

Mixchemist

Generated by Doxygen 1.11.0

1 Namespace Index	1
1.1 Package List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 Namespace Documentation	7
4.1 Dungeon Namespace Reference	7
4.2 Dungeon.Generator Namespace Reference	7
4.3 Dungeon.Walker Namespace Reference	7
4.4 mixchemist2 Namespace Reference	7
4.5 mixchemist2.spell Namespace Reference	7
5 Class Documentation	9
5.1 AbstractEnemy Class Reference	9
5.1.1 Member Function Documentation	9
5.1.1.1 _OnBulletHit()	9
5.1.1.2 SetElement()	10
5.1.1.3 TakeDamage()	10
5.2 ClassesAndEnums Class Reference	10
5.3 mixchemist2.spell.ConcreteSpell Class Reference	11
5.3.1 Member Function Documentation	11
5.3.1.1 GetDamage()	11
5.3.1.2 GetElement()	12
5.3.1.3 GetPosition()	12
5.3.1.4 SetElement()	12
5.4 DeathMenu Class Reference	12
5.5 DevScene Class Reference	13
5.6 ElementGauge Class Reference	13
5.6.1 Member Function Documentation	13
5.6.1.1 UpdateGauge()	13
5.7 ElementStorage Class Reference	14
5.7.1 Member Function Documentation	14
5.7.1.1 _Input()	14
5.7.1.2 CastFirstElementInStorage()	15
5.7.1.3 GetFirstRealmElement()	15
5.7.1.4 StoreSpellColor()	15
5.7.1.5 ToggleElementPanelColor()	15
5.8 EnemyTexture Class Reference	16
5.8.1 Member Function Documentation	16
5.8.1.1 SetEnemyTexture()	16

5.9 GameManager Class Reference	16
5.9.1 Member Function Documentation	17
5.9.1.1 GetScore()	17
5.9.1.2 SetScore()	17
5.10 HealthBar Class Reference	17
5.10.1 Member Function Documentation	18
5.10.1.1 UpdateHealthBar()	18
5.11 HealthBarControl Class Reference	18
5.12 Highscore Class Reference	18
5.13 LevelConfiguration Class Reference	19
5.14 MainMenu Class Reference	19
5.15 MusicManager Class Reference	20
5.15.1 Member Function Documentation	20
5.15.1.1 _Ready()	20
5.15.1.2 ChangeStream()	20
5.16 PauseMenu Class Reference	21
5.17 Player Class Reference	21
5.17.1 Member Function Documentation	21
5.17.1.1 TakeDamage()	21
5.18 PlayerInputField Class Reference	22
5.18.1 Member Function Documentation	22
5.18.1.1 AddTextToLabel()	22
5.19 PlayerTexture Class Reference	23
5.19.1 Member Function Documentation	23
5.19.1.1 SetPlayerTexture()	23
5.20 ClassesAndEnums.SpawnPosition Class Reference	23
5.21 Staff Class Reference	24
5.21.1 Member Function Documentation	24
5.21.1.1 _Input()	24
5.22 Dungeon.Walker.Walker Class Reference	24
5.22.1 Constructor & Destructor Documentation	25
5.22.1.1 Walker()	25
5.22.2 Member Function Documentation	25
5.22.2.1 Shuffle< T >()	25
5.22.2.2 step()	26
5.22.2.3 walk()	26
5.23 Wall Class Reference	26
5.24 Dungeon.Generator.World Class Reference	27
5.24.1 Member Function Documentation	27
5.24.1.1 GetStatus()	27

Chapter 1

Namespace Index

1.1 Package List

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

Dungeon	7
Dungeon.Generator	7
Dungeon.Walker	7
mixchemist2	7
mixchemist2.spell	7

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

AudioStreamPlayer	
MusicManager	20
ClassesAndEnums	10
Control	
DeathMenu	12
ElementStorage	14
HealthBarControl	18
MainMenu	19
PauseMenu	21
HBoxContainer	
ElementGauge	13
HealthBar	17
Highscore	18
KinematicBody2D	
AbstractEnemy	9
Player	21
Node	
GameManager	16
Node2D	
DevScene	13
Dungeon.Generator.World	27
LevelConfiguration	19
Staff	24
Wall	26
Panel	
PlayerInputField	22
RigidBody2D	
mixchemist2.spell.ConcreteSpell	11
ClassesAndEnums.SpawnPosition	23
Sprite	
EnemyTexture	16
PlayerTexture	23
TileMap	
Dungeon.Walker.Walker	24

Chapter 3

Class Index

3.1 Class List

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

AbstractEnemy	9
ClassesAndEnums	10
mixchemist2.spell.ConcreteSpell	11
DeathMenu	12
DevScene	13
ElementGauge	13
ElementStorage	14
EnemyTexture	16
GameManager	16
HealthBar	17
HealthBarControl	18
Highscore	18
LevelConfiguration	19
MainMenu	19
MusicManager	20
PauseMenu	21
Player	21
PlayerInputField	22
PlayerTexture	23
ClassesAndEnums.SpawnPosition	23
Staff	24
Dungeon.Walker.Walker	24
Wall	26
Dungeon.Generator.World	27

Chapter 4

Namespace Documentation

4.1 Dungeon Namespace Reference

4.2 Dungeon.Generator Namespace Reference

Classes

- class [World](#)

4.3 Dungeon.Walker Namespace Reference

Classes

- class [Walker](#)

4.4 mixchemist2 Namespace Reference

4.5 mixchemist2.spell Namespace Reference

Classes

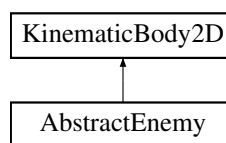
- class [ConcreteSpell](#)

Chapter 5

Class Documentation

5.1 AbstractEnemy Class Reference

Inheritance diagram for AbstractEnemy:



Public Member Functions

- override void **_Ready** ()
- override void **_Process** (float delta)
- void **Attack** ()
Function for the enemy to damage the player on collision.
- void **TargetPlayer** ()
Function for the enemy to walk into the direction of the player if it is in range.
- void **_OnBulletHit** (Node body)
Event for when the enemy is hit by a bullet.
- void **TakeDamage** (int damage, Element projectileElement, Vector2 knockback)
This method is called when the enemy takes damage.
- void **SetElement** (Element element)
Setter for the element of the enemy.

5.1.1 Member Function Documentation

5.1.1.1 _OnBulletHit()

```
void AbstractEnemy._OnBulletHit (  
    Node body)
```

Event for when the enemy is hit by a bullet.

Parameters

<i>body</i>	The body that hit the Enemy
-------------	-----------------------------

5.1.1.2 SetElement()

```
void AbstractEnemy.SetElement (
    Element element)
```

Setter for the element of the enemy.

Parameters

<i>element</i>	The element the Enemy should have
----------------	-----------------------------------

5.1.1.3 TakeDamage()

```
void AbstractEnemy.TakeDamage (
    int damage,
    Element projectileElement,
    Vector2 knockback)
```

This method is called when the enemy takes damage.

Parameters

<i>damage</i>	The amount of Damage the Enemy should take
<i>knockback</i>	The desired direction the enemy should take knockback to

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

- enemy/AbstractEnemy.cs

5.2 ClassesAndEnums Class Reference**Classes**

- class [SpawnPosition](#)

Public Types

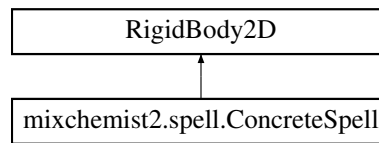
- enum **Element** {
FIRE , **WATER** , **EARTH** , **AIR** ,
FIRE_WATER , **FIRE_EARTH** , **FIRE_AIR** , **WATER_EARTH** ,
WATER_AIR , **EARTH_AIR** , **FIRE_WATER_EARTH** , **FIRE_EARTH_AIR** ,
WATER_EARTH_AIR , **FIRE_WATER_AIR** , **SHADOW** }

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

- manager/ClassesAndEnums.cs

5.3 mixchemist2.spell.ConcreteSpell Class Reference

Inheritance diagram for mixchemist2.spell.ConcreteSpell:



Public Member Functions

- override void **_Ready** ()
- override void **_Process** (float delta)
- ClassesAndEnums.Element **GetElement** ()
Get the element of the spell.
- void **SetElement** (ClassesAndEnums.Element element)
Set the element of the spell.
- int **GetDamage** ()
Get the damage of the spell.
- Vector2 **GetPosition** ()
Get the position of the spell.

Public Attributes

- int **Damage**
- ClassesAndEnums.Element **ElementType**

Properties

- Vector2 **Direction** [get, set]

5.3.1 Member Function Documentation

5.3.1.1 GetDamage()

```
int mixchemist2.spell.ConcreteSpell.GetDamage ()
```

Get the damage of the spell.

Returns

The amount of damage the spell does

5.3.1.2 GetElement()

```
ClassesAndEnums.Element mixchemist2.spell.ConcreteSpell.GetElement ()
```

Get the element of the spell.

Returns

The element of the spell

5.3.1.3 GetPosition()

```
Vector2 mixchemist2.spell.ConcreteSpell.GetPosition ()
```

Get the position of the spell.

Returns

The position of the spell

5.3.1.4 SetElement()

```
void mixchemist2.spell.ConcreteSpell.SetElement (  
    ClassesAndEnums.Element element)
```

Set the element of the spell.

Parameters

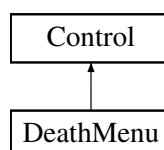
<i>element</i>	The element the spell should have
----------------	-----------------------------------

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

- spell/ConcreteSpell.cs

5.4 DeathMenu Class Reference

Inheritance diagram for DeathMenu:



Public Member Functions

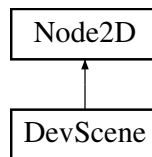
- override void **_Ready** ()

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

- UI/DeathMenu.cs

5.5 DevScene Class Reference

Inheritance diagram for DevScene:



Public Member Functions

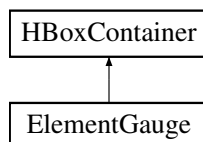
- override void **_Ready** ()

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

- devTesting/DevScene.cs

5.6 ElementGauge Class Reference

Inheritance diagram for ElementGauge:



Public Member Functions

- override void **_Ready** ()
- void **UpdateGauge** (Element element, int value)
Function to update the gauge of a specific element.

5.6.1 Member Function Documentation

5.6.1.1 UpdateGauge()

```
void ElementGauge.UpdateGauge (  
    Element element,  
    int value)
```

Function to update the gauge of a specific element.

Parameters

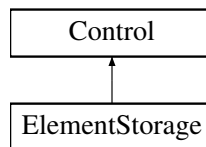
<i>element</i>	The element that should be updated
<i>value</i>	The new value it should have

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

- UI/ElementGauge.cs

5.7 ElementStorage Class Reference

Inheritance diagram for ElementStorage:



Public Member Functions

- override void **_Ready** ()
- void **ToggleElementPanelColor** (ClassesAndEnums.Element element, bool setColor)
Sets the colours of the element panels to the default colour.
- void **StoreSpellColor** (Element chosenElement)
The function to store the color of the spell in the queue.
- override void **_Input** (InputEvent @event)
Event handler for when the rotate button is pressed.
- Element **GetFirstRealmElement** ()
Gets the first element in the queue.
- ColorRect **CastFirstElementInStorage** ()
Return to cast the first element in the queue.
- override void **_Process** (float delta)

5.7.1 Member Function Documentation

5.7.1.1 _Input()

```

override void ElementStorage._Input (
    InputEvent @ event)
  
```

Event handler for when the rotate button is pressed.

Parameters

<i>event</i>	The event that needs to be handled
--------------	------------------------------------

5.7.1.2 CastFirstElementInStorage()

```
ColorRect ElementStorage.CastFirstElementInStorage ()
```

Return to cast the first element in the queue.

Returns

ColorRect of the Element that got cast

5.7.1.3 GetFirstRealmElement()

```
Element ElementStorage.GetFirstRealmElement ()
```

Gets the first element in the queue.

Returns

Returns the first element of the queue

5.7.1.4 StoreSpellColor()

```
void ElementStorage.StoreSpellColor (  
    Element chosenElement)
```

The function to store the color of the spell in the queue.

Parameters

<i>chosenElement</i>	Element that should be set
----------------------	----------------------------

5.7.1.5 ToggleElementPanelColor()

```
void ElementStorage.ToggleElementPanelColor (  
    ClassesAndEnums.Element element,  
    bool setColor)
```

Sets the colours of the element panels to the default colour.

Parameters

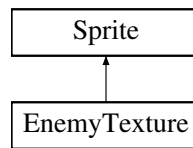
<i>element</i>	The element that needs to be set
<i>setColor</i>	Bool if the color should be set

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

- UI/ElementStorage.cs

5.8 EnemyTexture Class Reference

Inheritance diagram for EnemyTexture:



Public Member Functions

- override void **_Ready** ()
- void [SetEnemyTexture](#) (Element element)
Setter for the Enemy Texture.

5.8.1 Member Function Documentation

5.8.1.1 SetEnemyTexture()

```
void EnemyTexture.SetEnemyTexture (  
    Element element)
```

Setter for the Enemy Texture.

Parameters

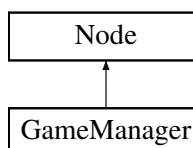
<i>element</i>	The element of the enemy
----------------	--------------------------

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

- enemy/EnemyTexture.cs

5.9 GameManager Class Reference

Inheritance diagram for GameManager:



Public Member Functions

- override void **_Ready** ()
- void **SetScore** (int score)
Set the score of the player.
- int **GetScore** ()
Gets the current score of the player.
- override void **_EnterTree** ()

Properties

- static **GameManager Instance** [get]
- List< Element > **AllowedBasicElements** [get, set]
Getter and Setter for the allowed basic elements.

5.9.1 Member Function Documentation**5.9.1.1 GetScore()**

```
int GameManager.GetScore ()
```

Gets the current score of the player.

Returns

Integer that is the score of the player

5.9.1.2 SetScore()

```
void GameManager.SetScore (
    int score)
```

Set the score of the player.

Parameters

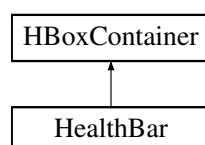
<i>score</i>	The score that should be set
--------------	------------------------------

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

- manager/GameManager.cs

5.10 HealthBar Class Reference

Inheritance diagram for HealthBar:



Public Member Functions

- override void **_Ready** ()
- void [UpdateHealthBar](#) (int currentHP)
Updates the health bar with the current HP value.

5.10.1 Member Function Documentation

5.10.1.1 UpdateHealthBar()

```
void HealthBar.UpdateHealthBar (  
    int currentHP)
```

Updates the health bar with the current HP value.

Parameters

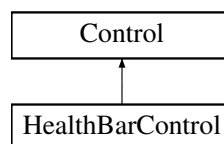
<i>currentHP</i>	The new health value of the player
------------------	------------------------------------

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

- UI/HealthBar.cs

5.11 HealthBarControl Class Reference

Inheritance diagram for HealthBarControl:



Public Member Functions

- override void **_Ready** ()

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

- enemy/HealthBarControl.cs

5.12 Highscore Class Reference

Public Member Functions

- **Highscore** (string name, int score)

Public Attributes

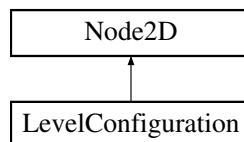
- string **Name**
- int **Score**

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

- JsonClasses/Highscore.cs

5.13 LevelConfiguration Class Reference

Inheritance diagram for LevelConfiguration:



Public Member Functions

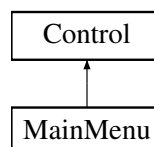
- override void **_Ready** ()
- override void **_Process** (float delta)

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

- level/LevelConfiguration.cs

5.14 MainMenu Class Reference

Inheritance diagram for MainMenu:

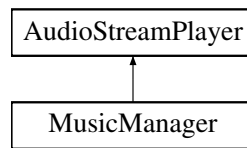


The documentation for this class was generated from the following file:

- UI/MainMenu.cs

5.15 MusicManager Class Reference

Inheritance diagram for MusicManager:



Public Member Functions

- override void [_Ready](#) ()
- void **SetVolume** (float db)
Changes Music Volume in decibels. Optimal range seems to be -80db to -10db.
- void [ChangeStream](#) (string path)
Plays the music at the given path.

Properties

- static [MusicManager Instance](#) [get]

5.15.1 Member Function Documentation

5.15.1.1 [_Ready\(\)](#)

```
override void MusicManager._Ready ()
```

Creates Singleton Instance and Starts the [MainMenu](#) song

5.15.1.2 [ChangeStream\(\)](#)

```
void MusicManager.ChangeStream (  
    string path)
```

Plays the music at the given path.

Parameters

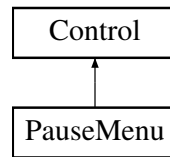
<i>path</i>	The path to the music file
-------------	----------------------------

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

- `manager/MusicManager.cs`

5.16 PauseMenu Class Reference

Inheritance diagram for PauseMenu:



Public Member Functions

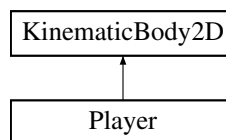
- override void **_Ready** ()
- override void **_Input** (InputEvent @event)

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

- UI/PauseMenu.cs

5.17 Player Class Reference

Inheritance diagram for Player:



Public Member Functions

- override void **_Ready** ()
- override void **_Process** (float delta)
- void **TakeDamage** (int dmgAmount, Vector2 damageVector)

Function to take damage from the enemy.

5.17.1 Member Function Documentation

5.17.1.1 TakeDamage()

```

void Player.TakeDamage (
    int   dmgAmount,
    Vector2 damageVector)
  
```

Function to take damage from the enemy.

Parameters

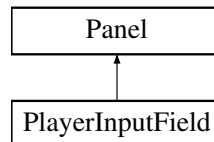
<i>dmgAmount</i>	Amount of damage that should be taken
<i>damageVector</i>	The vector from where the damage is taken from for knockback

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

- player/Player.cs

5.18 PlayerInputField Class Reference

Inheritance diagram for PlayerInputField:

**Public Member Functions**

- override void **_Ready** ()
- void [AddTextToLabel](#) (string text)
Adds text to the label.
- void **ResetLabelText** ()
Resets the Label Text.

5.18.1 Member Function Documentation

5.18.1.1 AddTextToLabel()

```
void PlayerInputField.AddTextToLabel (
    string text)
```

Adds text to the label.

Parameters

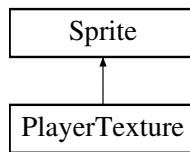
<i>text</i>	The text that should be added to the label
-------------	--

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

- UI/PlayerInputField.cs

5.19 PlayerTexture Class Reference

Inheritance diagram for PlayerTexture:



Public Member Functions

- override void **_Ready** ()
- void **SetPlayerTexture** (double hp, int maxHp)
Sets the player texture based on the current health of the player.

5.19.1 Member Function Documentation

5.19.1.1 SetPlayerTexture()

```
void PlayerTexture.SetPlayerTexture (
    double hp,
    int maxHp)
```

Sets the player texture based on the current health of the player.

Parameters

<i>hp</i>	The health of the player
<i>maxHp</i>	The maximum amount of health of the player

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

- player/PlayerTexture.cs

5.20 ClassesAndEnums.SpawnPosition Class Reference

Public Attributes

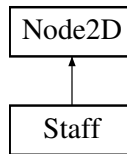
- Godot.Vector2 **Vector**
- bool **Valid** = false

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

- manager/ClassesAndEnums.cs

5.21 Staff Class Reference

Inheritance diagram for Staff:



Public Member Functions

- override void **_Ready** ()
- override void **_Input** (InputEvent @event)
Event based Input function for the player.
- override void **_Process** (float delta)

5.21.1 Member Function Documentation

5.21.1.1 _Input()

```
override void Staff._Input (  
    InputEvent @ event)
```

Event based Input function for the player.

Parameters

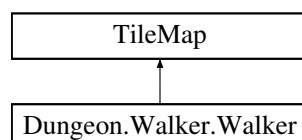
<i>event</i>	Input Event from the player
--------------	-----------------------------

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

- player/Staff.cs

5.22 Dungeon.Walker.Walker Class Reference

Inheritance diagram for Dungeon.Walker.Walker:



Public Member Functions

- [Walker](#) (Godot.Vector2 startingPosition, Godot.Rect2 new_borders)
A walker that walks in a 2D space.
- List< Godot.Vector2 > [walk](#) (int steps)
A method that makes the walker walk in a 2D space.
- bool [step](#) ()
One step of the walker.
- void **change_direction** ()
Changes the direction the walker walks to.

Static Public Member Functions

- static void [Shuffle](#)< T > (List< T > list)
Helper Method to shuffle a list.

5.22.1 Constructor & Destructor Documentation

5.22.1.1 Walker()

```
Dungeon.Walker.Walker.Walker (
    Godot.Vector2 startingPosition,
    Godot.Rect2 new_borders)
```

A walker that walks in a 2D space.

Parameters

<i>startingPosition</i>	The starting position of the Walker
<i>new_borders</i>	The border of the Map

5.22.2 Member Function Documentation

5.22.2.1 Shuffle< T >()

```
static void Dungeon.Walker.Walker.Shuffle< T > (
    List< T > list) [static]
```

Helper Method to shuffle a list.

Parameters

<i>list</i>	the list that has to be shuffled
-------------	----------------------------------

Template Parameters

<i>T</i>	
----------	--

5.22.2.2 step()

```
bool Dungeon.Walker.Walker.step ()
```

One step of the walker.

Returns

A boolean if the step was taken

5.22.2.3 walk()

```
List< Godot.Vector2 > Dungeon.Walker.Walker.walk (
    int steps)
```

A method that makes the walker walk in a 2D space.

Parameters

<i>steps</i>	The amount of steps the walker should take
--------------	--

Returns

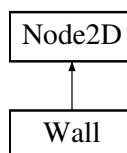
stepHistory is a list of Steps the walker has taken

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

- Dungeon/Walker.cs

5.23 Wall Class Reference

Inheritance diagram for Wall:



Public Member Functions

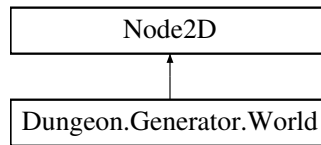
- override void **_Ready** ()
- void **_OnWallCollided** (Node body)

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

- devTesting/sceneAssets/Wall.cs

5.24 Dungeon.Generator.World Class Reference

Inheritance diagram for Dungeon.Generator.World:



Public Member Functions

- override void **_Ready** ()
- void **generate_level** ()
Lets the walker generate a level.
- bool **GetStatus** ()
Getters for the status of the generation.

5.24.1 Member Function Documentation

5.24.1.1 GetStatus()

```
bool Dungeon.Generator.World.GetStatus ()
```

Getters for the status of the generation.

Returns

boolean if the generation is done

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

- Dungeon/World.cs

Index

- [_Input](#)
 - [ElementStorage, 14](#)
 - [Staff, 24](#)
 - [_OnBulletHit](#)
 - [AbstractEnemy, 9](#)
 - [_Ready](#)
 - [MusicManager, 20](#)
- [AbstractEnemy, 9](#)
 - [_OnBulletHit, 9](#)
 - [SetElement, 10](#)
 - [TakeDamage, 10](#)
- [AddTextToLabel](#)
 - [PlayerInputField, 22](#)
- [CastFirstElementInStorage](#)
 - [ElementStorage, 14](#)
- [ChangeStream](#)
 - [MusicManager, 20](#)
- [ClassesAndEnums, 10](#)
- [ClassesAndEnums.SpawnPosition, 23](#)
- [DeathMenu, 12](#)
- [DevScene, 13](#)
- [Dungeon, 7](#)
- [Dungeon.Generator, 7](#)
- [Dungeon.Generator.World, 27](#)
 - [GetStatus, 27](#)
- [Dungeon.Walker, 7](#)
- [Dungeon.Walker.Walker, 24](#)
 - [Shuffle< T >, 25](#)
 - [step, 25](#)
 - [walk, 26](#)
 - [Walker, 25](#)
- [ElementGauge, 13](#)
 - [UpdateGauge, 13](#)
- [ElementStorage, 14](#)
 - [_Input, 14](#)
 - [CastFirstElementInStorage, 14](#)
 - [GetFirstRealmElement, 15](#)
 - [StoreSpellColor, 15](#)
 - [ToggleElementPanelColor, 15](#)
- [EnemyTexture, 16](#)
 - [SetEnemyTexture, 16](#)
- [GameManager, 16](#)
 - [GetScore, 17](#)
 - [SetScore, 17](#)
- [GetDamage](#)
 - [mixchemist2.spell.ConcreteSpell, 11](#)
- [GetElement](#)
 - [mixchemist2.spell.ConcreteSpell, 11](#)
- [GetFirstRealmElement](#)
 - [ElementStorage, 15](#)
- [GetPosition](#)
 - [mixchemist2.spell.ConcreteSpell, 12](#)
- [GetScore](#)
 - [GameManager, 17](#)
- [GetStatus](#)
 - [Dungeon.Generator.World, 27](#)
- [HealthBar, 17](#)
 - [UpdateHealthBar, 18](#)
- [HealthBarControl, 18](#)
- [Highscore, 18](#)
- [LevelConfiguration, 19](#)
- [MainMenu, 19](#)
- [mixchemist2, 7](#)
- [mixchemist2.spell, 7](#)
- [mixchemist2.spell.ConcreteSpell, 11](#)
 - [GetDamage, 11](#)
 - [GetElement, 11](#)
 - [GetPosition, 12](#)
 - [SetElement, 12](#)
- [MusicManager, 20](#)
 - [_Ready, 20](#)
 - [ChangeStream, 20](#)
- [PauseMenu, 21](#)
- [Player, 21](#)
 - [TakeDamage, 21](#)
- [PlayerInputField, 22](#)
 - [AddTextToLabel, 22](#)
- [PlayerTexture, 23](#)
 - [SetPlayerTexture, 23](#)
- [SetElement](#)
 - [AbstractEnemy, 10](#)
 - [mixchemist2.spell.ConcreteSpell, 12](#)
- [SetEnemyTexture](#)
 - [EnemyTexture, 16](#)
- [SetPlayerTexture](#)
 - [PlayerTexture, 23](#)
- [SetScore](#)
 - [GameManager, 17](#)
- [Shuffle< T >](#)
 - [Dungeon.Walker.Walker, 25](#)
- [Staff, 24](#)
 - [_Input, 24](#)

step

Dungeon.Walker.Walker, [25](#)

StoreSpellColor

ElementStorage, [15](#)

TakeDamage

AbstractEnemy, [10](#)

Player, [21](#)

ToggleElementPanelColor

ElementStorage, [15](#)

UpdateGauge

ElementGauge, [13](#)

UpdateHealthBar

HealthBar, [18](#)

walk

Dungeon.Walker.Walker, [26](#)

Walker

Dungeon.Walker.Walker, [25](#)

Wall, [26](#)