the [OnBossCountdownBegun](#) function.

Inheritance diagram for RoundManager.BossCountdownBegun:



Public Member Functions

- void [OnBossCountdownBegun](#) ([BossCountdownBegunEvent](#) e)

React to [BossCountdownBegunEvent](#).

5.5.1 Detailed Description

Template class. Use this as a template to react to a [BossCountdownBegunEvent](#). This event is raised whenever a boss countdown begins. Place your logic in the [OnBossCountdownBegun](#) function.

5.5.2 Member Function Documentation

5.5.2.1 void [RoundManager.BossCountdownBegun.OnBossCountdownBegun](#) ([BossCountdownBegunEvent](#) e)

React to [BossCountdownBegunEvent](#).

Parameters

<i>e</i>	Event.
----------	--------

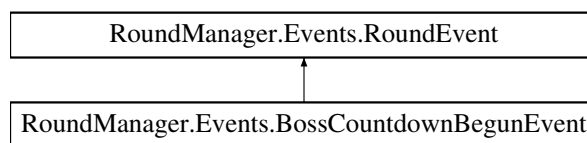
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Boss Events/BossCountdownBegun.cs

5.6 RoundManager.Events.BossCountdownBegunEvent Class Reference

Raised when the boss countdown begins. See [BossCountdownBegun](#) template and [BossCountdownText](#) for a concrete example of how to react to the event.

Inheritance diagram for RoundManager.Events.BossCountdownBegunEvent:



Public Member Functions

- **BossCountdownBegunEvent** ([Round](#) currentRound)

Properties

- [Round CurrentRound](#) [get]
Gets the current round that raised this event.

5.6.1 Detailed Description

Raised when the boss countdown begins. See [BossCountdownBegun](#) template and [BossCountdownText](#) for a concrete example of how to react to the event.

5.6.2 Property Documentation

5.6.2.1 Round RoundManager.Events.BossCountdownBegunEvent.CurrentRound [get]

Gets the current round that raised this event.

The current round.

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

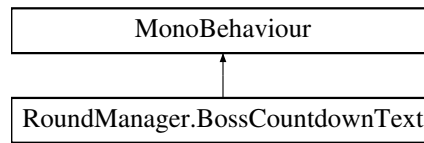
- Round System/Scripts/_Round System/Event System/Individual Events/Boss Events/BossCountdown↔ BegunEvent.cs

5.7 RoundManager.BossCountdownText Class Reference

Example class reacting to round event [BossCountdownBegunEvent](#).

Shows boss countdown text on screen when [BossCountdownBegunEvent](#) is raised.

Inheritance diagram for RoundManager.BossCountdownText:



Public Attributes

- string **PreBossCountdownTimeText**
- string **PostBossCountdownText**

5.7.1 Detailed Description

Example class reacting to round event [BossCountdownBegunEvent](#). Shows boss countdown text on screen when [BossCountdownBegunEvent](#) is raised.

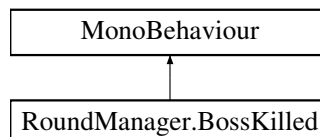
The documentation for this class was generated from the following file:

- Round System/Scripts/Demo Scripts/Event Examples/Text/BossCountdownText.cs

5.8 RoundManager.BossKilled Class Reference

Template class. Use this as a template to react to a [BossKilledEvent](#). This event is raised whenever a boss is killed. Place your logic in the [OnBossKilled](#) function.

Inheritance diagram for RoundManager.BossKilled:



Public Member Functions

- void [OnBossKilled](#) ([BossKilledEvent](#) e)
React to [BossKilledEvent](#).

5.8.1 Detailed Description

Template class. Use this as a template to react to a [BossKilledEvent](#). This event is raised whenever a boss is killed. Place your logic in the [OnBossKilled](#) function.

5.8.2 Member Function Documentation

5.8.2.1 void RoundManager.BossKilled.OnBossKilled ([BossKilledEvent](#) e)

React to [BossKilledEvent](#).

Parameters

<i>e</i>	Event.
----------	--------

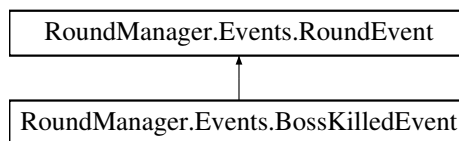
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Boss Events/BossKilled.cs

5.9 RoundManager.Events.BossKilledEvent Class Reference

Raised when the round boss is killed. See [BossKilled](#) template for a template of how to react to the event.

Inheritance diagram for RoundManager.Events.BossKilledEvent:



Public Member Functions

- **BossKilledEvent** ([Round](#) currentRound)

Properties

- [Round CurrentRound](#) [get]
Gets the current round that raised this event.

5.9.1 Detailed Description

Raised when the round boss is killed. See [BossKilled](#) template for a template of how to react to the event.

5.9.2 Property Documentation

5.9.2.1 Round RoundManager.Events.BossKilledEvent.CurrentRound [get]

Gets the current round that raised this event.

The current round.

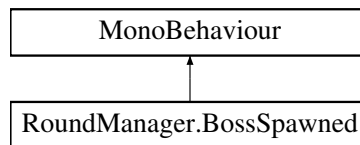
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/Individual Events/Boss Events/BossKilledEvent.cs

5.10 RoundManager.BossSpawned Class Reference

Template class. Use this as a template to react to a [BossSpawnedEvent](#). This event is raised when a rounds boss is spawned. Place your logic in the [OnBossSpawned](#) function.

Inheritance diagram for RoundManager.BossSpawned:



Public Member Functions

- void [OnBossSpawned](#) ([BossSpawnedEvent](#) e)
React to [BossSpawnedEvent](#).

5.10.1 Detailed Description

Template class. Use this as a template to react to a [BossSpawnedEvent](#). This event is raised when a rounds boss is spawned. Place your logic in the [OnBossSpawned](#) function.

5.10.2 Member Function Documentation

5.10.2.1 void RoundManager.BossSpawned.OnBossSpawned ([BossSpawnedEvent](#) e)

React to [BossSpawnedEvent](#).

Parameters

<i>e</i>	Event.
----------	--------

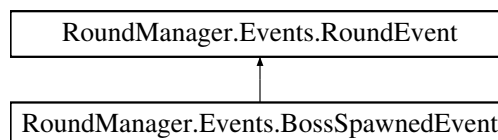
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Boss Events/BossSpawned.cs

5.11 RoundManager.Events.BossSpawnedEvent Class Reference

Raised when a boss is spawned. See [BossSpawned](#) for a template of how to react to this event. The event is currently raised in [EnemySpawner](#). If you want to react to this event, and you are using a custom boss spawner then ensure that the event is raised when a boss is spawned.

Inheritance diagram for RoundManager.Events.BossSpawnedEvent:



Public Member Functions

- [BossSpawnedEvent](#) ([Round](#) currentRound, GameObject boss)

Properties

- GameObject [Boss](#) [get]
Returns the instantiated boss.
- [Round](#) [CurrentRound](#) [get]

5.11.1 Detailed Description

Raised when a boss is spawned. See [BossSpawned](#) for a template of how to react to this event. The event is currently raised in [EnemySpawner](#). If you want to react to this event, and you are using a custom boss spawner then ensure that the event is raised when a boss is spawned.

5.11.2 Property Documentation

5.11.2.1 `GameObject RoundManager.Events.BossSpawnedEvent.Boss` [get]

Returns the instantiated boss.

The boss.

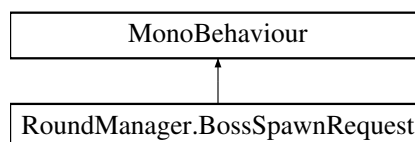
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/Individual Events/Boss Events/BossSpawnedEvent.cs

5.12 RoundManager.BossSpawnRequest Class Reference

Template class. Use this as a template to react to a [BossSpawnRequestEvent](#). This event is raised when a boss should be spawned. Place your logic in the `#OnBossSpawnRequest` function.

Inheritance diagram for RoundManager.BossSpawnRequest:



5.12.1 Detailed Description

Template class. Use this as a template to react to a [BossSpawnRequestEvent](#). This event is raised when a boss should be spawned. Place your logic in the `#OnBossSpawnRequest` function.

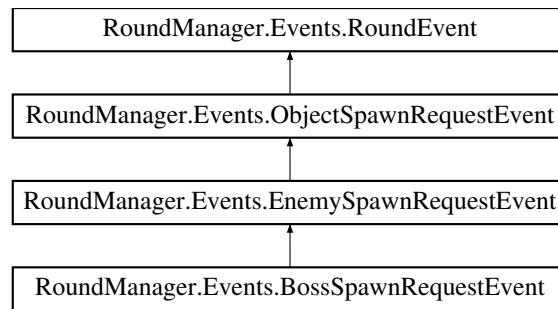
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Spawn Request Events/BossSpawnRequest.cs

5.13 RoundManager.Events.BossSpawnRequestEvent Class Reference

Raised when the a boss should be spawned. See [BossSpawnRequest](#) template and [EnemySpawner](#) for a concrete example of how to react to the event.

Inheritance diagram for RoundManager.Events.BossSpawnRequestEvent:



Public Member Functions

- **BossSpawnRequestEvent** ([Round](#) currentRound, `GameObject` enemyPrefab)

Additional Inherited Members

5.13.1 Detailed Description

Raised when the a boss should be spawned. See [BossSpawnRequest](#) template and [EnemySpawner](#) for a concrete example of how to react to the event.

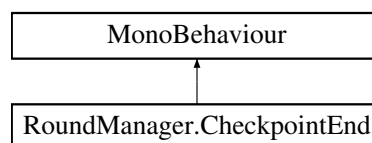
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/Individual Events/Boss Events/BossSpawnRequestEvent.cs

5.14 RoundManager.CheckpointEnd Class Reference

Template class. Use this as a template to react to a [CheckpointEndEvent](#). This event is raised whenever a checkpoint finishes. Place your logic in the [OnCheckpointEnd](#) function.

Inheritance diagram for RoundManager.CheckpointEnd:



Public Member Functions

- void [OnCheckpointEnd](#) ([CheckpointEndEvent](#) e)
React to [CheckpointEndEvent](#).

5.14.1 Detailed Description

Template class. Use this as a template to react to a [CheckpointEndEvent](#). This event is raised whenever a checkpoint finishes. Place your logic in the [OnCheckpointEnd](#) function.

5.14.2 Member Function Documentation

5.14.2.1 void RoundManager.CheckpointEnd.OnCheckpointEnd (CheckpointEndEvent e)

React to [CheckpointEndEvent](#).

Parameters

<i>e</i>	Event.
----------	--------

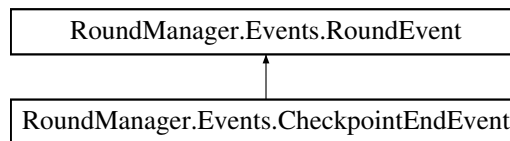
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Checkpoint Events/CheckpointEnd.cs

5.15 RoundManager.Events.CheckpointEndEvent Class Reference

Raised when the a checkpoint ends. See CheckPointEnd template for an example of how to react to the event.

Inheritance diagram for RoundManager.Events.CheckpointEndEvent:



Public Member Functions

- **CheckpointEndEvent** ([Round](#) currentRound, bool hasNextCheckpoint)

Properties

- [Round CurrentRound](#) [get]
Gets the current round that raised this event.
- bool [HasNextCheckpoint](#) [get]
A value indicating whether there is another checkpoint after this one.

5.15.1 Detailed Description

Raised when the a checkpoint ends. See CheckPointEnd template for an example of how to react to the event.

5.15.2 Property Documentation

5.15.2.1 Round RoundManager.Events.CheckpointEndEvent.CurrentRound [get]

Gets the current round that raised this event.

The current round.

5.15.2.2 bool RoundManager.Events.CheckpointEndEvent.HasNextCheckpoint [get]

A value indicating whether there is another checkpoint after this one.

true if this instance has next checkpoint; otherwise, false.

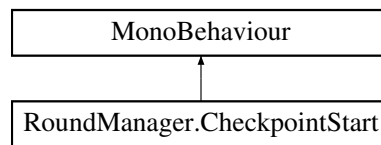
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/Individual Events/Checkpoint Events/Checkpoint↵EndEvent.cs

5.16 RoundManager.CheckpointStart Class Reference

Template class. Use this as a template to react to a [CheckpointStartEvent](#). This event is raised whenever a checkpoint starts. Place your logic in the [OnCheckpointStart](#) function.

Inheritance diagram for RoundManager.CheckpointStart:



Public Member Functions

- void [OnCheckpointStart](#) ([CheckpointStartEvent](#) e)
React to [CheckpointStartEvent](#).

5.16.1 Detailed Description

Template class. Use this as a template to react to a [CheckpointStartEvent](#). This event is raised whenever a checkpoint starts. Place your logic in the [OnCheckpointStart](#) function.

5.16.2 Member Function Documentation

5.16.2.1 void RoundManager.CheckpointStart.OnCheckpointStart (CheckpointStartEvent e)

React to [CheckpointStartEvent](#).

Parameters

<i>e</i>	Event.
----------	--------

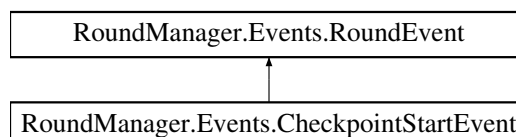
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Checkpoint Events/CheckpointStart.cs

5.17 RoundManager.Events.CheckpointStartEvent Class Reference

Raised when the a checkpoint starts. See CheckPointStart template for an example of how to react to the event.

Inheritance diagram for RoundManager.Events.CheckpointStartEvent:



Public Member Functions

- **CheckpointStartEvent** ([Round](#) currentRound)

Properties

- [Round CurrentRound](#) [get]
Gets the current round that raised this event.

5.17.1 Detailed Description

Raised when the a checkpoint starts. See CheckPointStart template for an example of how to react to the event.

5.17.2 Property Documentation

5.17.2.1 Round RoundManager.Events.CheckpointStartEvent.CurrentRound [get]

Gets the current round that raised this event.

The current round.

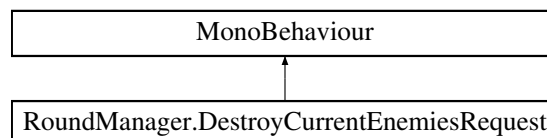
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/Individual Events/Checkpoint Events/CheckpointStart↔Event.cs

5.18 RoundManager.DestroyCurrentEnemiesRequest Class Reference

Template class. Use this as a template to react to a [DestroyCurrentEnemiesRequestEvent](#). This event is raised when all currently spawned enemies should be killed. Place your logic in the [OnDestroyCurrentEnemiesRequest](#) function.

Inheritance diagram for RoundManager.DestroyCurrentEnemiesRequest:



Public Member Functions

- void [OnDestroyCurrentEnemiesRequest](#) ([DestroyCurrentEnemiesRequestEvent](#) e)
React to [DestroyCurrentEnemiesRequestEvent](#). Place your logic here to destroy/disable all spawned enemies.

5.18.1 Detailed Description

Template class. Use this as a template to react to a [DestroyCurrentEnemiesRequestEvent](#). This event is raised when all currently spawned enemies should be killed. Place your logic in the [OnDestroyCurrentEnemiesRequest](#) function.

5.18.2 Member Function Documentation

5.18.2.1 void RoundManager.DestroyCurrentEnemiesRequest.OnDestroyCurrentEnemiesRequest ([DestroyCurrentEnemiesRequestEvent](#) e)

React to [DestroyCurrentEnemiesRequestEvent](#). Place your logic here to destroy/disable all spawned enemies.

Parameters

<i>e</i>	Event.
----------	--------

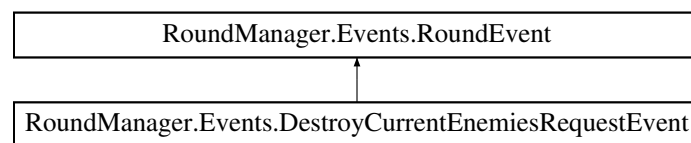
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Enemy Events/DestroyCurrentEnemiesRequest.cs

5.19 RoundManager.Events.DestroyCurrentEnemiesRequestEvent Class Reference

Raised to signify that all currently spawned enemies should be destroyed. See [DestroyCurrentEnemiesRequest](#) template and [EnemySpawner](#) for a concrete example of how to react to this event.

Inheritance diagram for RoundManager.Events.DestroyCurrentEnemiesRequestEvent:



Public Member Functions

- **DestroyCurrentEnemiesRequestEvent** ([Round](#) currentRound)

Properties

- [Round CurrentRound](#) [get]
Gets the current round that raised this event.

5.19.1 Detailed Description

Raised to signify that all currently spawned enemies should be destroyed. See [DestroyCurrentEnemiesRequest](#) template and [EnemySpawner](#) for a concrete example of how to react to this event.

5.19.2 Property Documentation

5.19.2.1 Round RoundManager.Events.DestroyCurrentEnemiesRequestEvent.CurrentRound [get]

Gets the current round that raised this event.

The current round.

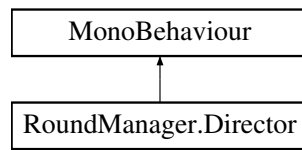
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/Individual Events/Enemy Events/DestroyCurrentEnemiesRequestEvent.cs

5.20 RoundManager.Director Class Reference

Entry point for the demo scene.

Inheritance diagram for RoundManager.Director:



Public Attributes

- [Environment](#) **EnvironmentGenerator**

5.20.1 Detailed Description

Entry point for the demo scene.

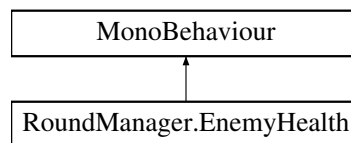
The documentation for this class was generated from the following file:

- Round System/Scripts/Demo Scripts/Director.cs

5.21 RoundManager.EnemyHealth Class Reference

Example enemy health script.

Inheritance diagram for RoundManager.EnemyHealth:



Public Member Functions

- void **OnHit** (int amount)
- void **OnDead** ()

Public Attributes

- int **MaxHealth**
- AudioClip[] **OnHitSounds**
- GameObject **OnDeadAnimation**
- GameObject[] **OnDeadSprites**
- GameObject[] **CollectiblesPrefabs**
- int **MinCollectiblesDropped**
- int **MaxCollectiblesDropped**

5.21.1 Detailed Description

Example enemy health script.

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

- Round System/Scripts/Demo Scripts/EnemyHealth.cs

5.22 RoundManager.EnemySpawner Class Reference

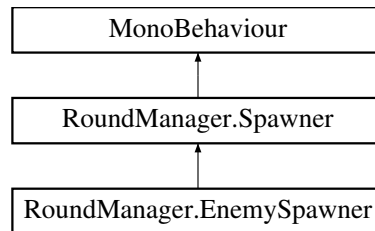
Example class reacting to spawn request events.

Spawns enemy when [EnemySpawnRequestEvent](#) is raised.

Spawns boss when [BossSpawnRequestEvent](#) is raised.

Destroys all spawned enemies when [DestroyCurrentEnemiesRequestEvent](#) is raised.

Inheritance diagram for RoundManager.EnemySpawner:



Additional Inherited Members

5.22.1 Detailed Description

Example class reacting to spawn request events.

Spawns enemy when [EnemySpawnRequestEvent](#) is raised.

Spawns boss when [BossSpawnRequestEvent](#) is raised.

Destroys all spawned enemies when [DestroyCurrentEnemiesRequestEvent](#) is raised.

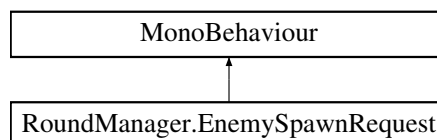
The documentation for this class was generated from the following file:

- Round System/Scripts/Demo Scripts/Event Examples/Spawn/EnemySpawner.cs

5.23 RoundManager.EnemySpawnRequest Class Reference

Template class. Use this as a template to react to a [EnemySpawnRequestEvent](#). This event is raised when an enemy should be spawned. Place your logic in the `#OnEnemySpawnRequest` function.

Inheritance diagram for RoundManager.EnemySpawnRequest:



5.23.1 Detailed Description

Template class. Use this as a template to react to a [EnemySpawnRequestEvent](#). This event is raised when an enemy should be spawned. Place your logic in the `#OnEnemySpawnRequest` function.

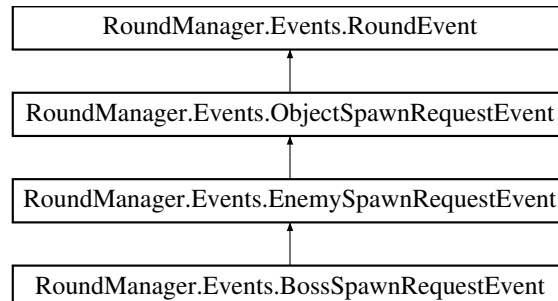
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Spawn Request Events/EnemySpawnRequest.cs

5.24 RoundManager.Events.EnemySpawnRequestEvent Class Reference

Raised when the an enemy should be spawned. See [EnemySpawnRequest](#) template and [EnemySpawner](#) for a concrete example of how to react to the event.

Inheritance diagram for RoundManager.Events.EnemySpawnRequestEvent:



Public Member Functions

- **EnemySpawnRequestEvent** ([Round](#) currentRound, GameObject enemyPrefab)

Additional Inherited Members

5.24.1 Detailed Description

Raised when the an enemy should be spawned. See [EnemySpawnRequest](#) template and [EnemySpawner](#) for a concrete example of how to react to the event.

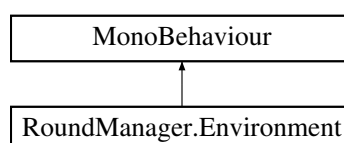
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/Individual Events/Spawn Request Events/EnemySpawnRequestEvent.cs

5.25 RoundManager.Environment Class Reference

Generates the demo scene environment.

Inheritance diagram for RoundManager.Environment:



Public Member Functions

- void [GenerateFloor](#) ()
Generates the demo environment.

Public Attributes

- GameObject **FloorPrefab**
- Sprite[] **FloorSprites**
- Rect **RoomSize**
- **Vector2i EnemySpawnArea** = new **Vector2i** (3, 3)

The enemy spawn area consists of the blocks surrounding the floor area. This defines the size of the surrounding area.

Properties

- List< Transform > **ObjectSpawnTiles** [get]
- List< Transform > **EnemySpawnTiles** [get]

5.25.1 Detailed Description

Generates the demo scene environment.

5.25.2 Member Function Documentation

5.25.2.1 void RoundManager.Environment.GenerateFloor ()

Generates the demo environment.

5.25.3 Member Data Documentation

5.25.3.1 Vector2i RoundManager.Environment.EnemySpawnArea = new Vector2i (3, 3)

The enemy spawn area consists of the blocks surrounding the floor area. This defines the size of the surrounding area.

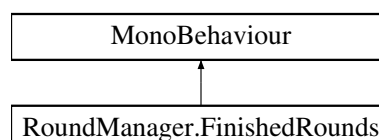
The documentation for this class was generated from the following file:

- Round System/Scripts/Demo Scripts/Environment/Environment.cs

5.26 RoundManager.FinishedRounds Class Reference

Template class. Use this as a template to react to a [FinishedRoundsEvent](#). This event is raised when all rounds are complete. Place your logic in the [OnFinishedRounds](#) function.

Inheritance diagram for RoundManager.FinishedRounds:



Public Member Functions

- void [OnFinishedRounds](#) ([FinishedRoundsEvent](#) e)
- React to [FinishedRoundsEvent](#).*

5.26.1 Detailed Description

Template class. Use this as a template to react to a [FinishedRoundsEvent](#). This event is raised when all rounds are complete. Place your logic in the [OnFinishedRounds](#) function.

5.26.2 Member Function Documentation

5.26.2.1 void RoundManager.FinishedRounds.OnFinishedRounds (FinishedRoundsEvent e)

React to [FinishedRoundsEvent](#).

Parameters

<i>e</i>	Event.
----------	--------

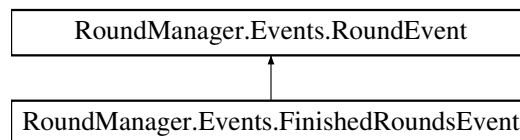
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Round Events/FinishedRounds.cs

5.27 RoundManager.Events.FinishedRoundsEvent Class Reference

Raised to signify that all rounds have finished. See [FinishedRounds](#) for a template of how to react to this event and [RoundsFinishedText](#) for a concrete example.

Inheritance diagram for RoundManager.Events.FinishedRoundsEvent:



Public Member Functions

- **FinishedRoundsEvent** ([Round](#) currentRound)

Properties

- [Round LastRound](#) [get]
Returns the last round.

5.27.1 Detailed Description

Raised to signify that all rounds have finished. See [FinishedRounds](#) for a template of how to react to this event and [RoundsFinishedText](#) for a concrete example.

5.27.2 Property Documentation

5.27.2.1 Round RoundManager.Events.FinishedRoundsEvent.LastRound [get]

Returns the last round.

The last round.

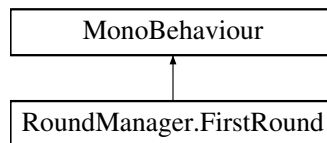
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/Individual Events/Round Events/FinishedRounds↔ Event.cs

5.28 RoundManager.FirstRound Class Reference

Template class. Use this as a template to react to a [FirstRoundEvent](#). This event is raised at the beginning of the first round. Place your logic in the [OnFirstRound](#) function.

Inheritance diagram for RoundManager.FirstRound:



Public Member Functions

- void [OnFirstRound](#) ([FirstRoundEvent](#) e)
React to [FirstRoundEvent](#).

5.28.1 Detailed Description

Template class. Use this as a template to react to a [FirstRoundEvent](#). This event is raised at the beginning of the first round. Place your logic in the [OnFirstRound](#) function.

5.28.2 Member Function Documentation

5.28.2.1 void RoundManager.FirstRound.OnFirstRound (FirstRoundEvent e)

React to [FirstRoundEvent](#).

Parameters

<code>e</code>	Event.
----------------	--------

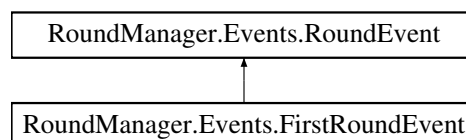
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_ Templates/Round Events/FirstRound.cs

5.29 RoundManager.Events.FirstRoundEvent Class Reference

Raised to signify that the first round has started. See [FirstRound](#) for a template of how to react to this event.

Inheritance diagram for RoundManager.Events.FirstRoundEvent:



Public Member Functions

- **FirstRoundEvent** ([Round](#) round)

Properties

- [Round CurrentRound](#) [get]
Gets the first round.

5.29.1 Detailed Description

Raised to signify that the first round has started. See [FirstRound](#) for a template of how to react to this event.

5.29.2 Property Documentation

5.29.2.1 Round RoundManager.Events.FirstRoundEvent.CurrentRound [get]

Gets the first round.

The first round.

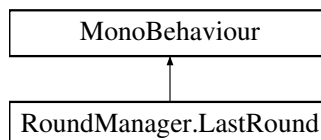
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/Individual Events/Round Events/FirstRoundEvent.cs

5.30 RoundManager.LastRound Class Reference

Template class. Use this as a template to react to a [LastRoundEvent](#). This event is raised at the beginning of the last round. Place your logic in the #OnLastRound function.

Inheritance diagram for RoundManager.LastRound:



5.30.1 Detailed Description

Template class. Use this as a template to react to a [LastRoundEvent](#). This event is raised at the beginning of the last round. Place your logic in the #OnLastRound function.

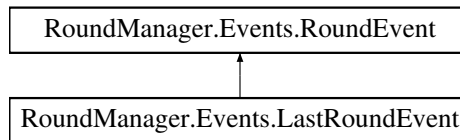
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Round Events/LastRound.cs

5.31 RoundManager.Events.LastRoundEvent Class Reference

Raised to signify that the last round has started. See [LastRound](#) for a template of how to react to this event.

Inheritance diagram for RoundManager.Events.LastRoundEvent:



Public Member Functions

- **LastRoundEvent** ([Round](#) round)

Properties

- [Round CurrentRound](#) [get]
Gets the last round.

5.31.1 Detailed Description

Raised to signify that the last round has started. See [LastRound](#) for a template of how to react to this event.

5.31.2 Property Documentation

5.31.2.1 Round RoundManager.Events.LastRoundEvent.CurrentRound [get]

Gets the last round.

The last round.

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

- Round System/Scripts/_Round System/Event System/Individual Events/Round Events/LastRoundEvent.cs

5.32 RoundManager.LerpOverTime Class Reference

Lerp from one float to another over time.

Public Member Functions

- **LerpOverTime** (float from, float to, float duration)
- void **Start** ()

Properties

- float **ElapsedTime** [get]
- float [Value](#) [get]
Current lerped value.

5.32.1 Detailed Description

Lerp from one float to another over time.

5.32.2 Property Documentation

5.32.2.1 float RoundManager.LerpOverTime.Value [get]

Current lerped value.

The value.

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

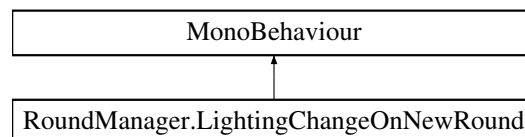
- Round System/Scripts/_Round System/LerpOverTime.cs

5.33 RoundManager.LightingChangeOnNewRound Class Reference

Example class reacting to round event [RoundStartEvent](#).

Changes scene lighting colour and intensity when a specific round is reached. Lerps between previous and new colour/intensity.

Inheritance diagram for RoundManager.LightingChangeOnNewRound:



Public Attributes

- [RoundLighting\[\] Lighting](#)

Defines new light colour and intensity as well as the associated round number (defining when to change the lighting).

5.33.1 Detailed Description

Example class reacting to round event [RoundStartEvent](#).

Changes scene lighting colour and intensity when a specific round is reached. Lerps between previous and new colour/intensity.

5.33.2 Member Data Documentation

5.33.2.1 RoundLighting [] RoundManager.LightingChangeOnNewRound.Lighting

Defines new light colour and intensity as well as the associated round number (defining when to change the lighting).

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

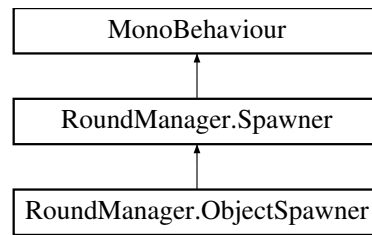
- Round System/Scripts/Demo Scripts/Event Examples/Lighting/LightingChangeOnNewRound.cs

5.34 RoundManager.ObjectSpawner Class Reference

Example class reacting to round event [ObjectSpawnRequestEvent](#).

Spawns preparation object when [ObjectSpawnRequestEvent](#) is raised.

Inheritance diagram for RoundManager.ObjectSpawner:



Additional Inherited Members

5.34.1 Detailed Description

Example class reacting to round event [ObjectSpawnRequestEvent](#).
Spawns preperation object when [ObjectSpawnRequestEvent](#) is raised.

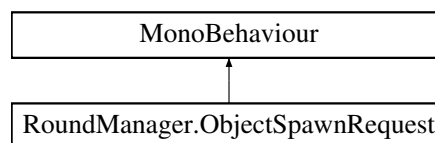
The documentation for this class was generated from the following file:

- Round System/Scripts/Demo Scripts/Event Examples/Spawn/ObjectSpawner.cs

5.35 RoundManager.ObjectSpawnRequest Class Reference

Template class. Use this as a template to react to a [ObjectSpawnRequestEvent](#). This event is raised when a object should be spawned. This is raised during the preperation stage of a round if the round has [Round::Preperation↔TimeObjects](#). Place your logic in the [OnObjectSpawnRequest](#) function.

Inheritance diagram for RoundManager.ObjectSpawnRequest:



Public Member Functions

- void [OnObjectSpawnRequest](#) ([ObjectSpawnRequestEvent](#) e)
React to [ObjectSpawnRequestEvent](#). Place logic to spawn preperation objects here.

5.35.1 Detailed Description

Template class. Use this as a template to react to a [ObjectSpawnRequestEvent](#). This event is raised when a object should be spawned. This is raised during the preperation stage of a round if the round has [Round::Preperation↔TimeObjects](#). Place your logic in the [OnObjectSpawnRequest](#) function.

5.35.2 Member Function Documentation

5.35.2.1 void RoundManager.ObjectSpawnRequest.OnObjectSpawnRequest ([ObjectSpawnRequestEvent](#) e)

React to [ObjectSpawnRequestEvent](#). Place logic to spawn preperation objects here.

Parameters

<i>e</i>	Event.
----------	--------

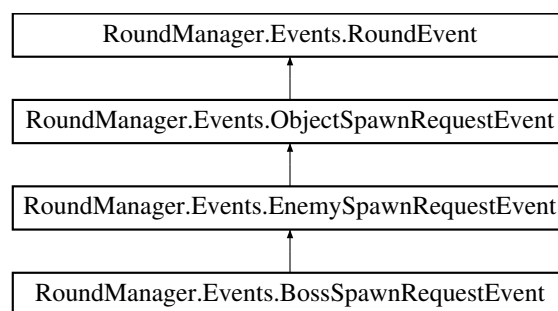
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Spawn Request Events/ObjectSpawnRequest.cs

5.36 RoundManager.Events.ObjectSpawnRequestEvent Class Reference

Raised when the an object should be spawned. See [ObjectSpawnRequest](#) template and [ObjectSpawner](#) for a concrete example of how to react to the event.

Inheritance diagram for RoundManager.Events.ObjectSpawnRequestEvent:



Public Member Functions

- **ObjectSpawnRequestEvent** ([Round](#) currentRound, GameObject objectPrefab)

Properties

- [Round](#) **CurrentRound** [get]
Gets the current round that raised the event.
- GameObject [ObjectPrefab](#) [get]
The prefab that should be instantiated.

5.36.1 Detailed Description

Raised when the an object should be spawned. See [ObjectSpawnRequest](#) template and [ObjectSpawner](#) for a concrete example of how to react to the event.

5.36.2 Property Documentation

5.36.2.1 Round RoundManager.Events.ObjectSpawnRequestEvent.CurrentRound [get]

Gets the current round that raised the event.

The current round.

5.36.2.2 GameObject RoundManager.Events.ObjectSpawnRequestEvent.ObjectPrefab [get]

The prefab that should be instantiated.

The object prefab.

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

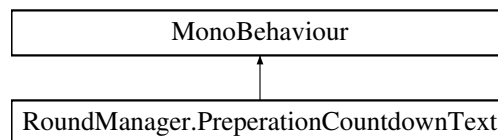
- Round System/Scripts/_Round System/Event System/Individual Events/Spawn Request Events/ObjectSpawnRequestEvent.cs

5.37 RoundManager.PreperationCountdownText Class Reference

Example class reacting to round event [PreperationStartEvent](#).

Shows round preperation countdown text on screen when [PreperationStartEvent](#) is raised.

Inheritance diagram for RoundManager.PreperationCountdownText:



Public Attributes

- string **PrePreperationTimeText**
- string **PostPreperationTimeText**

5.37.1 Detailed Description

Example class reacting to round event [PreperationStartEvent](#).

Shows round preperation countdown text on screen when [PreperationStartEvent](#) is raised.

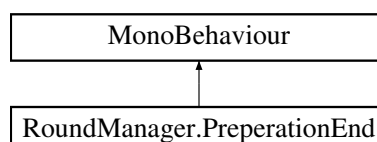
The documentation for this class was generated from the following file:

- Round System/Scripts/Demo Scripts/Event Examples/Text/PreperationCountdownText.cs

5.38 RoundManager.PreperationEnd Class Reference

Template class. Use this as a template to react to a [PreperationEndEvent](#). This event is raised when a rounds preperation stage has finished. Place your logic in the [OnPreperationEnd](#) function.

Inheritance diagram for RoundManager.PreperationEnd:



Public Member Functions

- void [OnPreperationEnd](#) ([PreperationEndEvent](#) e)
React to [PreperationEndEvent](#).

5.38.1 Detailed Description

Template class. Use this as a template to react to a [PreperationEndEvent](#). This event is raised when a rounds preperation stage has finished. Place your logic in the [OnPreperationEnd](#) function.

5.38.2 Member Function Documentation

5.38.2.1 void RoundManager.PreperationEnd.OnPreperationEnd ([PreperationEndEvent](#) e)

React to [PreperationEndEvent](#).

Parameters

<i>e</i>	Event.
----------	--------

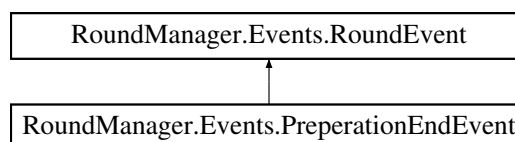
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Preperation Events/PreperationEnd.cs

5.39 RoundManager.Events.PreperationEndEvent Class Reference

Raised to signify that the current rounds preperation period has finished. See [PreperationEnd](#) for a template of how to react to this event.

Inheritance diagram for RoundManager.Events.PreperationEndEvent:



Public Member Functions

- **PreperationEndEvent** ([Round](#) currentRound)

Properties

- [Round CurrentRound](#) [get]
Gets the current round that raised this event.

5.39.1 Detailed Description

Raised to signify that the current rounds preperation period has finished. See [PreperationEnd](#) for a template of how to react to this event.

5.39.2 Property Documentation

5.39.2.1 Round RoundManager.Events.PreparationEndEvent.CurrentRound [get]

Gets the current round that raised this event.

The current round.

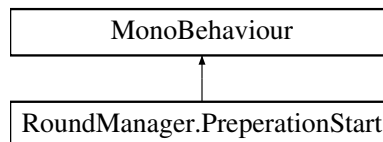
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/Individual Events/Preparation Events/PreparationEndEvent.cs

5.40 RoundManager.PreparationStart Class Reference

Template class. Use this as a template to react to a [PreparationStartEvent](#). This event is raised when a rounds preparation stage starts. Place your logic in the [OnPreparationStart](#) function.

Inheritance diagram for RoundManager.PreparationStart:



Public Member Functions

- void [OnPreparationStart](#) ([PreparationStartEvent](#) e)
React to [PreparationStartEvent](#).

5.40.1 Detailed Description

Template class. Use this as a template to react to a [PreparationStartEvent](#). This event is raised when a rounds preparation stage starts. Place your logic in the [OnPreparationStart](#) function.

5.40.2 Member Function Documentation

5.40.2.1 void RoundManager.PreparationStart.OnPreparationStart ([PreparationStartEvent](#) e)

React to [PreparationStartEvent](#).

Parameters

<i>e</i>	Event.
----------	--------

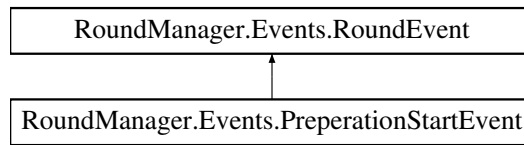
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Preparation Events/PreparationStart.cs

5.41 RoundManager.Events.PreparationStartEvent Class Reference

Raised to signify that the current rounds preparation period has started. See [PreparationStart](#) for a template of how to react to this event.

Inheritance diagram for RoundManager.Events.PreperationStartEvent:



Public Member Functions

- **PreperationStartEvent** ([Round](#) currentRound)

Properties

- [Round CurrentRound](#) [get]
Gets the current round that raised this event.

5.41.1 Detailed Description

Raised to signify that the current rounds preperation period has started. See [PreperationStart](#) for a template of how to react to this event.

5.41.2 Property Documentation

5.41.2.1 Round RoundManager.Events.PreperationStartEvent.CurrentRound [get]

Gets the current round that raised this event.

The current round.

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

- Round System/Scripts/_Round System/Event System/Individual Events/Preperation Events/PreperationStartEvent.cs

5.42 RoundManager.Round Class Reference

Responsible for storing, managing, and updating an array of Checkpoint. Handles round preperation time (if [HasPreperationTime](#)), preperation object spawning, and boss spawning (if [RoundHasBoss](#)).

Public Types

- enum **ProgressType** { **TimeUp**, **WhenTriggered**, **WaitForTrigger** }

Public Member Functions

- void **Enter** ([RoundManager](#) manager)
Invoked by [RoundManager](#). Not required to be called manually. Called at the beginning of the round. Raises [DestroyCurrentEnemiesRequestEvent](#) if [DestroyPreviousRoundEnemiesOnRoundStart](#) is true. Raises [RoundStartEvent](#).
- void **OnPreperationStart** ()

- Invoked by [RoundManager](#). Not required to be called manually. Called at the beginning of the round preparation stage. If the round has a preparation stage then [PreparationStartEvent](#) is raised and preparation objects spawned.
- void [OnPreparationOver](#) ()
Invoked by [RoundManager](#). Not required to be called manually. Called at the end of the rounds preparation stage. If the round has a preparation stage then [PreparationEndEvent](#) is raised.
 - void [Execute](#) ()
Invoked by [RoundManager](#) each time step that round is active. Not required to be called manually. Calls [Round↔Checkpoint::Execute](#) for current checkpoint. Increments checkpoint when current checkpoints time reaches zero.
 - void [Exit](#) ()
Invoked by [Round Manager](#). Not required to be called manually. Raises [RoundEndEvent](#) and if [DestroyEnemies↔OnRoundEnd](#) is true, raises [DestroyCurrentEnemiesRequestEvent](#).
 - void [TriggerRoundEnd](#) ()
Triggers the round to end immediately if the [RoundProgressType](#) is [WhenTriggered](#) or triggers the round to end when the round time has finished and the [RoundProgressType](#) is [WaitForTrigger](#).
 - void [BossKilled](#) ()
Invoked from [RoundBoss](#) when the boss is killed. Raises [BossKilledEvent](#).
 - IEnumerator [SpawnBoss](#) ()
Called if [RoundHasBoss](#). If [DestroyOtherEnemiesWhenBossCountdownBegins](#) is true then raises [DestroyCurrent↔EnemiesRequestEvent](#). If [BossCountdown](#) is greater than zero then a [BossCountdownBegunEvent](#) event is raised. [BossSpawnRequestEvent](#) is raised once the countdown reaches zero. If [DestroyOtherEnemiesWhenBossSpawns](#) is true then a [DestroyCurrentEnemiesRequestEvent](#) is raised.
 - void [RegisterEnemySpawned](#) ()
Registers an enemy spawned. Used when [OnlySpawnBossWhenAllEnemiesKilled](#) is true. Called by [RoundEnemy](#).
 - void [RegisterEnemyKilled](#) ()
Registers an enemy killed. Used when [OnlySpawnBossWhenAllEnemiesKilled](#) is true. Called by [RoundEnemy](#).

Public Attributes

- [RoundCheckpoint\[\]](#) [Checkpoints](#)
Array of checkpoints owned by this round.
- bool [HasPreparationTime](#) = true
A preparation time to allow for the player to prepare for the upcoming round.
- int [PreparationTime](#) = 10
Preparation time in seconds.
- bool [SpawnObjectsDuringPreparationTime](#) = true
Can spawn objects during preparation time.
- int [NumberOfObjectsToSpawn](#)
The number of objects to spawn. This number will be spawned as long as [PreparationTimeObjects](#) is greater than one.
- [RoundObjectSpawner\[\]](#) [PreparationTimeObjects](#)
The preparation time objects.
- bool [DestroyPreviousRoundEnemiesOnRoundStart](#)
If true, [DestroyCurrentEnemiesRequestEvent](#) is raised on round start.
- bool [DestroyEnemiesOnRoundEnd](#)
If true, [DestroyCurrentEnemiesRequestEvent](#) is raised on round end.
- bool [RoundHasBoss](#)
Each round can have a boss.
- GameObject [RoundBossPrefab](#)
The round boss prefab.
- float [BossCountdown](#)
Countdown until boss is spawned. Countdown begins at round end.
- bool [OnlySpawnBossWhenAllEnemiesKilled](#)

- *If true player must kill all of the rounds enemies before the boss countdown begins.*
 • bool [DestroyOtherEnemiesWhenBossCountdownBegins](#) = false
If true, DestroyCurrentEnemiesRequestEvent is raised when countdown to spawn boss begins. This is raised even if [BossCountdown](#) is 0.
- bool [DestroyOtherEnemiesWhenBossSpawns](#) = false
If true, DestroyCurrentEnemiesRequestEvent is raised when boss is spawned.
- ProgressType [RoundProgressType](#) = ProgressType.TimeUp
Defines how the round progresses to the next round.
TimeUp : round will end when the combined checkpoint time is up.
WhenTriggered : round will end immediately when [RoundManager::TriggerRoundEnd](#) is called.
WaitForTrigger : round will end when [RoundManager::TriggerRoundEnd](#) is called and the combined checkpoint time is up.

Properties

- [RoundCheckpoint CurrentCheckpoint](#) [get]
Gets the current checkpoint.
- bool [InPreperation](#) [get]
Gets a value indicating whether this [RoundManager.Round](#) is in the preperation stage.
- float [RoundTime](#) [get]
Returns the round time as float.
- int [RoundTimeInSeconds](#) [get]
Gets the round time in seconds.
- float [RoundTimeInMinutes](#) [get]
Gets the round time in minutes.
- string [RoundTimeInMinutesSeconds](#) [get]
Returns a string of format "00:00" representing the round time in minutes and seconds.
- float [RoundTotalTime](#) [get]
Gets the round total time, includes [PreperationTime](#).
- bool **RoundOver** [get]
- [RoundManager Manager](#) [get]

5.42.1 Detailed Description

Responsible for storing, managing, and updating an array of Checkpoint. Handles round preperation time (if [HasPreperationTime](#)), preperation object spawning, and boss spawning (if [RoundHasBoss](#)).

5.42.2 Member Function Documentation

5.42.2.1 void RoundManager.Round.BossKilled ()

Invoked from [RoundBoss](#) when the boss is killed. Raises [BossKilledEvent](#).

5.42.2.2 void RoundManager.Round.Enter (RoundManager manager)

Invoked by [RoundManager](#). Not required to be called manually. Called at the beginning of the round. Raises [DestroyCurrentEnemiesRequestEvent](#) if [DestroyPreviousRoundEnemiesOnRoundStart](#) is true. Raises [RoundStartEvent](#).

Parameters

<i>manager</i>	The rounds owner.
----------------	-------------------

5.42.2.3 void RoundManager.Round.Execute ()

Invoked by [RoundManager](#) each time step that round is active. Not required to be called manually. Calls [RoundCheckpoint::Execute](#) for current checkpoint. Increments checkpoint when current checkpoints time reaches zero.

5.42.2.4 void RoundManager.Round.Exit ()

Invoked by [Round Manager](#). Not required to be called manually. Raises [RoundEndEvent](#) and if [DestroyEnemiesOnRoundEnd](#) is true, raises [DestroyCurrentEnemiesRequestEvent](#).

5.42.2.5 void RoundManager.Round.OnPreparationOver ()

Invoked by [RoundManager](#). Not required to be called manually. Called at the end of the rounds preperation stage. If the round has a preperation stage then [PreperationEndEvent](#) is raised.

5.42.2.6 void RoundManager.Round.OnPreparationStart ()

Invoked by [RoundManager](#). Not required to be called manually. Called at the beginning of the round preperation stage. If the round has a preperation stage then [PreperationStartEvent](#). is raised and preperation objects spawned.

5.42.2.7 void RoundManager.Round.RegisterEnemyKilled ()

Registers an enemy killed. Used when [OnlySpawnBossWhenAllEnemiesKilled](#) is true. Called by [RoundEnemy](#).

5.42.2.8 void RoundManager.Round.RegisterEnemySpawned ()

Registers an enemy spawned. Used when [OnlySpawnBossWhenAllEnemiesKilled](#) is true. Called by [RoundEnemy](#).

5.42.2.9 IEnumerator RoundManager.Round.SpawnBoss ()

Called if [RoundHasBoss](#). If [DestroyOtherEnemiesWhenBossCountdownBegins](#) is true then raises [DestroyCurrentEnemiesRequestEvent](#). If [BossCountdown](#) is greater than zero then a [BossCountdownBegunEvent](#) event is raised. [BossSpawnRequestEvent](#) is raised once the countdown reaches zero. If [DestroyOtherEnemiesWhenBossSpawns](#) is true then a [DestroyCurrentEnemiesRequestEvent](#) is raised.

Returns

The boss.

5.42.2.10 void RoundManager.Round.TriggerRoundEnd ()

Triggers the round to end immediately if the [RoundProgressType](#) is WhenTriggered or triggers the round to end when the round time has finished and the [RoundProgressType](#) is WaitForTrigger .

5.42.3 Member Data Documentation

5.42.3.1 float RoundManager.Round.BossCountdown

Countdown until boss is spawned. Countdown begins at round end.

5.42.3.2 RoundCheckpoint [] RoundManager.Round.Checkpoints

Array of checkpoints owned by this round.

5.42.3.3 bool RoundManager.Round.DestroyEnemiesOnRoundEnd

If true, DestroyCurrentEnemiesRequestEvent is raised on round end.

5.42.3.4 bool RoundManager.Round.DestroyOtherEnemiesWhenBossCountdownBegins = false

If true, DestroyCurrentEnemiesRequestEvent is raised when countdown to spawn boss begins. This is raised even if [BossCountdown](#) is 0.

5.42.3.5 bool RoundManager.Round.DestroyOtherEnemiesWhenBossSpawns = false

If true, DestroyCurrentEnemiesRequestEvent is raised when boss is spawned.

5.42.3.6 bool RoundManager.Round.DestroyPreviousRoundEnemiesOnRoundStart

If true, DestroyCurrentEnemiesRequestEvent is raised on round start.

5.42.3.7 bool RoundManager.Round.HasPreperationTime = true

A preperation time to allow for the player to prepare for the upcoming round.

5.42.3.8 int RoundManager.Round.NumberOfObjectsToSpawn

The number of objects to spawn. This number will be spawned as long as [PreperationTimeObjects](#) is geater than one.

5.42.3.9 bool RoundManager.Round.OnlySpawnBossWhenAllEnemiesKilled

If true player must kill all of the rounds enemies before the boss countdown begins.

5.42.3.10 int RoundManager.Round.PreperationTime = 10

Preperation time in seconds.

5.42.3.11 RoundObjectSpawner [] RoundManager.Round.PreperationTimeObjects

The preperation time objects.

5.42.3.12 `GameObject RoundManager.Round.RoundBossPrefab`

The round boss prefab.

5.42.3.13 `bool RoundManager.Round.RoundHasBoss`

Each round can have a boss.

5.42.3.14 `ProgressType RoundManager.Round.RoundProgressType = ProgressType.TimeUp`

Defines how the round progresses to the next round.

TimeUp : round will end when the combined checkpoint time is up.

WhenTriggered : round will end immediately when [RoundManager::TriggerRoundEnd](#) is called.

WaitForTrigger : round will end when [RoundManager::TriggerRoundEnd](#) is called and the combined checkpoint time is up.

5.42.3.15 `bool RoundManager.Round.SpawnObjectsDuringPreperationTime = true`

Can spawn objects during preperation time.

5.42.4 Property Documentation

5.42.4.1 `RoundCheckpoint RoundManager.Round.CurrentCheckpoint` `[get]`

Gets the current checkpoint.

The current Checkpoint.

5.42.4.2 `bool RoundManager.Round.InPreperation` `[get]`

Gets a value indicating whether this [RoundManager.Round](#) is in the preperation stage.

`true` if in preperation; otherwise, `false`.

5.42.4.3 `float RoundManager.Round.RoundTime` `[get]`

Returns the round time as float.

The round time.

5.42.4.4 `float RoundManager.Round.RoundTimeInMinutes` `[get]`

Gets the round time in minutes.

The round time in minutes.

5.42.4.5 `string RoundManager.Round.RoundTimeInMinutesSeconds` `[get]`

Returns a string of format "00:00" representing the round time in minutes and seconds.

The round time in minutes seconds as formatted string.

5.42.4.6 `int RoundManager.Round.RoundTimeInSeconds` [get]

Gets the round time in seconds.

The round time in seconds.

5.42.4.7 `float RoundManager.Round.RoundTotalTime` [get]

Gets the round total time, includes [PreperationTime](#).

The round total time.

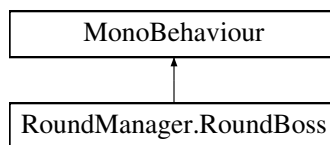
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Rounds/Round.cs

5.43 RoundManager.RoundBoss Class Reference

Attach to any gameobject that is classed as a rounds boss. When the object attached to this script is disabled the current round is notified.

Inheritance diagram for RoundManager.RoundBoss:



5.43.1 Detailed Description

Attach to any gameobject that is classed as a rounds boss. When the object attached to this script is disabled the current round is notified.

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

- Round System/Scripts/_Round System/Rounds/RoundBoss.cs

5.44 RoundManager.RoundCheckpoint Class Reference

Responsible for raising [EnemySpawnRequestEvent](#).

Public Member Functions

- void [Enter](#) ()

Invoked by [Round](#). Not required to be called manually. Called at the beginning of each checkpoint.

- void [Execute](#) ()

Invoked by [Round](#) each time step that checkpoint is active. Not required to be called manually. Raises [EnemySpawnRequestEvent](#) if:

Public Attributes

- float [CheckpointTime](#) = 10
Total checkpoint time.
- float [EnemySpawnChance](#) = 1
A number between 0 and 1. Every time an enemy is to be spawned, a random number between 0 and 1 is generated. If this random number is less than or equal to this then the enemy is spawned. Higher number results in a greater chance to spawn an enemy and 1 results in a spawned enemy every time.
- float [TimeBetweenEnemySpawns](#) = 0.5f
The time between enemy spawns.
- bool [LimitEnemyCount](#)
The number of enemies spawned during this checkpoint will be limited if this is true.
- int [MaxEnemies](#)
If [LimitEnemyCount](#) is true, this number will cap the number of enemies spawned.
- [RoundObjectSpawner\[\]](#) [RoundEnemies](#)
The enemies owned by this checkpoint.

Properties

- int [CheckpointTimeInSeconds](#) [get]
Gets the checkpoint time in seconds.
- float [CheckpointTimeInMinutes](#) [get]
Gets the checkpoint time in minutes.
- string [CheckpointTimeInMinutesSeconds](#) [get]
Returns a string of format "00:00" representing the checkpoint time in minutes and seconds.

5.44.1 Detailed Description

Responsible for raising [EnemySpawnRequestEvent](#).

5.44.2 Member Function Documentation

5.44.2.1 void RoundManager.RoundCheckpoint.Enter ()

Invoked by [Round](#). Not required to be called manually. Called at the beginning of each checkpoint.

5.44.2.2 void RoundManager.RoundCheckpoint.Execute ()

Invoked by [Round](#) each time step that checkpoint is active. Not required to be called manually. Raises [EnemySpawnRequestEvent](#) if:

- [RoundEnemies](#) count greater than 0.
- The current time between enemy spawns is \geq [TimeBetweenEnemySpawns](#).
- A random value between 0 and 1 is \leq [EnemySpawnChance](#).

5.44.3 Member Data Documentation

5.44.3.1 float RoundManager.RoundCheckpoint.CheckpointTime = 10

Total checkpoint time.

5.44.3.2 float RoundManager.RoundCheckpoint.EnemySpawnChance = 1

A number between 0 and 1. Every time an enemy is to be spawned, a random number between 0 and 1 is generated. If this random number is less than or equal to this then the enemy is spawned. Higher number results in a greater chance to spawn an enemy and 1 results in a spawned enemy every time.

5.44.3.3 bool RoundManager.RoundCheckpoint.LimitEnemyCount

The number of enemies spawned during this checkpoint will be limited if this is true.

5.44.3.4 int RoundManager.RoundCheckpoint.MaxEnemies

If [LimitEnemyCount](#) is true, this number will cap the number of enemies spawned.

5.44.3.5 RoundObjectSpawner [] RoundManager.RoundCheckpoint.RoundEnemies

The enemies owned by this checkpoint.

5.44.3.6 float RoundManager.RoundCheckpoint.TimeBetweenEnemySpawns = 0.5f

The time between enemy spawns.

5.44.4 Property Documentation

5.44.4.1 float RoundManager.RoundCheckpoint.CheckpointTimeInMinutes [get]

Gets the checkpoint time in minutes.

The checkpoint time in minutes.

5.44.4.2 string RoundManager.RoundCheckpoint.CheckpointTimeInMinutesSeconds [get]

Returns a string of format "00:00" representing the checkpoint time in minutes and seconds.

The checkpoint time in minutes seconds as formatted string.

5.44.4.3 int RoundManager.RoundCheckpoint.CheckpointTimeInSeconds [get]

Gets the checkpoint time in seconds.

The checkpoint time in seconds.

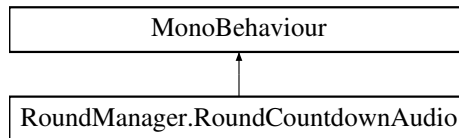
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Rounds/RoundCheckpoint.cs

5.45 RoundManager.RoundCountdownAudio Class Reference

Plays Audio Clip when [Round](#) has {SecondsLeftToBeginAudio} until it will begin.

Inheritance diagram for RoundManager.RoundCountdownAudio:



Public Attributes

- AudioClip **Audio**
- float **Volume** = 1
- int **SecondsLeftToBeginAudio** = 3

5.45.1 Detailed Description

Plays Audio Clip when [Round](#) has {SecondsLeftToBeginAudio} until it will begin.

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

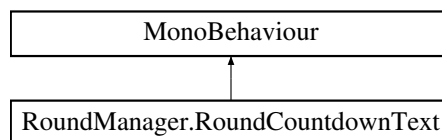
- Round System/Scripts/Demo Scripts/Event Examples/Audio/RoundCountdownAudio.cs

5.46 RoundManager.RoundCountdownText Class Reference

Example class reacting to round event [PreperationEndEvent](#).

Shows round countdown text on screen when [PreperationEndEvent](#) is raised and the round starts.

Inheritance diagram for RoundManager.RoundCountdownText:



Public Attributes

- string **PreRoundCountText**
- string **PostRoundText**

5.46.1 Detailed Description

Example class reacting to round event [PreperationEndEvent](#).

Shows round countdown text on screen when [PreperationEndEvent](#) is raised and the round starts.

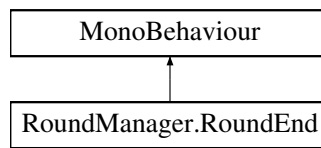
The documentation for this class was generated from the following file:

- Round System/Scripts/Demo Scripts/Event Examples/Text/RoundCountdownText.cs

5.47 RoundManager.RoundEnd Class Reference

Template class. Use this as a template to react to a [RoundEndEvent](#). This event is raised at the end of each round. Place your logic in the [OnRoundEnd](#) function.

Inheritance diagram for RoundManager.RoundEnd:



Public Member Functions

- void [OnRoundEnd](#) ([RoundEndEvent](#) e)
React to [RoundEndEvent](#).

5.47.1 Detailed Description

Template class. Use this as a template to react to a [RoundEndEvent](#). This event is raised at the end of each round. Place your logic in the [OnRoundEnd](#) function.

5.47.2 Member Function Documentation

5.47.2.1 void RoundManager.RoundEnd.OnRoundEnd (RoundEndEvent e)

React to [RoundEndEvent](#).

Parameters

<i>e</i>	Event.
----------	--------

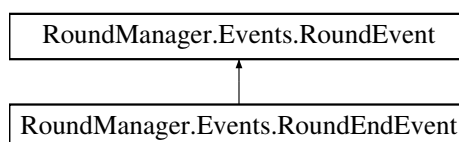
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Round Events/RoundEnd.cs

5.48 RoundManager.Events.RoundEndEvent Class Reference

Raised everytime a round ends. See [RoundEnd](#) for a template of how to react to this event and [AudioOnRoundEnd](#) for a concrete example.

Inheritance diagram for RoundManager.Events.RoundEndEvent:



Public Member Functions

- **RoundEndEvent** ([Round](#) currentRound, bool hasNextRound)

Properties

- [Round CurrentRound](#) [get]

Gets the current round that raised the event.

- bool [HasNextRound](#) [get]

Gets a value indicating whether this instance has next round.

5.48.1 Detailed Description

Raised everytime a round ends. See [RoundEnd](#) for a template of how to react to this event and [AudioOnRoundEnd](#) for a concrete example.

5.48.2 Property Documentation

5.48.2.1 Round RoundManager.Events.RoundEndEvent.CurrentRound [get]

Gets the current round that raised the event.

The current round.

5.48.2.2 bool RoundManager.Events.RoundEndEvent.HasNextRound [get]

Gets a value indicating whether this instance has next round.

true if this instance has next round; otherwise, false.

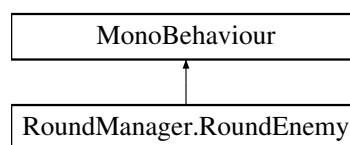
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/Individual Events/Round Events/RoundEndEvent.cs

5.49 RoundManager.RoundEnemy Class Reference

Attach to any gameobject that is classed as a round enemy. It registers with the current round when an enemy is spawned and killed. Used for when you want the player to have killed all enemies before the round boss is spawned.

Inheritance diagram for RoundManager.RoundEnemy:



5.49.1 Detailed Description

Attach to any gameobject that is classed as a round enemy. It registers with the current round when an enemy is spawned and killed. Used for when you want the player to have killed all enemies before the round boss is spawned.

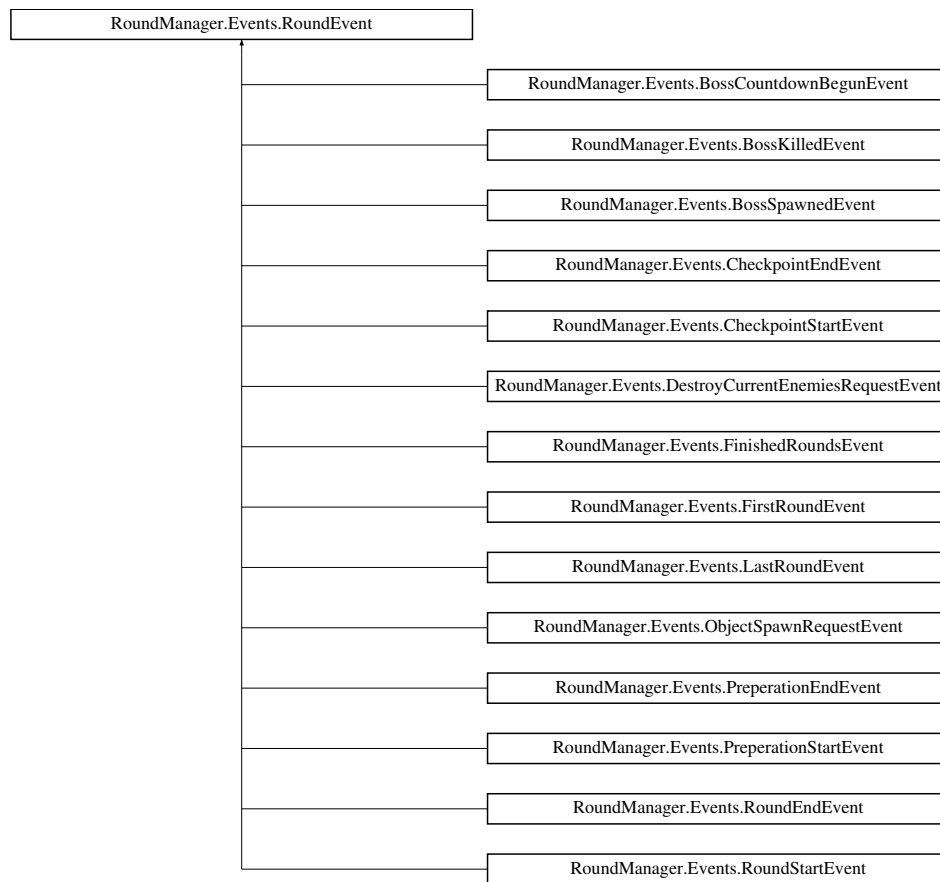
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Rounds/RoundEnemy.cs

5.50 RoundManager.Events.RoundEvent Class Reference

Base Event class. All events should derive from this class.

Inheritance diagram for RoundManager.Events.RoundEvent:



5.50.1 Detailed Description

Base Event class. All events should derive from this class.

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

- Round System/Scripts/_Round System/Event System/RoundEvent.cs

5.51 RoundManager.Events.RoundEvents Class Reference

Round events.

Public Member Functions

- delegate void **EventDelegate**< T > (T e)
- void **AddListener**< T > (EventDelegate< T > del)
Adds listener to be called when event T is called.
- void **RemoveListener**< T > (EventDelegate< T > del)
Removes listener for event T.
- void **Raise** (RoundEvent e)
Raise the specified event e.

Properties

- static [RoundEvents](#) **Instance** [get]

5.51.1 Detailed Description

[Round](#) events.

5.51.2 Member Function Documentation

5.51.2.1 void RoundManager.Events.RoundEvents.AddListener< T > (EventDelegate< T > del)

Adds listener to be called when event T is called.

Parameters

<i>del</i>	Del.
------------	------

Template Parameters

<i>T</i>	The 1st type parameter.
----------	-------------------------

Type Constraints

T : RoundEvent

5.51.2.2 void RoundManager.Events.RoundEvents.Raise (RoundEvent e)

Raise the specified event e.

Parameters

<i>e</i>	E.
----------	----

5.51.2.3 void RoundManager.Events.RoundEvents.RemoveListener< T > (EventDelegate< T > del)

Removes listener for event T.

Parameters

<i>del</i>	Del.
------------	------

Template Parameters

<i>T</i>	The 1st type parameter.
----------	-------------------------

Type Constraints

T : RoundEvent

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

- Round System/Scripts/_Round System/Event System/RoundEvents.cs

5.52 RoundManager.RoundLighting Class Reference

Used by [LightingChangeOnNewRound](#). Defines new light colour and intensity as well as the associated round number (defining when to change the lighting).

Public Attributes

- int [RoundNumber](#)
Round number to update lighting.
- Color **LightingColour**
- float **LightIntensity**

5.52.1 Detailed Description

Used by [LightingChangeOnNewRound](#). Defines new light colour and intensity as well as the associated round number (defining when to change the lighting).

5.52.2 Member Data Documentation

5.52.2.1 int RoundManager.RoundLighting.RoundNumber

[Round](#) number to update lighting.

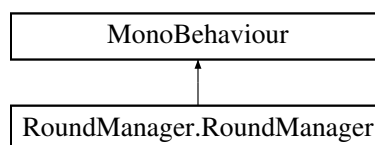
The documentation for this class was generated from the following file:

- Round System/Scripts/Demo Scripts/Event Examples/Lighting/RoundLighting.cs

5.53 RoundManager.RoundManager Class Reference

Main round manager class. Responsible for storing and updating rounds, and transitioning from one round to the next.

Inheritance diagram for RoundManager.RoundManager:



Public Member Functions

- delegate IEnumerator **Routine** ()
- void [Begin](#) ()
*Entry point into the round spawning system. Starts round management and begins first round. Raises [FirstRound](#)↔
[Event](#).*
- void [CoroutineCallback](#) (Routine callBack)
Helper method to start Coroutine as [Round](#) and [RoundCheckpoint](#) do not derive from Monobehaviour and therefore cannot start coroutines.
- bool [LastRound](#) ()
Returns true if last round.
- void [LogRoundMessage](#) (object message)
Helper method. Logs debug message if [ShowDebugMessages](#) is true.

Public Attributes

- bool [ShowDebugMessages](#)
Enables debug logging.
- [Round\[\] Rounds](#)
List of rounds.

Properties

- [Round CurrentRound](#) [get]
Returns the current active round.
- int [CurrentRoundIndex](#) [get]
Gets the index of the current round from [Rounds](#).
- static [RoundManager Instance](#) [get]
Gets an instance of [RoundManager](#). Class can be accessed from any script using: `RoundManager.Instance`;

5.53.1 Detailed Description

Main round manager class. Responsible for storing and updating rounds, and transitioning from one round to the next.

5.53.2 Member Function Documentation

5.53.2.1 void RoundManager.RoundManager.Begin ()

Entry point into the round spawning system. Starts round management and begins first round. Raises [FirstRoundEvent](#).

5.53.2.2 void RoundManager.RoundManager.CoroutineCallback (Routine *callBack*)

Helper method to start Coroutine as [Round](#) and [RoundCheckpoint](#) do not derive from MonoBehaviour and therefore cannot start coroutines.

Parameters

<i>callBack</i>	Call back.
-----------------	------------

5.53.2.3 bool RoundManager.RoundManager.LastRound ()

Returns true if last round.

Returns

`true`, if current round is last in list, `false` otherwise.

5.53.2.4 void RoundManager.RoundManager.LogRoundMessage (object *message*)

Helper method. Logs debug message if [ShowDebugMessages](#) is true.

Parameters

<i>message</i>	Message.
----------------	----------

5.53.3 Member Data Documentation

5.53.3.1 Round [] RoundManager.RoundManager.Rounds

List of rounds.

5.53.3.2 bool RoundManager.RoundManager.ShowDebugMessages

Enables debug logging.

5.53.4 Property Documentation

5.53.4.1 Round RoundManager.RoundManager.CurrentRound [get]

Returns the current active round.

The current round.

5.53.4.2 int RoundManager.RoundManager.CurrentRoundIndex [get]

Gets the index of the current round from [Rounds](#).

The index of the current round.

5.53.4.3 RoundManager RoundManager.RoundManager.Instance [static],[get]

Gets an instance of [RoundManager](#). Class can be accessed from any script using: RoundManager.Instance;

The instance.

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

- Round System/Scripts/_Round System/Rounds/RoundManager.cs

5.54 RoundObjectSpawner Class Reference

Round object spawner. Includes an object prefab and an associated spawn weight. Used by RoundCheckpoint::↵ RoundEnemies and Round::PreperationTimeObjects.

Public Attributes

- GameObject [Prefab](#)

The prefab.

- float [Weight](#)

The weight of the object prefab. Higher weights result in a greater chance that the [Prefab](#) will be spawned.

5.54.1 Detailed Description

Round object spawner. Includes an object prefab and an associated spawn weight. Used by RoundCheckpoint::↔RoundEnemies and Round::PreperationTimeObjects.

5.54.2 Member Data Documentation

5.54.2.1 GameObject RoundObjectSpawner.Prefab

The prefab.

5.54.2.2 float RoundObjectSpawner.Weight

The weight of the object prefab. Higher weights result in a greater chance that the [Prefab](#) will be spawned.

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

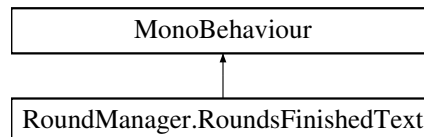
- Round System/Scripts/_Round System/Rounds/RoundObjectSpawner.cs

5.55 RoundManager.RoundsFinishedText Class Reference

Example class reacting to round event [FinishedRoundsEvent](#).

Shows #TextToShow on screen when [FinishedRoundsEvent](#) is raised i.e. all rounds have finished.

Inheritance diagram for RoundManager.RoundsFinishedText:



Public Attributes

- string **TextToShow**
- int **ShowForSeconds** = 1

5.55.1 Detailed Description

Example class reacting to round event [FinishedRoundsEvent](#).

Shows #TextToShow on screen when [FinishedRoundsEvent](#) is raised i.e. all rounds have finished.

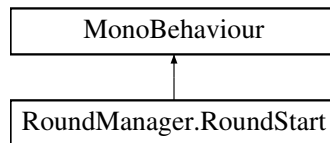
The documentation for this class was generated from the following file:

- Round System/Scripts/Demo Scripts/Event Examples/Text/RoundsFinishedText.cs

5.56 RoundManager.RoundStart Class Reference

Template class. Use this as a template to react to a [RoundStartEvent](#). This event is raised at the start of each round. Place your logic in the [OnRoundStart](#) function.

Inheritance diagram for RoundManager.RoundStart:



Public Member Functions

- void [OnRoundStart](#) ([RoundStartEvent](#) e)
React to [RoundStartEvent](#).

5.56.1 Detailed Description

Template class. Use this as a template to react to a [RoundStartEvent](#). This event is raised at the start of each round. Place your logic in the [OnRoundStart](#) function.

5.56.2 Member Function Documentation

5.56.2.1 void RoundManager.RoundStart.OnRoundStart (RoundStartEvent e)

React to [RoundStartEvent](#).

Parameters

<i>e</i>	Event.
----------	--------

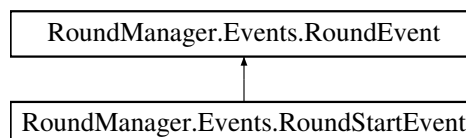
The documentation for this class was generated from the following file:

- Round System/Scripts/_Round System/Event System/_Templates/Round Events/RoundStart.cs

5.57 RoundManager.Events.RoundStartEvent Class Reference

Raised everytime a round starts. See [RoundStart](#) for a template of how to react to this event and [AudioOnRoundStart](#) for a concrete example.

Inheritance diagram for RoundManager.Events.RoundStartEvent:



Public Member Functions

- **RoundStartEvent** ([Round](#) currentRound)

Properties

- [Round Round](#) [get]
Gets the round that raised the event.

5.57.1 Detailed Description

Raised everytime a round starts. See [RoundStart](#) for a template of how to react to this event and [AudioOnRoundStart](#) for a concrete example.

5.57.2 Property Documentation

5.57.2.1 Round RoundManager.Events.RoundStartEvent.Round [get]

Gets the round that raised the event.

The round.

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

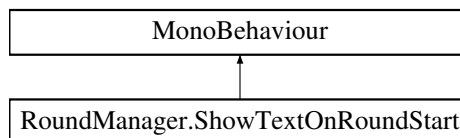
- Round System/Scripts/_Round System/Event System/Individual Events/Round Events/RoundStartEvent.cs

5.58 RoundManager.ShowTextOnRoundStart Class Reference

Example class reacting to round event [RoundStartEvent](#).

Shows round number on screen when [RoundStartEvent](#).

Inheritance diagram for RoundManager.ShowTextOnRoundStart:



Public Attributes

- string **PreRoundNumberText**
- string **PostRoundNumberText**
- int **ShowForSeconds** = 1

5.58.1 Detailed Description

Example class reacting to round event [RoundStartEvent](#).

Shows round number on screen when [RoundStartEvent](#).

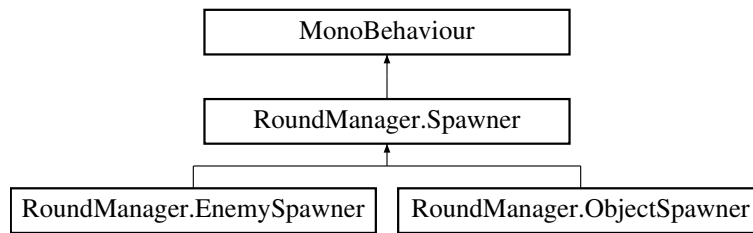
The documentation for this class was generated from the following file:

- Round System/Scripts/Demo Scripts/Event Examples/Text/ShowTextOnRoundStart.cs

5.59 RoundManager.Spawner Class Reference

Base Class for [ObjectSpawner](#) and [EnemySpawner](#). Provides functionality to instantiate object.

Inheritance diagram for RoundManager.Spawner:



Public Attributes

- [Environment Level](#)

Protected Member Functions

- `GameObject` **SpawnObject** (`GameObject` prefab, `List< Transform >` spawnTiles)

5.59.1 Detailed Description

Base Class for [ObjectSpawner](#) and [EnemySpawner](#). Provides functionality to instantiate object.

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

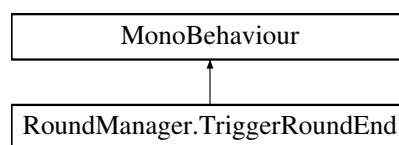
- Round System/Scripts/Demo Scripts/Event Examples/Spawn/Spawner.cs

5.60 RoundManager.TriggerRoundEnd Class Reference

Example class triggering the end of a round.

Invokes [Round::TriggerRoundEnd\(\)](#) when an object with tag 'Player' enters trigger. Can be used to end a round when player reaches specific point.

Inheritance diagram for RoundManager.TriggerRoundEnd:



5.60.1 Detailed Description

Example class triggering the end of a round.

Invokes [Round::TriggerRoundEnd\(\)](#) when an object with tag 'Player' enters trigger. Can be used to end a round when player reaches specific point.

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

- Round System/Scripts/Demo Scripts/Event Examples/TriggerRoundEnd.cs

5.61 RoundManager.Vector2i Struct Reference

Public Member Functions

- **Vector2i** (int x, int y)
- override string **ToString** ()

Public Attributes

- int **X**
- int **Y**

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

- Round System/Scripts/Demo Scripts/Environment/Vector2i.cs

Index

- AddListener< T >
 - RoundManager::Events::RoundEvents, [54](#)
- Audio
 - RoundManager::BackgroundAudioChangeOn↔NewRound, [15](#)
- Begin
 - RoundManager::RoundManager, [56](#)
- Boss
 - RoundManager::Events::BossSpawnedEvent, [20](#)
- BossCountdown
 - RoundManager::Round, [45](#)
- BossKilled
 - RoundManager::Round, [43](#)
- CheckpointTime
 - RoundManager::RoundCheckpoint, [48](#)
- CheckpointTimeInMinutes
 - RoundManager::RoundCheckpoint, [49](#)
- CheckpointTimeInMinutesSeconds
 - RoundManager::RoundCheckpoint, [49](#)
- CheckpointTimeInSeconds
 - RoundManager::RoundCheckpoint, [49](#)
- Checkpoints
 - RoundManager::Round, [45](#)
- CoroutineCallback
 - RoundManager::RoundManager, [56](#)
- CurrentCheckpoint
 - RoundManager::Round, [46](#)
- CurrentRound
 - RoundManager::Events::BossCountdownBegun↔Event, [16](#)
 - RoundManager::Events::BossKilledEvent, [18](#)
 - RoundManager::Events::CheckpointEndEvent, [23](#)
 - RoundManager::Events::CheckpointStartEvent, [25](#)
 - RoundManager::Events::DestroyCurrentEnemies↔RequestEvent, [26](#)
 - RoundManager::Events::FirstRoundEvent, [33](#)
 - RoundManager::Events::LastRoundEvent, [34](#)
 - RoundManager::Events::ObjectSpawnRequest↔Event, [37](#)
 - RoundManager::Events::PreperationEndEvent, [40](#)
 - RoundManager::Events::PreperationStartEvent, [41](#)
 - RoundManager::Events::RoundEndEvent, [52](#)
 - RoundManager::RoundManager, [57](#)
- CurrentRoundIndex
 - RoundManager::RoundManager, [57](#)
- DestroyEnemiesOnRoundEnd
 - RoundManager::Round, [45](#)
- DestroyOtherEnemiesWhenBossCountdownBegins
 - RoundManager::Round, [45](#)
- DestroyOtherEnemiesWhenBossSpawns
 - RoundManager::Round, [45](#)
- DestroyPreviousRoundEnemiesOnRoundStart
 - RoundManager::Round, [45](#)
- EnemySpawnArea
 - RoundManager::Environment, [30](#)
- EnemySpawnChance
 - RoundManager::RoundCheckpoint, [48](#)
- Enter
 - RoundManager::Round, [43](#)
 - RoundManager::RoundCheckpoint, [48](#)
- Execute
 - RoundManager::Round, [44](#)
 - RoundManager::RoundCheckpoint, [48](#)
- Exit
 - RoundManager::Round, [44](#)
- GenerateFloor
 - RoundManager::Environment, [30](#)
- HasNextCheckpoint
 - RoundManager::Events::CheckpointEndEvent, [23](#)
- HasNextRound
 - RoundManager::Events::RoundEndEvent, [52](#)
- HasPreperationTime
 - RoundManager::Round, [45](#)
- InPreperation
 - RoundManager::Round, [46](#)
- Instance
 - RoundManager::RoundManager, [57](#)
- LastRound
 - RoundManager::Events::FinishedRoundsEvent, [31](#)
 - RoundManager::RoundManager, [56](#)
- Lighting
 - RoundManager::LightingChangeOnNewRound, [35](#)
- LimitEnemyCount
 - RoundManager::RoundCheckpoint, [49](#)
- LogRoundMessage
 - RoundManager::RoundManager, [56](#)
- MaxEnemies
 - RoundManager::RoundCheckpoint, [49](#)
- NumberOfObjectsToSpawn
 - RoundManager::Round, [45](#)

- ObjectPrefab
 - RoundManager::Events::ObjectSpawnRequest↔
Event, [37](#)
- OnBossCountdownBegun
 - RoundManager::BossCountdownBegun, [15](#)
- OnBossKilled
 - RoundManager::BossKilled, [17](#)
- OnBossSpawned
 - RoundManager::BossSpawned, [19](#)
- OnCheckpointEnd
 - RoundManager::CheckpointEnd, [22](#)
- OnCheckpointStart
 - RoundManager::CheckpointStart, [24](#)
- OnDestroyCurrentEnemiesRequest
 - RoundManager::DestroyCurrentEnemiesRequest,
[25](#)
- OnFinishedRounds
 - RoundManager::FinishedRounds, [31](#)
- OnFirstRound
 - RoundManager::FirstRound, [32](#)
- OnObjectSpawnRequest
 - RoundManager::ObjectSpawnRequest, [36](#)
- OnPreparationOver
 - RoundManager::Round, [44](#)
- OnPreparationEnd
 - RoundManager::PreperationEnd, [39](#)
- OnPreparationStart
 - RoundManager::PreperationStart, [40](#)
 - RoundManager::Round, [44](#)
- OnRoundEnd
 - RoundManager::RoundEnd, [51](#)
- OnRoundStart
 - RoundManager::RoundStart, [59](#)
- OnlySpawnBossWhenAllEnemiesKilled
 - RoundManager::Round, [45](#)
- Prefab
 - RoundObjectSpawner, [58](#)
- PreperationTime
 - RoundManager::Round, [45](#)
- PreperationTimeObjects
 - RoundManager::Round, [45](#)
- Raise
 - RoundManager::Events::RoundEvents, [54](#)
- RegisterEnemyKilled
 - RoundManager::Round, [44](#)
- RegisterEnemySpawned
 - RoundManager::Round, [44](#)
- RemoveListener< T >
 - RoundManager::Events::RoundEvents, [54](#)
- Round
 - RoundManager::Events::RoundStartEvent, [60](#)
- RoundBossPrefab
 - RoundManager::Round, [45](#)
- RoundEnemies
 - RoundManager::RoundCheckpoint, [49](#)
- RoundHasBoss
 - RoundManager::Round, [46](#)
- RoundManager, [9](#)
 - RoundManager.AudioOnRoundEnd, [13](#)
 - RoundManager.AudioOnRoundStart, [13](#)
 - RoundManager.BackgroundAudio, [14](#)
 - RoundManager.BackgroundAudioChangeOnNew↔
Round, [14](#)
 - RoundManager.BossCountdownBegun, [15](#)
 - RoundManager.BossCountdownText, [16](#)
 - RoundManager.BossKilled, [17](#)
 - RoundManager.BossSpawnRequest, [20](#)
 - RoundManager.BossSpawned, [18](#)
 - RoundManager.CheckpointEnd, [21](#)
 - RoundManager.CheckpointStart, [24](#)
 - RoundManager.DestroyCurrentEnemiesRequest, [25](#)
 - RoundManager.Director, [26](#)
 - RoundManager.EnemyHealth, [27](#)
 - RoundManager.EnemySpawnRequest, [28](#)
 - RoundManager.EnemySpawner, [28](#)
 - RoundManager.Environment, [29](#)
 - RoundManager.Events, [11](#)
 - RoundManager.Events.BossCountdownBegunEvent, [16](#)
 - RoundManager.Events.BossKilledEvent, [18](#)
 - RoundManager.Events.BossSpawnRequestEvent, [20](#)
 - RoundManager.Events.BossSpawnedEvent, [19](#)
 - RoundManager.Events.CheckpointEndEvent, [23](#)
 - RoundManager.Events.CheckpointStartEvent, [24](#)
 - RoundManager.Events.DestroyCurrentEnemies↔
RequestEvent, [26](#)
 - RoundManager.Events.EnemySpawnRequestEvent, [29](#)
 - RoundManager.Events.FinishedRoundsEvent, [31](#)
 - RoundManager.Events.FirstRoundEvent, [32](#)
 - RoundManager.Events.LastRoundEvent, [33](#)
 - RoundManager.Events.ObjectSpawnRequestEvent, [37](#)
 - RoundManager.Events.PreperationEndEvent, [39](#)
 - RoundManager.Events.PreperationStartEvent, [40](#)
 - RoundManager.Events.RoundEndEvent, [51](#)
 - RoundManager.Events.RoundEvent, [52](#)
 - RoundManager.Events.RoundEvents, [53](#)
 - RoundManager.Events.RoundStartEvent, [59](#)
 - RoundManager.FinishedRounds, [30](#)
 - RoundManager.FirstRound, [32](#)
 - RoundManager.LastRound, [33](#)
 - RoundManager.LerpOverTime, [34](#)
 - RoundManager.LightingChangeOnNewRound, [35](#)
 - RoundManager.ObjectSpawnRequest, [36](#)
 - RoundManager.ObjectSpawner, [35](#)
 - RoundManager.PreperationCountdownText, [38](#)
 - RoundManager.PreperationEnd, [38](#)
 - RoundManager.PreperationStart, [40](#)
 - RoundManager.Round, [41](#)
 - RoundManager.RoundBoss, [47](#)
 - RoundManager.RoundCheckpoint, [47](#)
 - RoundManager.RoundCountdownAudio, [49](#)
 - RoundManager.RoundCountdownText, [50](#)
 - RoundManager.RoundEnd, [50](#)
 - RoundManager.RoundEnemy, [52](#)
 - RoundManager.RoundLighting, [55](#)
 - RoundManager.RoundManager, [55](#)

- RoundManager.RoundStart, [58](#)
- RoundManager.RoundsFinishedText, [58](#)
- RoundManager.ShowTextOnRoundStart, [60](#)
- RoundManager.Spawner, [60](#)
- RoundManager.TriggerRoundEnd, [61](#)
- RoundManager.Vector2i, [61](#)
- RoundManager::BackgroundAudioChangeOnNewRound↔
 - Audio, [15](#)
- RoundManager::BossCountdownBegun
 - OnBossCountdownBegun, [15](#)
- RoundManager::BossKilled
 - OnBossKilled, [17](#)
- RoundManager::BossSpawned
 - OnBossSpawned, [19](#)
- RoundManager::CheckpointEnd
 - OnCheckpointEnd, [22](#)
- RoundManager::CheckpointStart
 - OnCheckpointStart, [24](#)
- RoundManager::DestroyCurrentEnemiesRequest
 - OnDestroyCurrentEnemiesRequest, [25](#)
- RoundManager::Environment
 - EnemySpawnArea, [30](#)
 - GenerateFloor, [30](#)
- RoundManager::Events::BossCountdownBegunEvent
 - CurrentRound, [16](#)
- RoundManager::Events::BossKilledEvent
 - CurrentRound, [18](#)
- RoundManager::Events::BossSpawnedEvent
 - Boss, [20](#)
- RoundManager::Events::CheckpointEndEvent
 - CurrentRound, [23](#)
 - HasNextCheckpoint, [23](#)
- RoundManager::Events::CheckpointStartEvent
 - CurrentRound, [25](#)
- RoundManager::Events::DestroyCurrentEnemies↔RequestEvent
 - CurrentRound, [26](#)
- RoundManager::Events::FinishedRoundsEvent
 - LastRound, [31](#)
- RoundManager::Events::FirstRoundEvent
 - CurrentRound, [33](#)
- RoundManager::Events::LastRoundEvent
 - CurrentRound, [34](#)
- RoundManager::Events::ObjectSpawnRequestEvent
 - CurrentRound, [37](#)
 - ObjectPrefab, [37](#)
- RoundManager::Events::PreperationEndEvent
 - CurrentRound, [40](#)
- RoundManager::Events::PreperationStartEvent
 - CurrentRound, [41](#)
- RoundManager::Events::RoundEndEvent
 - CurrentRound, [52](#)
 - HasNextRound, [52](#)
- RoundManager::Events::RoundEvents
 - AddListener< T >, [54](#)
 - Raise, [54](#)
 - RemoveListener< T >, [54](#)
- RoundManager::Events::RoundStartEvent
 - Round, [60](#)
- RoundManager::FinishedRounds
 - OnFinishedRounds, [31](#)
- RoundManager::FirstRound
 - OnFirstRound, [32](#)
- RoundManager::LerpOverTime
 - Value, [35](#)
- RoundManager::LightingChangeOnNewRound
 - Lighting, [35](#)
- RoundManager::ObjectSpawnRequest
 - OnObjectSpawnRequest, [36](#)
- RoundManager::PreperationEnd
 - OnPreperationEnd, [39](#)
- RoundManager::PreperationStart
 - OnPreperationStart, [40](#)
- RoundManager::Round
 - BossCountdown, [45](#)
 - BossKilled, [43](#)
 - Checkpoints, [45](#)
 - CurrentCheckpoint, [46](#)
 - DestroyEnemiesOnRoundEnd, [45](#)
 - DestroyOtherEnemiesWhenBossCountdown↔Begins, [45](#)
 - DestroyOtherEnemiesWhenBossSpawns, [45](#)
 - DestroyPreviousRoundEnemiesOnRoundStart, [45](#)
 - Enter, [43](#)
 - Execute, [44](#)
 - Exit, [44](#)
 - HasPreperationTime, [45](#)
 - InPreperation, [46](#)
 - NumberOfObjectsToSpawn, [45](#)
 - OnPreparationOver, [44](#)
 - OnPreperationStart, [44](#)
 - OnlySpawnBossWhenAllEnemiesKilled, [45](#)
 - PreperationTime, [45](#)
 - PreperationTimeObjects, [45](#)
 - RegisterEnemyKilled, [44](#)
 - RegisterEnemySpawned, [44](#)
 - RoundBossPrefab, [45](#)
 - RoundHasBoss, [46](#)
 - RoundProgressType, [46](#)
 - RoundTime, [46](#)
 - RoundTimeInMinutes, [46](#)
 - RoundTimeInMinutesSeconds, [46](#)
 - RoundTimeInSeconds, [46](#)
 - RoundTotalTime, [47](#)
 - SpawnBoss, [44](#)
 - SpawnObjectsDuringPreperaionTime, [46](#)
 - TriggerRoundEnd, [44](#)
- RoundManager::RoundCheckpoint
 - CheckpointTime, [48](#)
 - CheckpointTimeInMinutes, [49](#)
 - CheckpointTimeInMinutesSeconds, [49](#)
 - CheckpointTimeInSeconds, [49](#)
 - EnemySpawnChance, [48](#)
 - Enter, [48](#)
 - Execute, [48](#)

- LimitEnemyCount, [49](#)
- MaxEnemies, [49](#)
- RoundEnemies, [49](#)
- TimeBetweenEnemySpawns, [49](#)
- RoundManager::RoundEnd
 - OnRoundEnd, [51](#)
- RoundManager::RoundLighting
 - RoundNumber, [55](#)
- RoundManager::RoundManager
 - Begin, [56](#)
 - CoroutineCallback, [56](#)
 - CurrentRound, [57](#)
 - CurrentRoundIndex, [57](#)
 - Instance, [57](#)
 - LastRound, [56](#)
 - LogRoundMessage, [56](#)
 - Rounds, [57](#)
 - ShowDebugMessages, [57](#)
- RoundManager::RoundStart
 - OnRoundStart, [59](#)
- RoundNumber
 - RoundManager::RoundLighting, [55](#)
- RoundObjectSpawner, [57](#)
 - Prefab, [58](#)
 - Weight, [58](#)
- RoundProgressType
 - RoundManager::Round, [46](#)
- RoundTime
 - RoundManager::Round, [46](#)
- RoundTimeInMinutes
 - RoundManager::Round, [46](#)
- RoundTimeInMinutesSeconds
 - RoundManager::Round, [46](#)
- RoundTimeInSeconds
 - RoundManager::Round, [46](#)
- RoundTotalTime
 - RoundManager::Round, [47](#)
- Rounds
 - RoundManager::RoundManager, [57](#)
- ShowDebugMessages
 - RoundManager::RoundManager, [57](#)
- SpawnBoss
 - RoundManager::Round, [44](#)
- SpawnObjectsDuringPreparationTime
 - RoundManager::Round, [46](#)
- TimeBetweenEnemySpawns
 - RoundManager::RoundCheckpoint, [49](#)
- TriggerRoundEnd
 - RoundManager::Round, [44](#)
- Value
 - RoundManager::LerpOverTime, [35](#)
- Weight
 - RoundObjectSpawner, [58](#)