

Kill Mo' Chickens

Generated by Doxygen 1.9.1

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 Class Documentation	5
3.1 DifficultyCalculator Class Reference	5
3.1.1 Detailed Description	6
3.1.2 Constructor & Destructor Documentation	6
3.1.2.1 DifficultyCalculator()	6
3.1.3 Member Function Documentation	6
3.1.3.1 CalculateSpawnOrder()	6
3.1.3.2 getEnemyRanks()	7
3.1.3.3 getSpawnAmount()	7
3.1.4 Member Data Documentation	7
3.1.4.1 difficultyTable	7
3.2 DifficultyCalculatorFactory Class Reference	7
3.2.1 Detailed Description	8
3.2.2 Member Function Documentation	8
3.2.2.1 CreateCalculator()	8
3.3 DifficultyTable Class Reference	8
3.3.1 Detailed Description	8
3.3.2 Member Data Documentation	9
3.3.2.1 EnemyRanks	9
3.3.2.2 RoundDifficultyValue	9
3.4 EasyDifficultyCalculator Class Reference	9
3.4.1 Detailed Description	10
3.4.2 Member Function Documentation	10
3.4.2.1 CalculateSpawnOrder()	10
3.5 HardDifficultyCalculator Class Reference	10
3.5.1 Detailed Description	11
3.5.2 Member Function Documentation	11
3.5.2.1 CalculateSpawnOrder()	11
3.6 MediumDifficultyCalculator Class Reference	11
3.6.1 Detailed Description	12
3.7 RoundManager Class Reference	12
3.7.1 Detailed Description	13
3.7.2 Member Function Documentation	13
3.7.2.1 _Process()	13
3.7.2.2 _Ready()	14
3.7.2.3 cleanLevel()	14
3.7.2.4 GameLostEventHandler()	14

3.7.2.5 GameWonEventHandler()	14
3.7.2.6 HandleEnemyDiesSignal()	14
3.7.2.7 HandleEnemyFinishedSignal()	15
3.7.2.8 HandleEnemySplit()	15
3.7.2.9 loadLevel()	15
3.7.2.10 spawnEnemy()	16
3.7.2.11 startRound()	16
3.7.3 Member Data Documentation	16
3.7.3.1 difficultyCalculator	16
3.7.3.2 levelData	16
3.7.3.3 liveEnemies	16
3.7.3.4 roundRunning	16
3.7.3.5 spawnQueue	17
3.8 SpawnOrder Class Reference	17
3.8.1 Detailed Description	17
3.8.2 Constructor & Destructor Documentation	17
3.8.2.1 SpawnOrder()	17
3.8.3 Member Data Documentation	18
3.8.3.1 Enemy	18
3.8.3.2 spawnDelay	18
Index	19

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

DifficultyCalculator	5
EasyDifficultyCalculator	9
HardDifficultyCalculator	10
MediumDifficultyCalculator	11
DifficultyCalculatorFactory	7
DifficultyTable	8
RoundManager	12
SpawnOrder	17

Chapter 2

Class Index

2.1 Class List

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

DifficultyCalculator	The Base class for a difficulty calculator. Cannot construct directly, instead use the DifficultyCalculatorFactory.CreateCalculator method	5
DifficultyCalculatorFactory	Factory Function for getting a DifficultyCalculator class object	7
DifficultyTable	A Resource defining the difficult table	8
EasyDifficultyCalculator	Difficulty Calculator For the Easy Difficulty	9
HardDifficultyCalculator	Difficulty Calculator For the Hard Difficulty	10
MediumDifficultyCalculator	Difficulty Calculator For the Medium Difficulty	11
RoundManager	The Round Manager for Processing/tracking round data across the rounds of a level. The class is also in charge of spawning the enemies, tracking round health and money, signaling when a round ends, and signaling when the player wins or loses	12
SpawnOrder	A Single spawn order element for controlling what and when to spawn an enemy	17

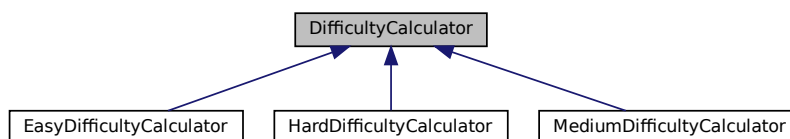
Chapter 3

Class Documentation

3.1 DifficultyCalculator Class Reference

The Base class for a difficulty calculator. Cannot construct directly, instead use the `DifficultyCalculatorFactory.Create` method.

Inheritance diagram for DifficultyCalculator:



Public Member Functions

- virtual List< [SpawnOrder](#) > [CalculateSpawnOrder](#) (int roundNumber)
Base Virtual function for Calculating the Spawn Order. This function returns a list of [SpawnOrder](#) objects that should be spawned for the corresponding round

Protected Member Functions

- int [getSpawnAmount](#) (int cost, ref int levelValue)
Getter function for getting the amount of a particular enemy rank to spawn
- Godot.Collections.Array< int > [getEnemyRanks](#) ()
Gets the available ranks of enemies that can be spawned.

Protected Attributes

- [DifficultyTable](#) `difficultyTable`
The current difficulty table that is loaded.

Package Functions

- [DifficultyCalculator](#) ([DifficultyTable](#) difficultyTable)

Internal/Private Constructor

3.1.1 Detailed Description

The Base class for a difficulty calculator. Cannot construct directly, instead use the [DifficultyCalculatorFactory.Create](#) method.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 DifficultyCalculator()

```
DifficultyCalculator.DifficultyCalculator (
    DifficultyTable difficultyTable ) [package]
```

Internal/Private Constructor

Parameters

<i>difficultyTable</i>	
------------------------	--

3.1.3 Member Function Documentation

3.1.3.1 CalculateSpawnOrder()

```
virtual List<SpawnOrder> DifficultyCalculator.CalculateSpawnOrder (
    int roundNumber ) [virtual]
```

Base Virtual function for Calculating the Spawn Order. This function returns a list of [SpawnOrder](#) objects that should be spawned for the corresponding round

Parameters

<i>roundNumber</i>	The Current Round Number
--------------------	--------------------------

Returns

Reimplemented in [HardDifficultyCalculator](#), and [EasyDifficultyCalculator](#).

3.1.3.2 getEnemyRanks()

```
Godot.Collections.Array<int> DifficultyCalculator.getEnemyRanks ( ) [protected]
```

Gets the available ranks of enemies that can be spawned.

Returns

The Available ranks to spawn

3.1.3.3 getSpawnAmount()

```
int DifficultyCalculator.getSpawnAmount (
    int cost,
    ref int levelValue ) [protected]
```

Getter function for getting the amount of a particular enemy rank to spawn

Parameters

<i>cost</i>	The cost of the enemy to spawn.
<i>levelValue</i>	The total amount of enemy "Value" for the level.

Returns

The amount of enemies to spawn.

3.1.4 Member Data Documentation

3.1.4.1 difficultyTable

```
DifficultyTable DifficultyCalculator.difficultyTable [protected]
```

The current difficulty table that is loaded.

3.2 DifficultyCalculatorFactory Class Reference

Factory Function for getting a [DifficultyCalculator](#) class object

Static Public Member Functions

- static [DifficultyCalculator](#) [CreateCalculator](#) ([DifficultyTable](#) difficultyTable, Difficulty difficulty)
Get a new Difficulty Calculator class obj based on the Difficulty

3.2.1 Detailed Description

Factory Function for getting a [DifficultyCalculator](#) class object

3.2.2 Member Function Documentation

3.2.2.1 CreateCalculator()

```
static DifficultyCalculator DifficultyCalculatorFactory.CreateCalculator (
    DifficultyTable difficultyTable,
    Difficulty difficulty ) [static]
```

Get a new Difficulty Calculator class obj based on the Difficulty

Parameters

<i>difficultyTable</i>	DifficultyTable to be used by the calculator.
<i>difficulty</i>	The Difficulty to be used for the calculator

Returns

A Difficulty Calculator of the passed difficulty

3.3 DifficultyTable Class Reference

A Resource defining the difficult table

Inherits Resource.

Public Attributes

- Godot.Collections.Array< int > [EnemyRanks](#)
- int[] [RoundDifficultyValue](#)
An Array of the total amount of enemy "value" to spawn each round

3.3.1 Detailed Description

A Resource defining the difficult table

3.3.2 Member Data Documentation

3.3.2.1 EnemyRanks

```
Godot.Collections.Array<int> DifficultyTable.EnemyRanks
```

3.3.2.2 RoundDifficultyValue

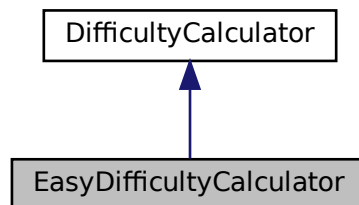
```
int [] DifficultyTable.RoundDifficultyValue
```

An Array of the total amount of enemy "value" to spawn each round

3.4 EasyDifficultyCalculator Class Reference

Difficulty Calculator For the Easy Difficulty

Inheritance diagram for EasyDifficultyCalculator:



Public Member Functions

- override List< [SpawnOrder](#) > [CalculateSpawnOrder](#) (int roundNumber)

Function for Calculating the Spawn Order. This function returns a list of [SpawnOrder](#) objects that should be spawned for the corresponding round. Easy Mode lowers the enemy 'spawn budget' by 20%

Package Functions

- [EasyDifficultyCalculator](#) ([DifficultyTable](#) difficultyTable)

Additional Inherited Members

3.4.1 Detailed Description

Difficulty Calculator For the Easy Difficulty

3.4.2 Member Function Documentation

3.4.2.1 CalculateSpawnOrder()

```
override List<SpawnOrder> EasyDifficultyCalculator.CalculateSpawnOrder (
    int roundNumber ) [virtual]
```

Function for Calculating the Spawn Order. This function returns a list of [SpawnOrder](#) objects that should be spawned for the corresponding round. Easy Mode lowers the enemy 'spawn budget' by 20%

Parameters

<i>roundNumber</i>	The Current Round Number
--------------------	--------------------------

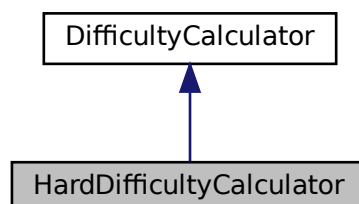
Returns

Reimplemented from [DifficultyCalculator](#).

3.5 HardDifficultyCalculator Class Reference

Difficulty Calculator For the Hard Difficulty

Inheritance diagram for HardDifficultyCalculator:



Public Member Functions

- override List< [SpawnOrder](#) > [CalculateSpawnOrder](#) (int roundNumber)

Function for Calculating the Spawn Order. This function returns a list of [SpawnOrder](#) objects that should be spawned for the corresponding round. Hard Mode raises the enemy 'spawn budget' by 50%

Package Functions

- [HardDifficultyCalculator](#) ([DifficultyTable](#) difficultyTable)

Additional Inherited Members

3.5.1 Detailed Description

Difficulty Calculator For the Hard Difficulty

3.5.2 Member Function Documentation

3.5.2.1 CalculateSpawnOrder()

```
override List<SpawnOrder> HardDifficultyCalculator.CalculateSpawnOrder (
    int roundNumber ) [virtual]
```

Function for Calculating the Spawn Order. This function returns a list of [SpawnOrder](#) objects that should be spawned for the corresponding round. Hard Mode raises the enemy 'spawn budget' by 50%

Parameters

<i>roundNumber</i>	The Current Round Number
--------------------	--------------------------

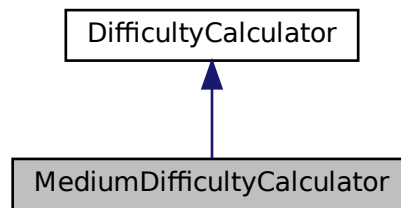
Returns

Reimplemented from [DifficultyCalculator](#).

3.6 MediumDifficultyCalculator Class Reference

Difficulty Calculator For the Medium Difficulty

Inheritance diagram for MediumDifficultyCalculator:



Package Functions

- **MediumDifficultyCalculator** ([DifficultyTable difficultyTable](#))

Additional Inherited Members

3.6.1 Detailed Description

Difficulty Calculator For the Medium Difficulty

3.7 RoundManager Class Reference

The Round Manager for Processing/tracking round data across the rounds of a level. The class is also in charge of spawning the enemies, tracking round health and money, signaling when a round ends, and signaling when the player wins or loses

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()
Called when the object enters the scene tree. Sets up queue objects
- void [loadLevel](#) (Level [levelData](#), int difficulty)
Loads a level from a `Level` object and creates a difficulty table.
- void [startRound](#) ()
Called to start a round. This method starts the next round, getting the [SpawnOrder](#)'s from the [DifficultyCalculator](#), and begins the spawning timer.
- override void [_Process](#) (double delta)
Godot Process function called every engine cycle Processes spawning enemies when the spawn timer is up. Also handles emitting the end game signals.
- delegate void [GameLostEventHandler](#) ()
Signal to emit when the game is lost.
- delegate void [GameWonEventHandler](#) ()
Signal to emit when the game is Won.

Public Attributes

- List< [SpawnOrder](#) > [spawnQueue](#)
Holds the queue of enemy's to spawn
- List< [BaseChicken](#) > [liveEnemies](#)
Holds the currently living enemies
- bool [roundRunning](#) = false
Indicates if a round is running.

Private Member Functions

- void [spawnEnemy](#) ()
Internal method used for spawning the next enemy in the spawn queue
- void [HandleEnemyDiesSignal](#) ([BaseChicken](#) enemy)
Enemy Death Signal Handler.
- void [HandleEnemyFinishedSignal](#) ([BaseChicken](#) enemy)
Enemy Finished path Signal Handler.
- void [HandleEnemySplit](#) ([BaseChicken](#) enemy)
Enemy Split Event Handler.
- void [cleanLevel](#) ()
Clears spawn queue and all loaded enemies. Called before exporting the level data to save

Private Attributes

- [DifficultyCalculator](#) [difficultyCalculator](#)
The [DifficultyCalculator](#) for the selected difficulty
- Level [levelData](#)
The Loaded Level object
- double **currentTime**
- double **nextSpawnTime**

3.7.1 Detailed Description

The Round Manager for Processing/tracking round data across the rounds of a level. The class is also in charge of spawning the enemies, tracking round health and money, signaling when a round ends, and signaling when the player wins or loses

3.7.2 Member Function Documentation

3.7.2.1 `_Process()`

```
override void RoundManager._Process (
    double delta )
```

Godot Process function called every engine cycle Processes spawning enemies when the spawn timer is up. Also handles emitting the end game signals.

Parameters

<i>delta</i>	
--------------	--

3.7.2.2 _Ready()

```
override void RoundManager._Ready ( )
```

Called when the object enters the scene tree. Sets up queue objects

3.7.2.3 cleanLevel()

```
void RoundManager.cleanLevel ( ) [private]
```

Clears spawn queue and all loaded enemies. Called before exporting the level data to save

3.7.2.4 GameLostEventHandler()

```
delegate void RoundManager.GameLostEventHandler ( )
```

Signal to emit when the game is lost.

3.7.2.5 GameWonEventHandler()

```
delegate void RoundManager.GameWonEventHandler ( )
```

Signal to emit when the game is Won.

3.7.2.6 HandleEnemyDiesSignal()

```
void RoundManager.HandleEnemyDiesSignal (
    BaseChicken enemy ) [private]
```

Enemy Death Signal Handler.

Parameters

<i>enemy</i>	The associated enemy.
--------------	-----------------------

3.7.2.7 HandleEnemyFinishedSignal()

```
void RoundManager.HandleEnemyFinishedSignal (
    BaseChicken enemy ) [private]
```

Enemy Finished path Signal Handler.

Parameters

<i>enemy</i>	The associated enemy.
--------------	-----------------------

3.7.2.8 HandleEnemySplit()

```
void RoundManager.HandleEnemySplit (
    BaseChicken enemy ) [private]
```

Enemy Split Event Handler.

Parameters

<i>enemy</i>	The associated enemy.
--------------	-----------------------

3.7.2.9 loadLevel()

```
void RoundManager.loadLevel (
    Level levelData,
    int difficulty )
```

Loads a level from a `Level` object and creates a difficulty table.

Parameters

<i>levelData</i>	The current level data.
<i>difficulty</i>	Indicates the Difficulty to use.

3.7.2.10 spawnEnemy()

```
void RoundManager.spawnEnemy ( ) [private]
```

Internal method used for spawning the next enemy in the spawn queue

3.7.2.11 startRound()

```
void RoundManager.startRound ( )
```

Called to start a round. This method starts the next round, getting the [SpawnOrder](#)'s from the [DifficultyCalculator](#), and begins the spawning timer.

3.7.3 Member Data Documentation

3.7.3.1 difficultyCalculator

```
DifficultyCalculator RoundManager.difficultyCalculator [private]
```

The [DifficultyCalculator](#) for the selected difficulty

3.7.3.2 levelData

```
Level RoundManager.levelData [private]
```

The Loaded Level object

3.7.3.3 liveEnemies

```
List<BaseChicken> RoundManager.liveEnemies
```

Holds the currently living enemies

3.7.3.4 roundRunning

```
bool RoundManager.roundRunning = false
```

Indicates if a round is running.

3.7.3.5 spawnQueue

```
List<SpawnOrder> RoundManager.spawnQueue
```

Holds the queue of enemy's to spawn

3.8 SpawnOrder Class Reference

A Single spawn order element for controlling what and when to spawn an enemy

Inherits Node.

Public Member Functions

- [SpawnOrder](#) (Chicken.BaseChicken enemy, int [spawnDelay](#))
Creates A spawn Order for a passed type that implements IEnemyType

Public Attributes

- BaseChicken [Enemy](#)
The Enemy class use for the spawn.
- int [spawnDelay](#)
The delay in milliseconds to wait before spawning this enemy

3.8.1 Detailed Description

A Single spawn order element for controlling what and when to spawn an enemy

3.8.2 Constructor & Destructor Documentation

3.8.2.1 SpawnOrder()

```
SpawnOrder.SpawnOrder (
    Chicken.BaseChicken enemy,
    int spawnDelay )
```

Creates A spawn Order for a passed type that implements IEnemyType

Parameters

<i>enemy</i>	
<i>spawnDelay</i>	

3.8.3 Member Data Documentation

3.8.3.1 Enemy

`BaseChicken SpawnOrder.Enemy`

The Enemy class use for the spawn.

3.8.3.2 spawnDelay

`int SpawnOrder.spawnDelay`

The delay in milliseconds to wait before spawning this enemy

Index

- [_Process](#)
 - [RoundManager, 13](#)
 - [_Ready](#)
 - [RoundManager, 14](#)
- [CalculateSpawnOrder](#)
 - [DifficultyCalculator, 6](#)
 - [EasyDifficultyCalculator, 10](#)
 - [HardDifficultyCalculator, 11](#)
- [cleanLevel](#)
 - [RoundManager, 14](#)
- [CreateCalculator](#)
 - [DifficultyCalculatorFactory, 8](#)
- [DifficultyCalculator, 5](#)
 - [CalculateSpawnOrder, 6](#)
 - [DifficultyCalculator, 6](#)
 - [difficultyTable, 7](#)
 - [getEnemyRanks, 7](#)
 - [getSpawnAmount, 7](#)
- [difficultyCalculator](#)
 - [RoundManager, 16](#)
- [DifficultyCalculatorFactory, 7](#)
 - [CreateCalculator, 8](#)
- [DifficultyTable, 8](#)
 - [EnemyRanks, 9](#)
 - [RoundDifficultyValue, 9](#)
- [difficultyTable](#)
 - [DifficultyCalculator, 7](#)
- [EasyDifficultyCalculator, 9](#)
 - [CalculateSpawnOrder, 10](#)
- [Enemy](#)
 - [SpawnOrder, 18](#)
- [EnemyRanks](#)
 - [DifficultyTable, 9](#)
- [GameLostEventHandler](#)
 - [RoundManager, 14](#)
- [GameWonEventHandler](#)
 - [RoundManager, 14](#)
- [getEnemyRanks](#)
 - [DifficultyCalculator, 7](#)
- [getSpawnAmount](#)
 - [DifficultyCalculator, 7](#)
- [HandleEnemyDiesSignal](#)
 - [RoundManager, 14](#)
- [HandleEnemyFinishedSignal](#)
 - [RoundManager, 15](#)
- [HandleEnemySplit](#)
 - [RoundManager, 15](#)
- [HardDifficultyCalculator, 10](#)
 - [CalculateSpawnOrder, 11](#)
- [levelData](#)
 - [RoundManager, 16](#)
- [liveEnemies](#)
 - [RoundManager, 16](#)
- [loadLevel](#)
 - [RoundManager, 15](#)
- [MediumDifficultyCalculator, 11](#)
- [RoundDifficultyValue](#)
 - [DifficultyTable, 9](#)
- [RoundManager, 12](#)
 - [_Process, 13](#)
 - [_Ready, 14](#)
 - [cleanLevel, 14](#)
 - [difficultyCalculator, 16](#)
 - [GameLostEventHandler, 14](#)
 - [GameWonEventHandler, 14](#)
 - [HandleEnemyDiesSignal, 14](#)
 - [HandleEnemyFinishedSignal, 15](#)
 - [HandleEnemySplit, 15](#)
 - [levelData, 16](#)
 - [liveEnemies, 16](#)
 - [loadLevel, 15](#)
 - [roundRunning, 16](#)
 - [spawnEnemy, 15](#)
 - [spawnQueue, 16](#)
 - [startRound, 16](#)
- [roundRunning](#)
 - [RoundManager, 16](#)
- [spawnDelay](#)
 - [SpawnOrder, 18](#)
- [spawnEnemy](#)
 - [RoundManager, 15](#)
- [SpawnOrder, 17](#)
 - [Enemy, 18](#)
 - [spawnDelay, 18](#)
 - [SpawnOrder, 17](#)
- [spawnQueue](#)
 - [RoundManager, 16](#)
- [startRound](#)
 - [RoundManager, 16](#)