Underbounders

Generated by Doxygen 1.9.6

Chapter 1

Namespace Index

1.1 Package List

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 BarsElements Namespace Reference

Classes

- class Bar

  Status bar ui element.

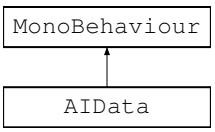
Chapter 6

Class Documentation

6.1 AIData Class Reference

Targets and obstacles positions container.

Inheritance diagram for AIData:



Public Member Functions

- int GetTargetsCount ()

  Provides number of targets with null handling.

Properties

- List< Transform > Targets `[get, set]`

  Current list of targets.
- Collider2D[ ] Obstacles `[get, set]`

  Current obstacles in monster vicinity.
- Transform CurrentTarget `[get, set]`

  Current target.

6.1.1 Detailed Description

Targets and obstacles positions container.

6.1.2 Member Function Documentation

6.1.2.1 GetTargetsCount()

```
int AIData.GetTargetsCount ( )
```

Provides number of targets with null handling.

Returns

Number of targets available

6.1.3 Property Documentation

6.1.3.1 CurrentTarget

```
Transform AIData.CurrentTarget  [get], [set]
```

Current target.

6.1.3.2 Obstacles

```
Collider2D [] AIData.Obstacles  [get], [set]
```

Current obstacles in monster vicinity.

6.1.3.3 Targets

```
List<Transform> AIData.Targets  [get], [set]
```

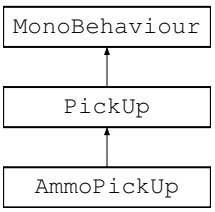
Current list of targets.

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

- Assets/Scripts/MonsterScripts/SlimesAI/AIData.cs

6.2 AmmoPickUp Class Reference

Inheritance diagram for AmmoPickUp:



Public Member Functions

- override bool OnPickUp (PlayerStatsController playerStatsController)

  Picks up ammunition if it player can pick it up.

- virtual bool OnPickUp (PlayerStatsController playerStatsController)

  Virtual method describing picking up the item.

6.2.1 Member Function Documentation

6.2.1.1 OnPickUp()

```
override bool AmmoPickUp.OnPickUp (
            PlayerStatsController playerStatsController )  [virtual]
```

Picks up ammunition if it player can pick it up.

Parameters

playerStatsController	Player stats controller of player, that picks up item

Returns

`true` if the item was used, `false` if the item wasn't used

Reimplemented from PickUp.

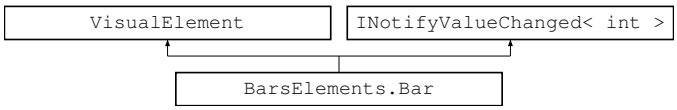
The documentation for this class was generated from the following file:

- Assets/Scripts/ItemsScripts/AmmoPickUp.cs

6.3 BarsElements.Bar Class Reference

Status bar ui element.

Inheritance diagram for BarsElements.Bar:



Classes

- class UxmlFactory

  Element as Uxml factory.
- class UxmlTraits

  Exclusive element traits factory for uxml editor and events.

Public Member Functions

- void SetValueWithoutNotify (int newValue)

  Sets value without special notification.

Properties

- int value `[get, set]`
- string spriteBaseName `[get, set]`
- string spriteName `[get, set]`

6.3.1 Detailed Description

Status bar ui element.

6.3.2 Member Function Documentation

6.3.2.1 SetValueWithoutNotify()

```
void BarsElements.Bar.SetValueWithoutNotify (
            int newValue )
```

Sets value without special notification.

Parameters

newValue	New value to set

6.3.3 Property Documentation

6.3.3.1 spriteBaseName

```
string BarsElements.Bar.spriteBaseName  [get], [set]
```

6.3.3.2 spriteName

```
string BarsElements.Bar.spriteName  [get], [set]
```

6.3.3.3 value

```
int BarsElements.Bar.value  [get], [set]
```

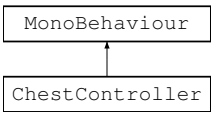
The documentation for this class was generated from the following file:

- Assets/Scripts/CustomUIElelements/Bar.cs

6.4 ChestController Class Reference

Controlles chest behaviours.

Inheritance diagram for ChestController:



Public Member Functions

- void OpenChest ()

  Open chests with item spawn.
- void SpawnItem ()

  Spawns items from chests.
- void ChangeSprite ()

  Changes sprite and opens the chest.

Public Attributes

- int x

  Position of the chest in the room.

Properties

- bool _isChestOpen `[get, set]`

  Determines if chest is open.
- int int y chestPos `[get, set]`

6.4.1 Detailed Description

Controlles chest behaviours.

6.4.2 Member Function Documentation

6.4.2.1 ChangeSprite()

```
void ChestController.ChangeSprite ( )
```

Changes sprite and opens the chest.

6.4.2.2 OpenChest()

```
void ChestController.OpenChest ( )
```

Open chests with item spawn.

6.4.2.3 SpawnItem()

```
void ChestController.SpawnItem ( )
```

Spawns items from chests.

6.4.3 Member Data Documentation

6.4.3.1 x

```
int ChestController.x
```

Position of the chest in the room.

6.4.4 Property Documentation

6.4.4.1 _isChestOpen

```
bool ChestController._isChestOpen  [get], [set]
```

Determines if chest is open.

6.4.4.2 chestPos

```
int int y ChestController.chestPos  [get], [set]
```

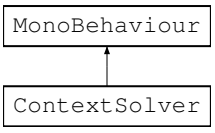
The documentation for this class was generated from the following file:

- Assets/Scripts/MapScripts/Obstacles/ChestController.cs

6.5 ContextSolver Class Reference

Calculates movement using steering directions.

Inheritance diagram for ContextSolver:



Public Member Functions

- Vector2 GetDirectionToMove (List< SteeringBehaviour > behaviours, AIData aiData)

  Moves slime in desired direction.

6.5.1 Detailed Description

Calculates movement using steering directions.

6.5.2 Member Function Documentation

6.5.2.1 GetDirectionToMove()

```
Vector2 ContextSolver.GetDirectionToMove (
            List< SteeringBehaviour > behaviours,
            AIData aiData )
```

Moves slime in desired direction.

Parameters

behaviours	List of monster steering behaviours
aiData	AI data containing obstacles and targets

Returns

Calculated movement direction

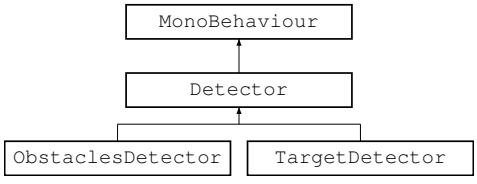
The documentation for this class was generated from the following file:

- Assets/Scripts/MonsterScripts/SlimesAI/ContextSolver.cs

6.6 Detector Class Reference

Detectors abstraction layer.

Inheritance diagram for Detector:



Public Member Functions

- abstract void Detect (AIData ai)

  Detects targets in vasinity.

6.6.1 Detailed Description

Detectors abstraction layer.

6.6.2 Member Function Documentation

6.6.2.1 Detect()

```
abstract void Detector.Detect (
            AIData ai )  [pure virtual]
```

Detects targets in vasinity.

Parameters

ai	Data holding curent targets and rest of the data

Implemented in ObstaclesDetector, and TargetDetector.

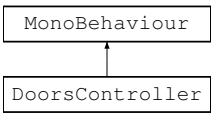
The documentation for this class was generated from the following file:

- Assets/Scripts/MonsterScripts/SlimesAI/Detector.cs

6.7 DoorsController Class Reference

Doors behaviour controller.

Inheritance diagram for DoorsController:



Properties

- Direction dir `[get, set]`

  Door facing direction.

6.7.1 Detailed Description

Doors behaviour controller.

6.7.2 Property Documentation

6.7.2.1 dir

```
Direction DoorsController.dir  [get], [set]
```

Door facing direction.

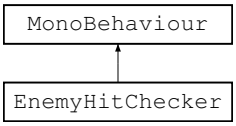
The documentation for this class was generated from the following file:

- Assets/Scripts/MapScripts/DoorsController.cs

6.8 EnemyHitChecker Class Reference

Checks if player was hit.

Inheritance diagram for EnemyHitChecker:



6.8.1 Detailed Description

Checks if player was hit.

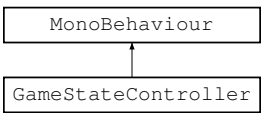
The documentation for this class was generated from the following file:

- Assets/Scripts/PlayerScripts/EnemyHitChecker.cs

6.9 GameStateController Class Reference

Class managing state of the game, responsible for pausing and etc.

Inheritance diagram for GameStateController:



Public Member Functions

- void NewGame ()

  Initialize new game coroutine, on click.
- void LoadMainMenu ()

  Loads main menu.
- void SwitchRooms (Direction dir)

  Exits rooms and enters new one in given direction.
- void LoadPauseMenu ()

  Loads pause menu.
- void UnloadPauseMenu ()

  UnloadsPauseMenu.
- void OnGameEnd ()

  Ends game on death of player.
- void SaveGame ()

  Saves game on click.
- void LoadGame ()

  Loads game on click.
- void Exit ()

  Exits game on click.
- void MapClear ()

  Clears map from any unnecessary object.

Properties

- static GameStateController instance `[get, set]`

  Game state controlle instance.
- bool isPaused `[get, set]`

  Determines if the game is paused.
- bool isSwitchingRoom `[get, set]`

  Determines if the game is in process of swithing the rooms.
- Room[ ][ ] rooms `[get, set]`

  Rooms matrix.
- Room currentRoom `[get, set]`

  Current room.

6.9.1 Detailed Description

Class managing state of the game, responsible for pausing and etc.

6.9.2 Member Function Documentation

6.9.2.1 Exit()

```
void GameStateController.Exit ( )
```

Exits game on click.

6.9.2.2 LoadGame()

```
void GameStateController.LoadGame ( )
```

Loads game on click.

6.9.2.3 LoadMainMenu()

```
void GameStateController.LoadMainMenu ( )
```

Loads main menu.

6.9.2.4 LoadPauseMenu()

```
void GameStateController.LoadPauseMenu ( )
```

Loads pause menu.

6.9.2.5 MapClear()

```
void GameStateController.MapClear ( )
```

Clears map from any unnecessary object.

6.9.2.6 NewGame()

```
void GameStateController.NewGame ( )
```

Initialize new game coroutine, on click.

6.9.2.7 OnGameEnd()

```
void GameStateController.OnGameEnd ( )
```

Ends game on death of player.

6.9.2.8 SaveGame()

```
void GameStateController.SaveGame ( )
```

Saves game on click.

6.9.2.9 SwitchRooms()

```
void GameStateController.SwitchRooms (
            Direction dir )
```

Exits rooms and enters new one in given direction.

Parameters

dir	Direction that player go towards

6.9.2.10 UnloadPauseMenu()

```
void GameStateController.UnloadPauseMenu ( )
```

UnloadsPauseMenu.

6.9.3 Property Documentation

6.9.3.1 currentRoom

```
Room GameStateController.currentRoom  [get], [set]
```

Current room.

6.9.3.2 instance

```
GameStateController GameStateController.instance  [static], [get], [set]
```

Game state controlle instance.

6.9.3.3 isPaused

```
bool GameStateController.isPaused  [get], [set]
```

Determines if the game is paused.

6.9.3.4 isSwitchingRoom

```
bool GameStateController.isSwitchingRoom  [get], [set]
```

Determines if the game is in process of swithing the rooms.

6.9.3.5 rooms

```
Room [][] GameStateController.rooms  [get], [set]
```

Rooms matrix.

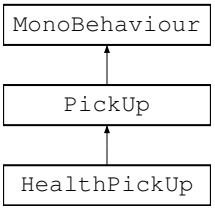
The documentation for this class was generated from the following file:

- Assets/Scripts/GameStateController.cs

6.10 HealthPickUp Class Reference

Pick up healing item behaviour.

Inheritance diagram for HealthPickUp:



Public Member Functions

- override bool OnPickUp (PlayerStatsController playerStatsController)

  Picks up healing item if it player can pick it up.

- virtual bool OnPickUp (PlayerStatsController playerStatsController)

  Virtual method describing picking up the item.

6.10.1 Detailed Description

Pick up healing item behaviour.

6.10.2 Member Function Documentation

6.10.2.1 OnPickUp()

```
override bool HealthPickUp.OnPickUp (
            PlayerStatsController playerStatsController )  [virtual]
```

Picks up healing item if it player can pick it up.

Parameters

playerStatsController	Player stats controller of player, that picks up item

Returns

`true` if the item was used, `false` if the item wasn't used

Reimplemented from PickUp.

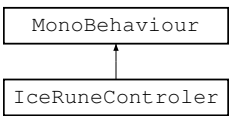
The documentation for this class was generated from the following file:

- Assets/Scripts/ItemsScripts/HealthPickUp.cs

6.11 IceRuneControler Class Reference

Ice rune behaviour controller.

Inheritance diagram for IceRuneControler:



6.11.1 Detailed Description

Ice rune behaviour controller.

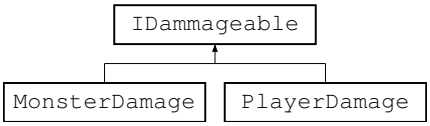
The documentation for this class was generated from the following file:

- Assets/Scripts/MapScripts/Obstacles/IceRuneControler.cs

6.12 IDammageable Interface Reference

Damage handling interface.

Inheritance diagram for IDammageable:



Public Member Functions

- void OnHit (float damage, Vector2 knockback)

  Deals certain amount of damage with knockback.
- void OnHit (float damage)

  Deals certain amount of damage.

Properties

- bool Targetable `[get, set]`

6.12.1 Detailed Description

Damage handling interface.

6.12.2 Member Function Documentation

6.12.2.1 OnHit() `[1/2]`

```
void IDammageable.OnHit (
            float damage )
```

Deals certain amount of damage.

Parameters

damage	Damamge dealt

Implemented in MonsterDamage, and PlayerDamage.

6.12.2.2 OnHit() [2/2]

```
void IDammageable.OnHit (
            float damage,
            Vector2 knockback )
```

Deals certain amount of damage with knockback.

Parameters

damage	Damamge dealt
knockback	Knockback to be added to position

Implemented in MonsterDamage, and PlayerDamage.

6.12.3 Property Documentation

6.12.3.1 Targetable

```
bool IDammageable.Targetable  [get], [set]
```

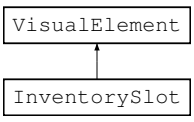
Implemented in MonsterDamage, and PlayerDamage.

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

- Assets/Scripts/CommonIntefaces/IDammageable.cs

6.13 InventorySlot Class Reference

Inheritance diagram for InventorySlot:



Public Member Functions

- InventorySlot ()

Public Attributes

- Image Icon
- string ItemGuid = ""

6.13.1 Constructor & Destructor Documentation

6.13.1.1 InventorySlot()

```
InventorySlot.InventorySlot ( )
```

6.13.2 Member Data Documentation

6.13.2.1 Icon

```
Image InventorySlot.Icon
```

6.13.2.2 ItemGuid

```
string InventorySlot.ItemGuid = ""
```

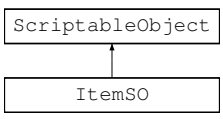
The documentation for this class was generated from the following file:

- Assets/Resources/UI/InventorySlot.cs

6.14 ItemSO Class Reference

Unused items Scriptable object, left for future.

Inheritance diagram for ItemSO:



Public Attributes

- int attackStat

  Additional attack value of the item.
- int attackSpeedStat

  Additional attack speed value of the item.
- int speedStat

  Additional speed value of the item.
- int defenceStat

  Scrapped defence stat.
- Sprite sprite

  Item sprite.
- string blessing

  Unused name of the item kind.

6.14.1 Detailed Description

Unused items Scriptable object, left for future.

6.14.2 Member Data Documentation

6.14.2.1 attackSpeedStat

```
int ItemSO.attackSpeedStat
```

Additional attack speed value of the item.

6.14.2.2 attackStat

```
int ItemSO.attackStat
```

Additional attack value of the item.

6.14.2.3 blessing

```
string ItemSO.blessing
```

Unused name of the item kind.

6.14.2.4 defenceStat

```
int ItemSO.defenceStat
```

Scrapped defence stat.

6.14.2.5 speedStat

```
int ItemSO.speedStat
```

Additional speed value of the item.

6.14.2.6 sprite

```
Sprite ItemSO.sprite
```

Item sprite.

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

- Assets/Scripts/PlayerScripts/ItemSO.cs

6.15 ItemSpawnRate Class Reference

Spawn possibility container.

Public Attributes

- GameObject item
  Item to spawn from chest.
- float chance
  Possibility of item spawning.

6.15.1 Detailed Description

Spawn possibility container.

6.15.2 Member Data Documentation

6.15.2.1 chance

```
float ItemSpawnRate.chance
```

Possibility of item spawning.

6.15.2.2 item

```
GameObject ItemSpawnRate.item
```

Item to spawn from chest.

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

- Assets/Scripts/MapScripts/Obstacles/ItemSpawnRate.cs

6.16 ITileKind< T > Interface Template Reference

Interface allowing the class to be used in map matrix class.

Public Member Functions

- Dictionary< T, float > FilterTiles (Dictionary< T, float > tileTofilter, Direction dir)
  It filters tiles in certain neighbourhood.

6.16.1 Detailed Description

Interface allowing the class to be used in map matrix class.

Template Parameters

T	Tile type

6.16.2 Member Function Documentation

6.16.2.1 FilterTiles()

```
Dictionary< T, float > ITileKind< T >.FilterTiles (
            Dictionary< T, float > tileTofilter,
            Direction dir )
```

It filters tiles in certain neighbourhood.

Parameters

tileTofilter	Tiles possibilities of neighbouring tile
dir	Defines direction distincting restrictions of the tile

Returns

Filtered tiles posibilities

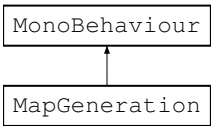
The documentation for this interface was generated from the following file:

- Assets/Scripts/MapScripts/ITileKind.cs

6.17 MapGeneration Class Reference

Controlls room generation.

Inheritance diagram for MapGeneration:



Public Member Functions

- void ClearMap ()

  Clears map from all of the tiles.
- void GenerateRoom (int seed, int x, int y)

  Main world generation function.

Public Attributes

- int x

  Size of the room.
- int y
- int seed = 23

  Ramdom generator seed for the room.

Static Public Attributes

- static MapGeneration instance

  Room generation instance of singleton.

Properties

- bool isDone [get, set]

  Determins if the room generation is done.
- List<(int x, int y)> isChestOpen [get, set]

  List of chests opened.
- bool[] IsDoorOnSide [get, set]

  Determines on which side the doors should be.

6.17.1 Detailed Description

Controlls room generation.

6.17.2 Member Function Documentation

6.17.2.1 ClearMap()

```
void MapGeneration.ClearMap ( )
```

Clears map from all of the tiles.

6.17.2.2 GenerateRoom()

```
void MapGeneration.GenerateRoom (
            int seed,
            int x,
            int y )
```

Main world generation function.

Parameters

seed	Room random generation seed
x	Room number of collumns
y	Room number of rows

6.17.3 Member Data Documentation

6.17.3.1 instance

```
MapGeneration MapGeneration.instance  [static]
```

Room generation instance of singleton.

6.17.3.2 seed

```
int MapGeneration.seed = 23
```

Ramdom generator seed for the room.

6.17.3.3 x

```
int MapGeneration.x
```

Size of the room.

6.17.3.4 y

```
int MapGeneration.y
```

6.17.4 Property Documentation

6.17.4.1 isChestOpen

```
List<(int x, int y)> MapGeneration.isChestOpen  [get], [set]
```

List of chests opened.

6.17.4.2 isDone

```
bool MapGeneration.isDone  [get], [set]
```

Determins if the room generation is done.

6.17.4.3 IsDoorOnSide

```
bool [] MapGeneration.IsDoorOnSide  [get], [set]
```

Determines on which side the doors should be.

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

- Assets/Scripts/MapScripts/MapGeneration.cs

6.18 MapMatrix< T > Class Template Reference

Matrix using wave function collapse to be resolved.

Public Member Functions

- MapMatrix (int x, int y, IEnumerable< T > intializator)

  This constructor creates matrix with equal number of equally propable possibilities.
- MapMatrix (int x, int y, List< Dictionary< T, float > > initialValues)

  This constructor creates matrix with unequal primary state.
- int PickRandomTile ()
- void PickTileValue (int i, int j)

  Randomly chooses particullar tiles possibility as its only value.
- List< Dictionary< T, float > > GetLowestCountList ()

  Searches for lowest, greater than 0, enthropy tiles from matrix.
- void RemoveImpossiblePairs ()

  Removes impossible candidats per tile from matrix.
- bool AreAllTilesSet ()

  Checks if there is enthropy equal 0 or contradiction occurs on every tile of the whole matrix.
- T GetTile (int i, int j)

  Returns particular tile from matrix.
- void ResolveMatrix ()

  Resolves matrix with wave function collapse algorythm.

Public Attributes

- int

  Picks random tile from map that has lowest, greater than 0, enthropy.

6.18.1 Detailed Description

Matrix using wave function collapse to be resolved.

Template Parameters

T	Tile reference type of possibility group, it has to implement ITileKind<T>

Type Constraints

T **:** ***ITileKind*****<*****T*****>**

6.18.2 Constructor & Destructor Documentation

6.18.2.1 MapMatrix() [1/2]

```
MapMatrix< T >.MapMatrix (
            int x,
            int y,
            IEnumerable< T > intializator )
```

This constructor creates matrix with equal number of equally propable possibilities.

Parameters

x	Number of collumns in the matrix
y	Number of rows in the matrix
intializator	Collection of tiles possibilities

6.18.2.2 MapMatrix() [2/2]

```
MapMatrix< T >.MapMatrix (
            int x,
            int y,
            List< Dictionary< T, float > > initialValues )
```

This constructor creates matrix with unequal primary state.

Parameters

x	Number of collumns in the matrix
y	Number of rows in the matrix
initialValues	List of tiles with its possibilities and its chances

6.18.3 Member Function Documentation

6.18.3.1 AreAllTilesSet()

```
bool MapMatrix< T >.AreAllTilesSet ( )
```

Checks if there is enthropy equal 0 or contradiction occurs on every tile of the whole matrix.

Returns

`true` if there is enthropy equal 0 or contradiction occurs on every tile of the whole matrix, `false` otherwise

6.18.3.2 GetLowestCountList()

```
List< Dictionary< T, float > > MapMatrix< T >.GetLowestCountList ( )
```

Searches for lowest, greater than 0, enthropy tiles from matrix.

Returns

List of lowest, greater than 0, enthropy tiles from matrix

6.18.3.3 GetTile()

```
T MapMatrix< T >.GetTile (
            int i,
            int j )
```

Returns particular tile from matrix.

Parameters

i	Column of the tile
j	Row of the tile

Returns

Tile T from particular cell if exists, default T value otherwise

6.18.3.4 PickRandomTile()

```
int MapMatrix< T >.PickRandomTile ( )
```

6.18.3.5 PickTileValue()

```
void MapMatrix< T >.PickTileValue (
            int i,
            int j )
```

Randomly chooses particullar tiles possibility as its only value.

Parameters

i	Collumn of tile
j	Row of tile

6.18.3.6 RemoveImposiblePairs()

```
void MapMatrix< T >.RemoveImposiblePairs ( )
```

Removes impossible candidats per tile from matrix.

6.18.3.7 ResolveMatrix()

```
void MapMatrix< T >.ResolveMatrix ( )
```

Resolves matrix with wave function collapse algorythm.

6.18.4 Member Data Documentation

6.18.4.1 int

```
MapMatrix< T >.int
```

Picks random tile from map that has lowest, greater than 0, enthropy.

Returns

Position of the chosen tile in the matrix

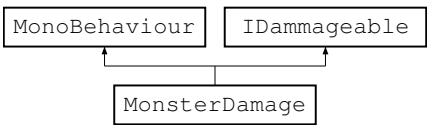
The documentation for this class was generated from the following file:

- Assets/Scripts/MapScripts/MapMatrix.cs

6.19 MonsterDamage Class Reference

Damage controller for monsters.

Inheritance diagram for MonsterDamage:



Public Member Functions

- void OnHit (float damage, Vector2 knockback)

  Deals damage to monster and if it is possible applies knockback.
- void OnHit (float damage)

  Deals damage to monster.
- bool TryMove (Vector2 direction)

  Moves player back if it's possible.

- void OnHit (float damage, Vector2 knockback)

  Deals certain amount of damage with knockback.
- void OnHit (float damage)

  Deals certain amount of damage.

Properties

- float Health `[get, set]`

  Current monster health.
- bool Targetable `[get, set]`

  Determines if monster is targetable.

Properties inherited from IDammageable

- bool Targetable `[get, set]`

6.19.1 Detailed Description

Damage controller for monsters.

6.19.2 Member Function Documentation

6.19.2.1 OnHit() `[1/2]`

```
void MonsterDamage.OnHit (
            float damage )
```

Deals damage to monster.

Parameters

damage	Damage dealt

Implements IDammageable.

6.19.2.2 OnHit() [2/2]

```
void MonsterDamage.OnHit (
            float damage,
            Vector2 knockback )
```

Deals damage to monster and if it is possible applies knockback.

Parameters

damage	Damage used to implement inteface, unused due to how player damage is constructed
knockback	Knockback to the player

Implements IDammageable.

6.19.2.3 TryMove()

```
bool MonsterDamage.TryMove (
            Vector2 direction )
```

Moves player back if it's possible.

Parameters

direction	Knockback vector

Returns

`true` if the monster can be moved back moving him, `false` if the knockback was imposible

6.19.3 Property Documentation

6.19.3.1 Health

```
float MonsterDamage.Health  [get], [set]
```

Current monster health.

6.19.3.2 Targetable

```
bool MonsterDamage.Targetable  [get], [set]
```

Determines if monster is targetable.

Implements IDammageable.

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

- Assets/Scripts/MonsterScripts/MonsterDamage.cs

6.20 MonsterDictionaryPair Class Reference

Dictionary holding monster type and its probability.

Public Attributes

- MonsterType monster
  Monster type to be restricted.
- float chance
  Chance modifier.

6.20.1 Detailed Description

Dictionary holding monster type and its probability.

6.20.2 Member Data Documentation

6.20.2.1 chance

```
float MonsterDictionaryPair.chance
```

Chance modifier.

6.20.2.2 monster

```
MonsterType MonsterDictionaryPair.monster
```

Monster type to be restricted.

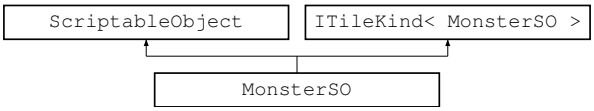
The documentation for this class was generated from the following file:

- Assets/Scripts/MonsterScripts/MonsterDictionaryPair.cs

6.21 MonsterSO Class Reference

Monster scriptable object to hadle their distributions in the room.

Inheritance diagram for MonsterSO:



Public Member Functions

- Dictionary< MonsterSO, float > FilterTiles (Dictionary< MonsterSO, float > tilesToFilter, Direction dir)
  Tile filtering method changing propability in the certain tiles.

Public Member Functions inherited from ITileKind< MonsterSO >

- Dictionary< T, float > FilterTiles (Dictionary< T, float > tileTofilter, Direction dir)
  It filters tiles in certain neighbourhood.

Public Attributes

- GameObject Monster
  Monster represented by the tile.
- MonsterType MonsterType
  Monster type of the tile.
- List< MonsterDictionaryPair > UpRestrictions
  Restrictions to the upward adjacent tile.
- List< MonsterDictionaryPair > DownRestrictions
  Restrictions to the downward adjacent tile.
- List< MonsterDictionaryPair > LeftRestrictions
  Restrictions to the left adjacent tile.
- List< MonsterDictionaryPair > RightRestrictions
  Restrictions to the left adjacent tile.

6.21.1 Detailed Description

Monster scriptable object to hadle their distributions in the room.

6.21.2 Member Function Documentation

6.21.2.1 FilterTiles()

```
Dictionary< MonsterSO, float > MonsterSO.FilterTiles (
            Dictionary< MonsterSO, float > tilesToFilter,
            Direction dir )
```

Tile filtering method changing propability in the certain tiles.

Parameters

tilesToFilter	Monsters to filter in the neighbourhood
dir	Direction of adjency

Returns

Filtered list of monsters in the adjencent tile

6.21.3 Member Data Documentation

6.21.3.1 DownRestrictions

```
List<MonsterDictionaryPair> MonsterSO.DownRestrictions
```

Restrictions to the downward adjacent tile.

6.21.3.2 LeftRestrictions

```
List<MonsterDictionaryPair> MonsterSO.LeftRestrictions
```

Restrictions to the left adjacent tile.

6.21.3.3 Monster

```
GameObject MonsterSO.Monster
```

Monster represented by the tile.

6.21.3.4 MonsterType

```
MonsterType MonsterSO.MonsterType
```

Monster type of the tile.

6.21.3.5 RightRestrictions

```
List<MonsterDictionaryPair> MonsterSO.RightRestrictions
```

Restrictions to the left adjacent tile.

6.21.3.6 UpRestrictions

```
List<MonsterDictionaryPair> MonsterSO.UpRestrictions
```

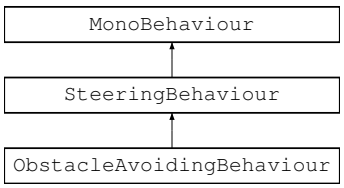
Restrictions to the upward adjacent tile.

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

- Assets/Scripts/MonsterScripts/MonsterSO.cs

6.22 ObstacleAvoidingBehaviour Class Reference

Inheritance diagram for ObstacleAvoidingBehaviour:



Public Member Functions

- override (float[ ] danger, float[ ] intrest) GetSteering(float[ ] danger

  Calculates steering for all of the obstacles.

Public Member Functions inherited from SteeringBehaviour

- abstract float[ ] float[ ] intrest GetSteering (float[ ] danger, float[ ] intrest, AIData aiData)

Additional Inherited Members

Public Attributes inherited from SteeringBehaviour

- abstract float[ ] danger

  Gets steering for monster.

6.22.1 Member Function Documentation

6.22.1.1 override()

```
ObstacleAvoidingBehaviour.override (
            float[] danger,
            float[] intrest )
```

Calculates steering for all of the obstacles.

Parameters

danger	Current danger values
intrest	Current intrest values
aiData	AI data containing all of the targets and colliders

Returns

Gets steering for monster in form of value from 0 to 1 of dangers in 8 directions and current intrest values

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

- Assets/Scripts/MonsterScripts/SlimesAI/ObstacleAvoidingBehaviour.cs

6.23 ObstacleDictionaryPair Class Reference

Obstacle type and its chance modifier.

Public Attributes

- ObstacleType obstacle

  Obstacle type.
- float chance

  Propability modifier for particular obstacle.

6.23.1 Detailed Description

Obstacle type and its chance modifier.

6.23.2 Member Data Documentation

6.23.2.1 chance

```
float ObstacleDictionaryPair.chance
```

Propability modifier for particular obstacle.

6.23.2.2 obstacle

```
ObstacleType ObstacleDictionaryPair.obstacle
```

Obstacle type.

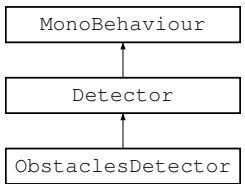
The documentation for this class was generated from the following file:

- Assets/Scripts/MapScripts/ObstacleDictionaryPair.cs

6.24 ObstaclesDetector Class Reference

Class detecting obstacles in slime vicinity.

Inheritance diagram for ObstaclesDetector:



Public Member Functions

- override void Detect (AIData aiData)
  Detects all colliders from obstacles layer mask.

- abstract void Detect (AIData ai)
  Detects targets in vasinity.

6.24.1 Detailed Description

Class detecting obstacles in slime vicinity.

6.24.2 Member Function Documentation

6.24.2.1 Detect()

```
override void ObstaclesDetector.Detect (
            AIData aiData )  [virtual]
```

Detects all colliders from obstacles layer mask.

Parameters

aiData	Current targets and obstacles

Implements Detector.

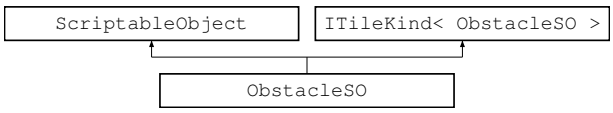
The documentation for this class was generated from the following file:

- Assets/Scripts/MonsterScripts/SlimesAI/ObstaclesDetector.cs

6.25 ObstacleSO Class Reference

Tile representation of obstacles.

Inheritance diagram for ObstacleSO:



Public Member Functions

- Dictionary< ObstacleSO, float > FilterTiles (Dictionary< ObstacleSO, float > tilesToFilter, Direction dir)

  Filters obstacle possiblities in its particullar neighbourhood.

Public Member Functions inherited from ITileKind< ObstacleSO >

- Dictionary< T, float > FilterTiles (Dictionary< T, float > tileTofilter, Direction dir)

  It filters tiles in certain neighbourhood.

Public Attributes

- UnityEngine.Object tile

  Obstacle.
- ObstacleType obstacleType

  Type of obstacle.
- List< ObstacleDictionaryPair > UpRestrictions

  Restrictions to upside adjecent tile.
- List< ObstacleDictionaryPair > DownRestrictions

  Restrictions to left side adjecent tile.
- List< ObstacleDictionaryPair > LeftRestrictions

  Restrictions to downside adjecent tile.
- List< ObstacleDictionaryPair > RightRestrictions

  Restrictions to right side adjecent tile.
- List< MonsterDictionaryPair > spawnableMonsters

  List of monsters spawnrates on the obstacle.

6.25.1 Detailed Description

Tile representation of obstacles.

6.25.2 Member Function Documentation

6.25.2.1 FilterTiles()

```
Dictionary< ObstacleSO, float > ObstacleSO.FilterTiles (
            Dictionary< ObstacleSO, float > tilesToFilter,
            Direction dir )
```

Filters obstacle possiblities in its particullar neighbourhood.

Parameters

tilesToFilter	TObstacles possibilities that needs filtering
dir	Obstacle tile adjency

Returns

Filtered obstacles possiblities

6.25.3 Member Data Documentation

6.25.3.1 DownRestrictions

```
List<ObstacleDictionaryPair> ObstacleSO.DownRestrictions
```

Restrictions to left side adjecent tile.

6.25.3.2 LeftRestrictions

```
List<ObstacleDictionaryPair> ObstacleSO.LeftRestrictions
```

Restrictions to downside adjecent tile.

6.25.3.3 obstacleType

```
ObstacleType ObstacleSO.obstacleType
```

Type of obstacle.

6.25.3.4 RightRestrictions

```
List<ObstacleDictionaryPair> ObstacleSO.RightRestrictions
```

Restrictions to right side adjecent tile.

6.25.3.5 spawnableMonsters

```
List<MonsterDictionaryPair> ObstacleSO.spawnableMonsters
```

List of monsters spawnrates on the obstacle.

6.25.3.6 tile

```
UnityEngine.Object ObstacleSO.tile
```

Obstacle.

6.25.3.7 UpRestrictions

```
List<ObstacleDictionaryPair> ObstacleSO.UpRestrictions
```

Restrictions to upside adjecent tile.

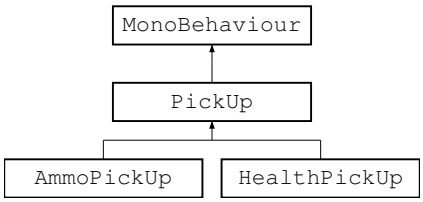
The documentation for this class was generated from the following file:

- Assets/Scripts/MapScripts/ObstacleSO.cs

6.26 PickUp Class Reference

Pick up behaviour for items.

Inheritance diagram for PickUp:



Public Member Functions

- virtual bool OnPickUp (PlayerStatsController playerStatsController)
  Virtual method describing picking up the item.

6.26.1 Detailed Description

Pick up behaviour for items.

6.26.2 Member Function Documentation

6.26.2.1 OnPickUp()

```
virtual bool PickUp.OnPickUp (
            PlayerStatsController playerStatsController )  [virtual]
```

Virtual method describing picking up the item.

Parameters

playerStatsController	Player stats controller of player, that picks up item

Returns

`true` if the item was used, `false` if the item wasn't used

Reimplemented in AmmoPickUp, and HealthPickUp.

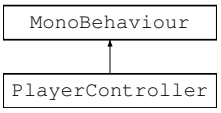
The documentation for this class was generated from the following file:

- Assets/Scripts/ItemsScripts/PickUp.cs

6.27 PlayerController Class Reference

Controls movement and on map input of the player.

Inheritance diagram for PlayerController:



6.27.1 Detailed Description

Controls movement and on map input of the player.

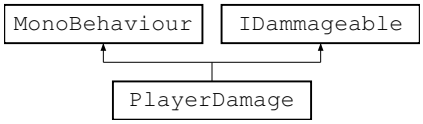
The documentation for this class was generated from the following file:

- Assets/Scripts/PlayerScripts/PlayerController.cs

6.28 PlayerDamage Class Reference

Controller for damage dealt to the player.

Inheritance diagram for PlayerDamage:



Public Member Functions

- void OnHit (float damage, Vector2 knockback)

  Deals 1 point of damage on the player and if it is possible applies knockback.
- void OnHit (float damage)

  Deals 1 point of damage on the player.
- bool TryMove (Vector2 direction)

  Moves player back if it's possible.

- void OnHit (float damage, Vector2 knockback)

  Deals certain amount of damage with knockback.
- void OnHit (float damage)

  Deals certain amount of damage.

Properties

- bool Targetable `[get, set]`

  Is player able to be targetted by the monsters.

Properties inherited from IDammageable

- bool Targetable `[get, set]`

6.28.1 Detailed Description

Controller for damage dealt to the player.

6.28.2 Member Function Documentation

6.28.2.1 OnHit() [1/2]

```
void PlayerDamage.OnHit (
            float damage )
```

Deals 1 point of damage on the player.

Parameters

damage	Damage used to implement inteface, unused due to how player damage is constructed

Implements IDammageable.

6.28.2.2 OnHit() [2/2]

```
void PlayerDamage.OnHit (
            float damage,
            Vector2 knockback )
```

Deals 1 point of damage on the player and if it is possible applies knockback.

Parameters

damage	Damage used to implement inteface, unused due to how player damage is constructed
knockback	Knockback to the player

Implements IDammageable.

6.28.2.3 TryMove()

```
bool PlayerDamage.TryMove (
            Vector2 direction )
```

Moves player back if it's possible.

Parameters

direction	Knockback vector

Returns

`true` if the player can be moved back moving him, `false` if the knockback was imposible

6.28.3 Property Documentation

6.28.3.1 Targetable

```
bool PlayerDamage.Targetable  [get], [set]
```

Is player able to be targetted by the monsters.

Implements IDammageable.

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

- Assets/Scripts/PlayerScripts/PlayerDamage.cs

6.29 PlayerDataSave Class Reference

Class responsible of containing current player data.

Properties

- int PlayerPosx `[get, set]`
  Current player pos on x axis in the room.
- int PlayerPosy `[get, set]`
  Current player pos on y axis in the room.
- float Health `[get, set]`
  Currnet player health.
- int SecondaryAmmo `[get, set]`
  Current ammo state of player.

6.29.1 Detailed Description

Class responsible of containing current player data.

6.29.2 Property Documentation

6.29.2.1 Health

```
float PlayerDataSave.Health  [get], [set]
```

Currnet player health.

6.29.2.2 PlayerPosx

```
int PlayerDataSave.PlayerPosx  [get], [set]
```

Current player pos on x axis in the room.

6.29.2.3 PlayerPosy

```
int PlayerDataSave.PlayerPosy  [get], [set]
```

Current player pos on y axis in the room.

6.29.2.4 SecondaryAmmo

```
int PlayerDataSave.SecondaryAmmo  [get], [set]
```

Current ammo state of player.

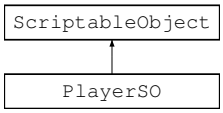
The documentation for this class was generated from the following file:

- Assets/Scripts/Saves/PlayerDataSave.cs

6.30 PlayerSO Class Reference

Players Scriptable object containing player stats.

Inheritance diagram for PlayerSO:



Public Attributes

- float speed = 1.4f

  Speed used for movement calculations.
- float attack = 1.4f

  Sword attack power.
- float attackSpeed = 1.4f

  Sword attack speed.
- float knocbackMultiplier = 1.4f

  Knockback distance modifier.
- bool isSpeedChanged = false

  Determines if the speed was changed beside the items.
- float baseSpeed = 1.4f

  Base speed of the player.
- float baseAttack = 1.4f

  Base attack power of the player.
- float baseAttackSpeed = 1.4f

  Base attack speed of the player.
- float baseKnocbackMultiplier = 1.4f

  Base knockback range of the player.
- List< ItemSO > equipment

  Unused players equipment.
- int eqSpace = 4

  Number of max items that can be equiped.
- List< ItemSO > inventory

  Unused players inverntory.
- float CurrentHealth = 5f

  Current player health.
- float MaxHealth = 5f

  Maximum number of healthpoints.
- int secondaryAmmo = 5

  Current number of bombs at disposal.
- int maxSecondaryAmmo = 5

  Maximum number of bombs player can have.

6.30.1 Detailed Description

Players Scriptable object containing player stats.

6.30.2 Member Data Documentation

6.30.2.1 attack

```
float PlayerSO.attack = 1.4f
```

Sword attack power.

6.30.2.2 attackSpeed

```
float PlayerSO.attackSpeed = 1.4f
```

Sword attack speed.

6.30.2.3 baseAttack

```
float PlayerSO.baseAttack = 1.4f
```

Base attack power of the player.

6.30.2.4 baseAttackSpeed

```
float PlayerSO.baseAttackSpeed = 1.4f
```

Base attack speed of the player.

6.30.2.5 baseKnocbackMultiplier

```
float PlayerSO.baseKnocbackMultiplier = 1.4f
```

Base knockback range of the player.

6.30.2.6 baseSpeed

```
float PlayerSO.baseSpeed = 1.4f
```

Base speed of the player.

6.30.2.7 CurrentHealth

```
float PlayerSO.CurrentHealth = 5f
```

Current player health.

6.30.2.8 eqSpace

```
int PlayerSO.eqSpace = 4
```

Number of max items that can be equiped.

6.30.2.9 equipment

```
List<ItemSO> PlayerSO.equipment
```

Unused players equipment.

6.30.2.10 inventory

```
List<ItemSO> PlayerSO.inventory
```

Unused players inverntory.

6.30.2.11 isSpeedChanged

```
bool PlayerSO.isSpeedChanged = false
```

Determines if the speed was changed beside the items.

6.30.2.12 knocbackMultiplier

```
float PlayerSO.knocbackMultiplier = 1.4f
```

Knockback distance modifier.

6.30.2.13 MaxHealth

```
float PlayerSO.MaxHealth = 5f
```

Maximum number of healthpoints.

6.30.2.14 maxSecondaryAmmo

```
int PlayerSO.maxSecondaryAmmo = 5
```

Maximum number of bombs player can have.

6.30.2.15 secondaryAmmo

```
int PlayerSO.secondaryAmmo = 5
```

Current number of bombs at disposal.

6.30.2.16 speed

```
float PlayerSO.speed = 1.4f
```

Speed used for movement calculations.

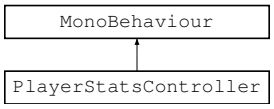
The documentation for this class was generated from the following file:

- Assets/Scripts/PlayerScripts/PlayerSO.cs

6.31 PlayerStatsController Class Reference

Controlls player stats.

Inheritance diagram for PlayerStatsController:



Public Member Functions

- float GetHealth ()

  Returns current player health.
- int GetAmmo ()

  Returns number of bombs available.
- bool Heal ()

  Heals player for one healthpoint.
- bool AmmoPickUp ()

  Adds one bomb to the amount.
- bool UseSecondary ()

  Uses one of the bombs.
- void SetPlayerCords (int x, int y)

  Teleports player to desired position according to grid.
- void SetPlayerBegginningStats ()

  Sets player statistics on the begining of the game with the default values.
- void SetPlayerLoadedStats (float health, int ammo)

  Sets up player statistics to default beside health and bombs which comes from save file.

Static Public Attributes

- static PlayerStatsController Instance

  Instance of player stats controller.

6.31.1 Detailed Description

Controlls player stats.

6.31.2 Member Function Documentation

6.31.2.1 AmmoPickUp()

```
bool PlayerStatsController.AmmoPickUp ( )
```

Adds one bomb to the amount.

Returns

`true` if piucking up the bomb was succesful, `false` if player has maximum number of bombs

6.31.2.2 GetAmmo()

```
int PlayerStatsController.GetAmmo ( )
```

Returns number of bombs available.

Returns

Current ammo level

6.31.2.3 GetHealth()

```
float PlayerStatsController.GetHealth ( )
```

Returns current player health.

Returns

Current player health

6.31.2.4 Heal()

```
bool PlayerStatsController.Heal ( )
```

Heals player for one healthpoint.

Returns

`true` if the healing try was successful, `false` if player already has full health

6.31.2.5 SetPlayerBegginningStats()

```
void PlayerStatsController.SetPlayerBegginningStats ( )
```

Sets player statistics on the begining of the game with the default values.

6.31.2.6 SetPlayerCords()

```
void PlayerStatsController.SetPlayerCords (
            int x,
            int y )
```

Teleports player to desired position according to grid.

Parameters

x	X coordinate on the grid
y	Y coordinate on the grid

6.31.2.7 SetPlayerLoadedStats()

```
void PlayerStatsController.SetPlayerLoadedStats (
            float health,
            int ammo )
```

Sets up player statistics to default beside health and bombs which comes from save file.

Parameters

health	Number of hearths to set up
ammo	Number of bombs available to player

6.31.2.8 UseSecondary()

```
bool PlayerStatsController.UseSecondary ( )
```

Uses one of the bombs.

Returns

`true` if player can use bomb, `false` if player has no bombs available

6.31.3 Member Data Documentation

6.31.3.1 Instance

```
PlayerStatsController PlayerStatsController.Instance  [static]
```

Instance of player stats controller.

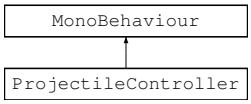
The documentation for this class was generated from the following file:

- Assets/Scripts/PlayerScripts/PlayerStatsController.cs

6.32 ProjectileController Class Reference

Projectile creation, movement and destruction controller.

Inheritance diagram for ProjectileController:



Properties

- Vector3 Target `[get, set]`
  Target position.

6.32.1 Detailed Description

Projectile creation, movement and destruction controller.

6.32.2 Property Documentation

6.32.2.1 Target

```
Vector3 ProjectileController.Target  [get], [set]
```

Target position.

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

- Assets/Scripts/MonsterScripts/ProjectileController.cs

6.33 Room Class Reference

Class containing all rooms data from particular room in the dungeon.

Public Member Functions

- Room (int i, int j, RoomSO roomKind, int seed)
  This constructor creates new unconquered room.
- bool[ ] GetRoomsDoorsAsArray ()
  Translates invidual determinates of the doors into array.

Properties

- int x [get, set]

  Room collumn in the matrix.
- int y [get, set]

  Room row in the matrix.
- int Seed [get, set]

  Random generation seed of the room.
- List<(int x, int y)> ChestOpened [get, set]

  List of the opened chests positions.
- RoomSO RoomKind [get, set]

  Room kind description containing rooms.
- bool IsConquered [get, set]

  Determines if room is conquered.

6.33.1 Detailed Description

Class containing all rooms data from particular room in the dungeon.

6.33.2 Constructor & Destructor Documentation

6.33.2.1 Room()

```
Room.Room (
            int i,
            int j,
            RoomSO roomKind,
            int seed )
```

This constructor creates new unconquered room.

Parameters

i	Room collumn in the matrix
j	Room row in the matrix
roomKind	Initilize roomKind for doors
seed	Room seed for map generation

6.33.3 Member Function Documentation

6.33.3.1 GetRoomsDoorsAsArray()

```
bool[] Room.GetRoomsDoorsAsArray ( )
```

Translates invidual determinates of the doors into array.

Returns

Array with the doors determinates in order: right, up, left, down

6.33.4 Property Documentation

6.33.4.1 ChestOpened

```
List<(int x, int y)> Room.ChestOpened  [get], [set]
```

List of the opened chests positions.

6.33.4.2 IsConquered

```
bool Room.IsConquered  [get], [set]
```

Determines if room is conquered.

6.33.4.3 RoomKind

```
RoomSO Room.RoomKind  [get], [set]
```

Room kind description containing rooms.

6.33.4.4 Seed

```
int Room.Seed  [get], [set]
```

Random generation seed of the room.

6.33.4.5 x

```
int Room.x [get], [set]
```

Room collumn in the matrix.

6.33.4.6 y

```
int Room.y [get], [set]
```

Room row in the matrix.

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

- Assets/Scripts/RoomsGenerator/Room.cs

6.34 RoomsDataSave Class Reference

Class responsible of containing current dungeon data.

Properties

- int RoomPosx `[get, set]`
  Room collumn in the matrix.
- int RoomPosy `[get, set]`
  Room row in the matrix.
- List<(int x, int y)> ChestOpened `[get, set]`
  List of the opened chests in the room.
- bool IsConquered `[get, set]`
  Determines if room was conquered.

6.34.1 Detailed Description

Class responsible of containing current dungeon data.

6.34.2 Property Documentation

6.34.2.1 ChestOpened

```
List<(int x, int y)> RoomsDataSave.ChestOpened  [get], [set]
```

List of the opened chests in the room.

6.34.2.2 IsConquered

```
bool RoomsDataSave.IsConquered  [get], [set]
```

Determines if room was conquered.

6.34.2.3 RoomPosx

```
int RoomsDataSave.RoomPosx  [get], [set]
```

Room collumn in the matrix.

6.34.2.4 RoomPosy

```
int RoomsDataSave.RoomPosy  [get], [set]
```

Room row in the matrix.

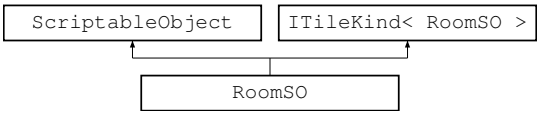
The documentation for this class was generated from the following file:

- Assets/Scripts/Saves/RoomsDataSave.cs

6.35 RoomSO Class Reference

Scriptable object representing kind of room by describing its sides.

Inheritance diagram for RoomSO:



Public Member Functions

- Dictionary< RoomSO, float > FilterTiles (Dictionary< RoomSO, float > tilesToFilter, Direction dir)

  Filteres tiles from certain neighbouring field.

Public Member Functions inherited from ITileKind< RoomSO >

- Dictionary< T, float > FilterTiles (Dictionary< T, float > tileTofilter, Direction dir)

  It filters tiles in certain neighbourhood.

Public Attributes

- bool IsDoorUp

  Determines if there are doors up.
- bool IsDoorDown

  Determines if there are doors down.
- bool IsDoorLeft

  Determines if there are doors left.
- bool IsDoorRight

  Determines if there are doors right.

6.35.1 Detailed Description

Scriptable object representing kind of room by describing its sides.

6.35.2 Member Function Documentation

6.35.2.1 FilterTiles()

```
Dictionary< RoomSO, float > RoomSO.FilterTiles (
            Dictionary< RoomSO, float > tilesToFilter,
            Direction dir )
```

Filteres tiles from certain neighbouring field.

Parameters

tilesToFilter	Tiles posiblities aligned in neighborhood
dir	Determines which side is considered in comparations

Returns

Filtered rooms list

6.35.3 Member Data Documentation

6.35.3.1 IsDoorDown

```
bool RoomSO.IsDoorDown
```

Determines if there are doors down.

6.35.3.2 IsDoorLeft

```
bool RoomSO.IsDoorLeft
```

Determines if there are doors left.

6.35.3.3 IsDoorRight

```
bool RoomSO.IsDoorRight
```

Determines if there are doors right.

6.35.3.4 IsDoorUp

```
bool RoomSO.IsDoorUp
```

Determines if there are doors up.

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

- Assets/Scripts/RoomsGenerator/RoomSO.cs

6.36 SaveData Class Reference

Class holding save data of the game.

Public Attributes

- int x

  Current room position in matrix.

Properties

- static SaveData Current [get, set]

  Public getter and setter for singleton instance of save file.
- PlayerDataSave Player [get, set]

  PlayerData to save.
- int Seed [get, set]

  Seed of the world.
- List< RoomsDataSave > Rooms [get, set]

  Rooms with their seeds, chests and conquers.
- int int y CurrentRoom [get, set]
- float Timer [get, set]

  Current timer state.

6.36.1 Detailed Description

Class holding save data of the game.

6.36.2 Member Data Documentation

6.36.2.1 x

```
int SaveData.x
```

Current room position in matrix.

6.36.3 Property Documentation

6.36.3.1 Current

```
SaveData SaveData.Current [static], [get], [set]
```

Public getter and setter for singleton instance of save file.

6.36.3.2 CurrentRoom

```
int int y SaveData.CurrentRoom [get], [set]
```

6.36.3.3 Player

```
PlayerDataSave SaveData.Player  [get], [set]
```

PlayerData to save.

6.36.3.4 Rooms

```
List<RoomsDataSave> SaveData.Rooms  [get], [set]
```

Rooms with their seeds, chests and conquers.

6.36.3.5 Seed

```
int SaveData.Seed  [get], [set]
```

Seed of the world.

6.36.3.6 Timer

```
float SaveData.Timer  [get], [set]
```

Current timer state.

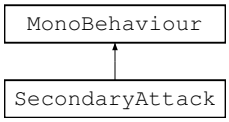
The documentation for this class was generated from the following file:

- Assets/Scripts/Saves/SaveData.cs

6.37 SecondaryAttack Class Reference

Evaporation bombs controller.

Inheritance diagram for SecondaryAttack:



6.37.1 Detailed Description

Evaporation bombs controller.

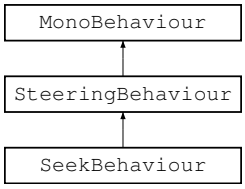
The documentation for this class was generated from the following file:

- Assets/Scripts/PlayerScripts/SecondaryAttack.cs

6.38 SeekBehaviour Class Reference

Behaviour for seeking the players.

Inheritance diagram for SeekBehaviour:



Public Member Functions

- override (float[ ] danger, float[ ] intrest) GetSteering(float[ ] danger

  Applies intrest depending on current target position.

Public Member Functions inherited from SteeringBehaviour

- abstract float[ ] float[ ] intrest GetSteering (float[ ] danger, float[ ] intrest, AIData aiData)

Additional Inherited Members

Public Attributes inherited from SteeringBehaviour

- abstract float[ ] danger

  Gets steering for monster.

6.38.1 Detailed Description

Behaviour for seeking the players.

6.38.2 Member Function Documentation

6.38.2.1 override()

```
SeekBehaviour.override (
            float[] danger,
            float[] intrest )
```

Applies intrest depending on current target position.

Parameters

danger	Current dangers values
intrest	Previous intrest values
aiData	Current data for steering with positions of obstacles and targets

Returns

Gets steering for monster in form of value from 0 to 1 of intrest in 8 directions and current dangers values

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

- Assets/Scripts/MonsterScripts/SlimesAI/SeekBehaviour.cs

6.39 SerializationManager Class Reference

Class responsible for serializing save game obbject into binary.

Static Public Member Functions

- static bool Save (string saveName, object saveData)
  Saves object into save file by overriding it.
- static object Load (string path)
  Load game file from file.
- static BinaryFormatter GetBinaryFromatter ()
  Provides binary formatters with all surogates.

6.39.1 Detailed Description

Class responsible for serializing save game obbject into binary.

6.39.2 Member Function Documentation

6.39.2.1 GetBinaryFromatter()

```
static BinaryFormatter SerializationManager.GetBinaryFromatter ( )  [static]
```

Provides binary formatters with all surogates.

Returns

Sutable to creating save file binary formatter

6.39.2.2 Load()

```
static object SerializationManager.Load (
            string path )  [static]
```

Load game file from file.

Parameters

path	Path to save file

Returns

Game state object

6.39.2.3 Save()

```
static bool SerializationManager.Save (
            string saveName,
            object saveData )  [static]
```

Saves object into save file by overriding it.

Parameters

saveName	Save file name
saveData	Object with save data

Returns

True if succided, false if savve havent succeded

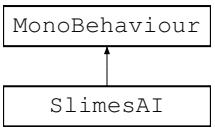
The documentation for this class was generated from the following file:

- Assets/Scripts/Saves/SerializationManager.cs

6.40 SlimesAI Class Reference

Class controling all behaviours of slimes.

Inheritance diagram for SlimesAI:



6.40.1 Detailed Description

Class controling all behaviours of slimes.

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

- Assets/Scripts/MonsterScripts/SlimesAI/SlimesAI.cs

6.41 SpawnRate Class Reference

Monster spawn chance modifier with its monster type.

Public Attributes

- MonsterType spawn

  Monster type to spawn.
- float chance

  Propability modifier.

6.41.1 Detailed Description

Monster spawn chance modifier with its monster type.

6.41.2 Member Data Documentation

6.41.2.1 chance

```
float SpawnRate.chance
```

Propability modifier.

6.41.2.2 spawn

```
MonsterType SpawnRate.spawn
```

Monster type to spawn.

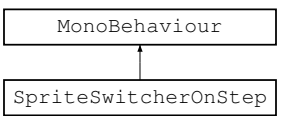
The documentation for this class was generated from the following file:

- Assets/Scripts/MapScripts/SpawnRate.cs

6.42 SpriteSwitcherOnStep Class Reference

Switches sprites when stepped on by player.

Inheritance diagram for SpriteSwitcherOnStep:



6.42.1 Detailed Description

Switches sprites when stepped on by player.

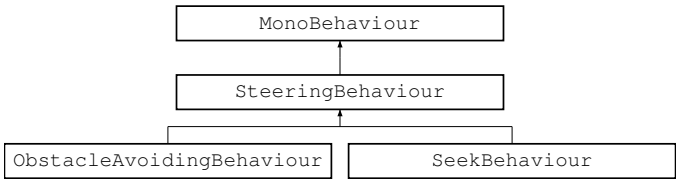
The documentation for this class was generated from the following file:

- Assets/Scripts/MapScripts/Obstacles/SpriteSwitcherOnStep.cs

6.43 SteeringBehaviour Class Reference

Abstraction for monster steering.

Inheritance diagram for SteeringBehaviour:



Public Member Functions

- abstract float[ ] float[ ] intrest GetSteering (float[ ] danger, float[ ] intrest, AIData aiData)

Public Attributes

- abstract float[ ] danger
  Gets steering for monster.

6.43.1 Detailed Description

Abstraction for monster steering.

6.43.2 Member Function Documentation

6.43.2.1 GetSteering()

```
abstract float[] float[] intrest SteeringBehaviour.GetSteering (
            float[] danger,
            float[] intrest,
            AIData aiData )
```

6.43.3 Member Data Documentation

6.43.3.1 danger

```
abstract float [] SteeringBehaviour.danger
```

Gets steering for monster.

Parameters

danger	Current dangers values
intrest	Current intrest values
aiData	Current data for steering with positions of obstacles and targets

Returns

Gets steering for monster in form of value from 0 to 1 of intrest in 8 directions and same number of values of danger in tuple

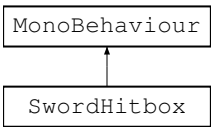
The documentation for this class was generated from the following file:

- Assets/Scripts/MonsterScripts/SlimesAI/SteeringBehaviour.cs

6.44 SwordHitbox Class Reference

Sword attacking hitbox controller.

Inheritance diagram for SwordHitbox:



Public Member Functions

- void OnTriggerEnter2D (Collider2D col)
- void Attack (Direction dir)

  Control attack i regards to direction.
- void AttackFinish ()

  Finishes attacking state.

6.44.1 Detailed Description

Sword attacking hitbox controller.

6.44.2 Member Function Documentation

6.44.2.1 Attack()

```
void SwordHitbox.Attack (
            Direction dir )
```

Control attack i regards to direction.

Parameters

dir	Direction of atacking

6.44.2.2 AttackFinish()

```
void SwordHitbox.AttackFinish ( )
```

Finishes attacking state.

6.44.2.3 OnTriggerEnter2D()

```
void SwordHitbox.OnTriggerEnter2D (
            Collider2D col )
```

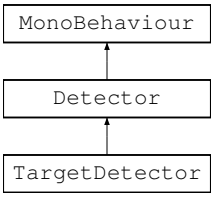
The documentation for this class was generated from the following file:

- Assets/Scripts/PlayerScripts/SwordHitbox.cs

6.45 TargetDetector Class Reference

Detector implementation for targets.

Inheritance diagram for TargetDetector:



Public Member Functions

- override void Detect (AIData aiData)

  Detects player in monster vicinity.

- abstract void Detect (AIData ai)

  Detects targets in vasinity.

6.45.1 Detailed Description

Detector implementation for targets.

6.45.2 Member Function Documentation

6.45.2.1 Detect()

```
override void TargetDetector.Detect (
            AIData aiData )  [virtual]
```

Detects player in monster vicinity.

Parameters

aiData	Data stored for the ai

Implements Detector.

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

- Assets/Scripts/MonsterScripts/SlimesAI/TargetDetector.cs

6.46 TileDictionaryPair Class Reference

Tiles side propability modifier.

Public Attributes

- SideDescription kindOfSide
  Name of the side of the tile.
- float chance
  Propability modifier.

6.46.1 Detailed Description

Tiles side propability modifier.

6.46.2 Member Data Documentation

6.46.2.1 chance

```
float TileDictionaryPair.chance
```

Propability modifier.

6.46.2.2 kindOfSide

```
SideDescription TileDictionaryPair.kindOfSide
```

Name of the side of the tile.

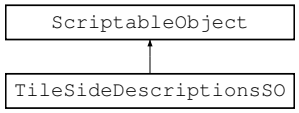
The documentation for this class was generated from the following file:

- Assets/Scripts/MapScripts/TileDictionaryPair.cs

6.47 TileSideDescriptionsSO Class Reference

Tile side descriptor, it is used to prepare sides restriction per side kind.

Inheritance diagram for TileSideDescriptionsSO:



Public Attributes

- SideDescription SideName
  Name of the tile side kind.
- List< TileDictionaryPair > sideRestrictions
  Tile side Restrictions.

6.47.1 Detailed Description

Tile side descriptor, it is used to prepare sides restriction per side kind.

6.47.2 Member Data Documentation

6.47.2.1 SideName

```
SideDescription TileSideDescriptionsSO.SideName
```

Name of the tile side kind.

6.47.2.2 sideRestrictions

```
List<TileDictionaryPair> TileSideDescriptionsSO.sideRestrictions
```

Tile side Restrictions.

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

- Assets/Scripts/MapScripts/TileSideDescriptionsSO.cs

6.48 TileSO Class Reference

Representation of singular floor tile, descripting its sides.

Inheritance diagram for TileSO:

Public Member Functions

- Dictionary< TileSO, float > FilterTiles (Dictionary< TileSO, float > tileToFilter, Direction dir)
  Filters floor tiles in its particullar neighbourhood.

Public Member Functions inherited from ITileKind< TileSO >

- Dictionary< T, float > FilterTiles (Dictionary< T, float > tileTofilter, Direction dir)
  It filters tiles in certain neighbourhood.

Public Attributes

- Tile tile

  Tile it is representing.
- TileSideDescriptionsSO upSide

  Upside descriptor.
- TileSideDescriptionsSO leftSide

  Left descriptor.
- TileSideDescriptionsSO rightSide

  Right descriptor.
- TileSideDescriptionsSO downSide

  Down descriptor.
- TileSideDescriptionsSO upLeftCorner

  Upper left corner descriptor.
- TileSideDescriptionsSO upRightCorner

  Upper right corner descriptor.
- TileSideDescriptionsSO downLeftCorner

  Down left corner descriptor.
- TileSideDescriptionsSO downRightCorner

  Down right corner descriptor.
- List< ObstacleDictionaryPair > secondLayerRestrictions

  Layer above restrictions.
- List< MonsterDictionaryPair > spawnableMonsters

  Monsters spawn restrictions.

6.48.1 Detailed Description

Representation of singular floor tile, descripting its sides.

6.48.2 Member Function Documentation

6.48.2.1 FilterTiles()

```
Dictionary< TileSO, float > TileSO.FilterTiles (
            Dictionary< TileSO, float > tileToFilter,
            Direction dir )
```

Filters floor tiles in its particullar neighbourhood.

Parameters

tileToFilter	Tiles possibilities that needs filtering
dir	Floor tile adjency

Returns

Filtered tiles possiblities

6.48.3 Member Data Documentation

6.48.3.1 downLeftCorner

`TileSideDescriptionsSO TileSO.downLeftCorner`

Down left corner descriptor.

6.48.3.2 downRightCorner

`TileSideDescriptionsSO TileSO.downRightCorner`

Down right corner descriptor.

6.48.3.3 downSide

`TileSideDescriptionsSO TileSO.downSide`

Down descriptor.

6.48.3.4 leftSide

`TileSideDescriptionsSO TileSO.leftSide`

Left descriptor.

6.48.3.5 rightSide

`TileSideDescriptionsSO TileSO.rightSide`

Right descriptor.

6.48.3.6 secondLayerRestrictions

```
List<ObstacleDictionaryPair> TileSO.secondLayerRestrictions
```

Layer above restrictions.

6.48.3.7 spawnableMonsters

```
List<MonsterDictionaryPair> TileSO.spawnableMonsters
```

Monsters spawn restrictions.

6.48.3.8 tile

```
Tile TileSO.tile
```

Tile it is representing.

6.48.3.9 upLeftCorner

```
TileSideDescriptionsSO TileSO.upLeftCorner
```

Upper left corner descriptor.

6.48.3.10 upRightCorner

```
TileSideDescriptionsSO TileSO.upRightCorner
```

Upper right corner descriptor.

6.48.3.11 upSide

```
TileSideDescriptionsSO TileSO.upSide
```

Upside descriptor.

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

- Assets/Scripts/MapScripts/TileSO.cs

6.49 BarsElements.Bar.UxmlFactory Class Reference

Element as Uxml factory.

6.49.1 Detailed Description

Element as Uxml factory.

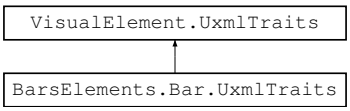
The documentation for this class was generated from the following file:

- Assets/Scripts/CustomUIElelements/Bar.cs

6.50 BarsElements.Bar.UxmlTraits Class Reference

Exclusive element traits factory for uxml editor and events.

Inheritance diagram for BarsElements.Bar.UxmlTraits:



Public Member Functions

- override void Init (VisualElement ve, IUxmlAttributes bag, CreationContext cc)

Properties

- override IEnumerable< UxmlChildElementDescription > uxmlChildElementsDescription [get]

6.50.1 Detailed Description

Exclusive element traits factory for uxml editor and events.

6.50.2 Member Function Documentation

6.50.2.1 Init()

```
override void BarsElements.Bar.UxmlTraits.Init (
            VisualElement ve,
            IUxmlAttributes bag,
            CreationContext cc )
```

6.50.3 Property Documentation

6.50.3.1 uxmlChildElementsDescription

```
override IEnumerable<UxmlChildElementDescription> BarsElements.Bar.UxmlTraits.uxmlChild↩
ElementsDescription [get]
```

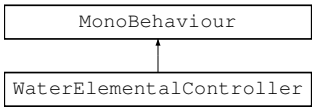
The documentation for this class was generated from the following file:

- Assets/Scripts/CustomUIElelements/Bar.cs

6.51 WaterElementalController Class Reference

Controls water elemental behaviours.

Inheritance diagram for WaterElementalController:



Public Member Functions

- void Spawn ()

  Plays spawn animation.
- void Despawn ()

  Plays despawn animation.

6.51.1 Detailed Description

Controls water elemental behaviours.

6.51.2 Member Function Documentation

6.51.2.1 Despawn()

```
void WaterElementalController.Despawn ( )
```

Plays despawn animation.

6.51.2.2 Spawn()

```
void WaterElementalController.Spawn ( )
```

Plays spawn animation.

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

- Assets/Scripts/MonsterScripts/WaterElementalController.cs

Chapter 7

File Documentation

7.1 Assets/Resources/UI/InventorySlot.cs File Reference

Classes

- class InventorySlot

7.2 Assets/Scripts/CommonIntefaces/IDammageable.cs File Reference

Classes

- interface IDammageable

  Damage handling interface.

7.3 Assets/Scripts/CustomUIElelements/Bar.cs File Reference

Classes

- class BarsElements.Bar

  Status bar ui element.
- class BarsElements.Bar.UxmlFactory

  Element as Uxml factory.
- class BarsElements.Bar.UxmlTraits

  Exclusive element traits factory for uxml editor and events.

Namespaces

- namespace BarsElements

7.4 Assets/Scripts/Direction.cs File Reference

Enumerations

- enum Direction {
  up , left , down , right ,
  upRight , upLeft , downLeft , downRight }
  Directions descriptions.

7.4.1 Enumeration Type Documentation

7.4.1.1 Direction

```
enum Direction
```

Directions descriptions.

Enumerator

up	
left	
down	
right	
upRight	
upLeft	
downLeft	
downRight	

7.5 Assets/Scripts/GameStateController.cs File Reference

Classes

- class GameStateController
  Class managing state of the game, responsible for pausing and etc.

7.6 Assets/Scripts/HelperFunctions.cs File Reference

Classes

- class **HelperFunctions**
  Static class for common functions.

7.7 Assets/Scripts/ItemsScripts/AmmoPickUp.cs File Reference

Classes

- class AmmoPickUp

7.8 Assets/Scripts/ItemsScripts/HealthPickUp.cs File Reference

Classes

- class HealthPickUp

  Pick up healing item behaviour.

7.9 Assets/Scripts/ItemsScripts/PickUp.cs File Reference

Classes

- class PickUp

  Pick up behaviour for items.

7.10 Assets/Scripts/MapScripts/DoorsController.cs File Reference

Classes

- class DoorsController

  Doors behaviour controller.

7.11 Assets/Scripts/MapScripts/ITileKind.cs File Reference

Classes

- interface ITileKind< T >

  Interface allowing the class to be used in map matrix class.

7.12 Assets/Scripts/MapScripts/MapGeneration.cs File Reference

Classes

- class MapGeneration

  Controlls room generation.

7.13 Assets/Scripts/MapScripts/MapMatrix.cs File Reference

Classes

- class MapMatrix< T >

  Matrix using wave function collapse to be resolved.

7.14 Assets/Scripts/MapScripts/ObstacleDictionaryPair.cs File Reference

Classes

- class ObstacleDictionaryPair

  Obstacle type and its chance modifier.

7.15 Assets/Scripts/MapScripts/Obstacles/ChestController.cs File Reference

Classes

- class ChestController

  Controlles chest behaviours.

7.16 Assets/Scripts/MapScripts/Obstacles/IceRuneControler.cs File Reference

Classes

- class IceRuneControler

  Ice rune behaviour controller.

7.17 Assets/Scripts/MapScripts/Obstacles/ItemSpawnRate.cs File Reference

Classes

- class ItemSpawnRate

  Spawn possibility container.

7.18 Assets/Scripts/MapScripts/Obstacles/SpriteSwitcherOnStep.cs File Reference

Classes

- class SpriteSwitcherOnStep

  Switches sprites when stepped on by player.

7.19 Assets/Scripts/MapScripts/ObstacleSO.cs File Reference

Classes

- class ObstacleSO

  Tile representation of obstacles.

7.20 Assets/Scripts/MapScripts/ObstacleType.cs File Reference

Enumerations

- enum ObstacleType {
  Empty , FireRune , IceRune , MossyBoulder ,
  Boudler , WoodenChest , MetalChest , GrassyMetalChest ,
  GrassyWoodenChest }

  Contains all of obstacles types.

7.20.1 Enumeration Type Documentation

7.20.1.1 ObstacleType

```
enum ObstacleType
```

Contains all of obstacles types.

Enumerator

Empty	
FireRune	
IceRune	
MossyBoulder	
Boudler	
WoodenChest	
MetalChest	
GrassyMetalChest	
GrassyWoodenChest	

7.21 Assets/Scripts/MapScripts/SideDescription.cs File Reference

Enumerations

- enum SideDescription {
  Nothing , UpWaterAndGroundDown , DownWaterAndGroundUp , Water ,
  Ground , Other , UpWaterAndGroundDownCorner , DownWaterAndGroundUpCorner ,
  WaterGround , GroundWater , CrackedGournd , PartGrass ,
  Grass , GrassWater , WaterGrass , CrackedWater ,
  WaterCracked }

  Contains all floor tiles names available.

7.21.1 Enumeration Type Documentation

7.21.1.1 SideDescription

`enum SideDescription`

Contains all floor tiles names available.

Enumerator

Nothing	
UpWaterAndGroundDown	
DownWaterAndGroundUp	
Water	
Ground	
Other	
UpWaterAndGroundDownCorner	
DownWaterAndGroundUpCorner	
WaterGround	
GroundWater	
CrackedGournd	
PartGrass	
Grass	
GrassWater	
WaterGrass	
CrackedWater	
WaterCracked	

7.22 Assets/Scripts/MapScripts/SpawnRate.cs File Reference

Classes

- class SpawnRate

  Monster spawn chance modifier with its monster type.

7.23 Assets/Scripts/MapScripts/TileDictionaryPair.cs File Reference

Classes

- class TileDictionaryPair

  Tiles side propability modifier.

7.24 Assets/Scripts/MapScripts/TileSideDescriptionsSO.cs File Reference

Classes

- class TileSideDescriptionsSO

  Tile side descriptor, it is used to prepare sides restriction per side kind.

7.25 Assets/Scripts/MapScripts/TileSO.cs File Reference

Classes

- class TileSO

  Representation of singular floor tile, descripting its sides.

7.26 Assets/Scripts/MonsterScripts/MonsterDamage.cs File Reference

Classes

- class MonsterDamage

  Damage controller for monsters.

7.27 Assets/Scripts/MonsterScripts/MonsterDictionaryPair.cs File Reference

Classes

- class MonsterDictionaryPair

  Dictionary holding monster type and its probability.

7.28 Assets/Scripts/MonsterScripts/MonsterSO.cs File Reference

Classes

- class MonsterSO

  Monster scriptable object to hadle their distributions in the room.

7.29 Assets/Scripts/MonsterScripts/MonsterType.cs File Reference

Enumerations

- enum MonsterType {
  WaterElemental , GrassSlime , IceSlime , FireSlime ,
  Empty }

  Enum for counting monster types.

7.29.1 Enumeration Type Documentation

7.29.1.1 MonsterType

```
enum MonsterType
```

Enum for counting monster types.

Enumerator

WaterElemental	
GrassSlime	
IceSlime	
FireSlime	
Empty	

7.30 Assets/Scripts/MonsterScripts/ProjectileController.cs File Reference

Classes

- class ProjectileController

  Projectile creation, movement and destruction controller.

7.31 Assets/Scripts/MonsterScripts/SlimesAI/AIData.cs File Reference

Classes

- class AIData

  Targets and obstacles positions container.

7.32 Assets/Scripts/MonsterScripts/SlimesAI/ContextSolver.cs File Reference

Classes

- class ContextSolver

  Calculates movement using steering directions.

7.33 Assets/Scripts/MonsterScripts/SlimesAI/Detector.cs File Reference

Classes

- class Detector

  Detectors abstraction layer.

7.34 Assets/Scripts/MonsterScripts/SlimesAI/ObstacleAvoidingBehaviour.cs File Reference

Classes

- class ObstacleAvoidingBehaviour

7.35 Assets/Scripts/MonsterScripts/SlimesAI/ObstaclesDetector.cs File Reference

Classes

- class ObstaclesDetector

  Class detecting obstacles in slime vicinity.

7.36 Assets/Scripts/MonsterScripts/SlimesAI/SeekBehaviour.cs File Reference

Classes

- class SeekBehaviour

  Behaviour for seeking the players.

7.37 Assets/Scripts/MonsterScripts/SlimesAI/SlimesAI.cs File Reference

Classes

- class SlimesAI

  Class controling all behaviours of slimes.

7.38 Assets/Scripts/MonsterScripts/SlimesAI/SteeringBehaviour.cs File Reference

Classes

- class SteeringBehaviour

  Abstraction for monster steering.

7.39 Assets/Scripts/MonsterScripts/SlimesAI/TargetDetector.cs File Reference

Classes

- class TargetDetector

  Detector implementation for targets.

7.40 Assets/Scripts/MonsterScripts/WaterElementalController.cs File Reference

Classes

- class WaterElementalController

  Controls water elemental behaviours.

7.41 Assets/Scripts/PlayerScripts/EnemyHitChecker.cs File Reference

Classes

- class EnemyHitChecker

  Checks if player was hit.

7.42 Assets/Scripts/PlayerScripts/ItemSO.cs File Reference

Classes

- class ItemSO

  Unused items Scriptable object, left for future.

7.43 Assets/Scripts/PlayerScripts/PlayerController.cs File Reference

Classes

- class PlayerController

  Controls movement and on map input of the player.

7.44 Assets/Scripts/PlayerScripts/PlayerDamage.cs File Reference

Classes

- class PlayerDamage

  Controller for damage dealt to the player.

7.45 Assets/Scripts/PlayerScripts/PlayerSO.cs File Reference

Classes

- class PlayerSO

  Players Scriptable object containing player stats.

7.46 Assets/Scripts/PlayerScripts/PlayerStatsController.cs File Reference

Classes

- class PlayerStatsController

  Controlls player stats.

7.47 Assets/Scripts/PlayerScripts/SecondaryAttack.cs File Reference

Classes

- class SecondaryAttack

  Evaporation bombs controller.

7.48 Assets/Scripts/PlayerScripts/SwordHitbox.cs File Reference

Classes

- class SwordHitbox

  Sword attacking hitbox controller.

7.49 Assets/Scripts/RoomsGenerator/Room.cs File Reference

Classes

- class Room

  Class containing all rooms data from particular room in the dungeon.

7.50 Assets/Scripts/RoomsGenerator/RoomSO.cs File Reference

Classes

- class RoomSO

  Scriptable object representing kind of room by describing its sides.

7.51 Assets/Scripts/Saves/PlayerDataSave.cs File Reference

Classes

- class PlayerDataSave

  Class responsible of containing current player data.

7.52 Assets/Scripts/Saves/RoomsDataSave.cs File Reference

Classes

- class RoomsDataSave

  Class responsible of containing current dungeon data.

7.53 Assets/Scripts/Saves/SaveData.cs File Reference

Classes

- class SaveData

  Class holding save data of the game.

7.54 Assets/Scripts/Saves/SerializationManager.cs File Reference

Classes

- class SerializationManager

  Class responsible for serializing save game obbject into binary.

Index