

MB2D

Generated by Doxygen 1.8.12

Contents

1	Namespace Index	2
1.1	Namespace List	2
2	Hierarchical Index	2
2.1	Class Hierarchy	2
3	Class Index	4
3.1	Class List	4
4	Namespace Documentation	8
4.1	MB2D Namespace Reference	8
4.1.1	Enumeration Type Documentation	9
4.2	MB2D.Collision Namespace Reference	10
4.3	MB2D.EntityComponent Namespace Reference	10
4.4	MB2D.Geometry Namespace Reference	11
4.5	MB2D.IO Namespace Reference	11
4.5.1	Enumeration Type Documentation	12
4.6	MB2D.Scenes Namespace Reference	12
4.6.1	Enumeration Type Documentation	13
4.7	MB2D.Testing Namespace Reference	13
4.8	MB2D.Tiles Namespace Reference	13
4.9	MB2D.UI Namespace Reference	13
4.9.1	Enumeration Type Documentation	14

5	Class Documentation	14
5.1	MB2D.UI.Button Class Reference	14
5.1.1	Detailed Description	17
5.1.2	Constructor & Destructor Documentation	17
5.1.3	Member Function Documentation	17
5.1.4	Property Documentation	18
5.1.5	Event Documentation	18
5.2	MB2D.Collectable Class Reference	18
5.2.1	Detailed Description	19
5.2.2	Constructor & Destructor Documentation	19
5.2.3	Member Function Documentation	19
5.2.4	Property Documentation	21
5.3	MB2D.Collision.CollisionCell Class Reference	21
5.3.1	Detailed Description	22
5.3.2	Constructor & Destructor Documentation	22
5.3.3	Member Function Documentation	22
5.3.4	Property Documentation	23
5.4	MB2D.EntityComponent.CollisionComponent Class Reference	24
5.4.1	Detailed Description	25
5.4.2	Constructor & Destructor Documentation	25
5.4.3	Property Documentation	25
5.5	MB2D.Collision.CollisionMap Class Reference	26
5.5.1	Detailed Description	27
5.5.2	Constructor & Destructor Documentation	27
5.5.3	Member Function Documentation	27
5.5.4	Property Documentation	30
5.6	MB2D.Testing.CollisionRenderSystem Class Reference	31
5.6.1	Member Function Documentation	32
5.7	MB2D.EntityComponent.CollisionSystem Class Reference	33
5.7.1	Detailed Description	35

5.7.2	Constructor & Destructor Documentation	35
5.7.3	Member Function Documentation	35
5.7.4	Property Documentation	38
5.8	MB2D.IO.Command Class Reference	38
5.8.1	Detailed Description	39
5.8.2	Constructor & Destructor Documentation	39
5.8.3	Member Function Documentation	39
5.8.4	Property Documentation	40
5.9	MB2D.IO.ConsoleCommand Class Reference	41
5.9.1	Detailed Description	42
5.9.2	Constructor & Destructor Documentation	42
5.9.3	Member Function Documentation	42
5.10	MB2D.EntityComponent.Depth Class Reference	43
5.10.1	Detailed Description	44
5.11	MB2D.EntityComponent.DepthSystem Class Reference	44
5.11.1	Detailed Description	46
5.11.2	Constructor & Destructor Documentation	47
5.11.3	Member Function Documentation	47
5.12	MB2D.EntityComponent.Entity Class Reference	47
5.12.1	Detailed Description	48
5.12.2	Constructor & Destructor Documentation	48
5.12.3	Member Function Documentation	49
5.12.4	Property Documentation	52
5.13	MB2D.Testing.EntityContainerTests Struct Reference	53
5.14	MB2D.EntityComponent.EntityMap Class Reference	53
5.14.1	Detailed Description	55
5.14.2	Constructor & Destructor Documentation	55
5.14.3	Member Function Documentation	55
5.14.4	Property Documentation	59
5.15	MB2D.EntityComponent.EntitySystem Class Reference	60

5.15.1 Detailed Description	62
5.15.2 Constructor & Destructor Documentation	62
5.15.3 Member Function Documentation	62
5.15.4 Member Data Documentation	65
5.15.5 Property Documentation	65
5.16 MB2D.Geometry.Grid Class Reference	66
5.16.1 Detailed Description	67
5.16.2 Constructor & Destructor Documentation	67
5.16.3 Property Documentation	67
5.17 MB2D.EntityComponent.IComponent Interface Reference	68
5.17.1 Detailed Description	69
5.18 MB2D.IO.InputMap Class Reference	69
5.18.1 Detailed Description	69
5.18.2 Constructor & Destructor Documentation	70
5.18.3 Member Function Documentation	70
5.18.4 Property Documentation	70
5.19 MB2D.EntityComponent.InputSystem Class Reference	71
5.19.1 Detailed Description	73
5.19.2 Member Function Documentation	73
5.20 MB2D.EntityComponent.Inventory Class Reference	74
5.20.1 Detailed Description	75
5.20.2 Property Documentation	75
5.21 MB2D.UI.Label Class Reference	75
5.21.1 Detailed Description	77
5.21.2 Constructor & Destructor Documentation	78
5.21.3 Member Function Documentation	78
5.22 MB2D.UI.Layout Class Reference	78
5.22.1 Detailed Description	81
5.22.2 Constructor & Destructor Documentation	81
5.22.3 Member Function Documentation	81

5.23 MB2D.Geometry.Line Class Reference	82
5.23.1 Detailed Description	83
5.23.2 Constructor & Destructor Documentation	83
5.23.3 Member Function Documentation	84
5.23.4 Property Documentation	85
5.24 MB2D.UI.ListControl Class Reference	85
5.24.1 Detailed Description	88
5.24.2 Constructor & Destructor Documentation	88
5.24.3 Member Function Documentation	89
5.24.4 Property Documentation	89
5.25 MB2D.MBConsole Class Reference	90
5.25.1 Detailed Description	91
5.25.2 Constructor & Destructor Documentation	92
5.25.3 Member Function Documentation	92
5.25.4 Property Documentation	98
5.26 MB2D.MBConsoleASTNode Class Reference	99
5.26.1 Detailed Description	99
5.26.2 Member Function Documentation	100
5.27 MB2D.MBConsoleLexer Class Reference	100
5.27.1 Detailed Description	101
5.27.2 Constructor & Destructor Documentation	101
5.27.3 Member Function Documentation	101
5.27.4 Property Documentation	102
5.28 MB2D.MBConsoleParser Class Reference	103
5.28.1 Detailed Description	104
5.28.2 Constructor & Destructor Documentation	104
5.28.3 Member Function Documentation	104
5.29 MB2D.MBGame Class Reference	105
5.29.1 Detailed Description	107
5.29.2 Constructor & Destructor Documentation	107

5.29.3	Member Function Documentation	107
5.29.4	Property Documentation	109
5.30	MB2D.IO.MoveBackward Class Reference	111
5.30.1	Detailed Description	112
5.30.2	Constructor & Destructor Documentation	112
5.30.3	Member Function Documentation	112
5.31	MB2D.IO.MoveDown Class Reference	113
5.31.1	Detailed Description	114
5.31.2	Constructor & Destructor Documentation	114
5.31.3	Member Function Documentation	115
5.32	MB2D.IO.MoveForward Class Reference	115
5.32.1	Detailed Description	116
5.32.2	Constructor & Destructor Documentation	116
5.32.3	Member Function Documentation	117
5.33	MB2D.IO.MoveLeft Class Reference	117
5.33.1	Detailed Description	118
5.33.2	Constructor & Destructor Documentation	118
5.33.3	Member Function Documentation	119
5.34	MB2D.EntityComponent.Movement Class Reference	119
5.34.1	Detailed Description	120
5.34.2	Constructor & Destructor Documentation	121
5.34.3	Property Documentation	121
5.35	MB2D.EntityComponent.MovementSystem Class Reference	122
5.35.1	Detailed Description	123
5.35.2	Constructor & Destructor Documentation	124
5.35.3	Member Function Documentation	124
5.36	MB2D.IO.MoveRight Class Reference	124
5.36.1	Detailed Description	126
5.36.2	Constructor & Destructor Documentation	126
5.36.3	Member Function Documentation	126

5.37 MB2D.IO.MoveUp Class Reference	127
5.37.1 Detailed Description	128
5.37.2 Constructor & Destructor Documentation	128
5.37.3 Member Function Documentation	128
5.38 MB2D.EntityComponent.PhysicsComponent Class Reference	129
5.38.1 Detailed Description	130
5.38.2 Property Documentation	130
5.39 MB2D.EntityComponent.PhysicsEnvironment Class Reference	131
5.39.1 Detailed Description	131
5.39.2 Property Documentation	131
5.40 MB2D.EntityComponent.PhysicsSystem Class Reference	132
5.40.1 Detailed Description	134
5.40.2 Constructor & Destructor Documentation	134
5.40.3 Member Function Documentation	134
5.40.4 Property Documentation	134
5.41 MB2D.EntityComponent.PlayerController Class Reference	135
5.41.1 Detailed Description	136
5.41.2 Constructor & Destructor Documentation	136
5.41.3 Property Documentation	136
5.42 MB2D.Testing.Position Class Reference	136
5.43 MB2D.PrintASTNode Class Reference	137
5.43.1 Detailed Description	139
5.43.2 Constructor & Destructor Documentation	139
5.43.3 Member Function Documentation	139
5.44 MB2D.QuitASTNode Class Reference	140
5.44.1 Detailed Description	141
5.44.2 Member Function Documentation	141
5.45 MB2D.EntityComponent.RenderSystem Class Reference	141
5.45.1 Detailed Description	143
5.45.2 Constructor & Destructor Documentation	144

5.45.3	Member Function Documentation	144
5.46	MB2D.RootASTNode Class Reference	144
5.46.1	Detailed Description	146
5.46.2	Constructor & Destructor Documentation	146
5.46.3	Member Function Documentation	146
5.47	MB2D.IO.RotateLeft Class Reference	147
5.47.1	Detailed Description	148
5.47.2	Constructor & Destructor Documentation	148
5.47.3	Member Function Documentation	149
5.48	MB2D.IO.RotateRight Class Reference	149
5.48.1	Detailed Description	150
5.48.2	Constructor & Destructor Documentation	150
5.48.3	Member Function Documentation	151
5.49	MB2D.RunASTNode Class Reference	151
5.49.1	Detailed Description	152
5.49.2	Constructor & Destructor Documentation	152
5.49.3	Member Function Documentation	153
5.50	MB2D.Scenes.Scene Class Reference	153
5.50.1	Detailed Description	156
5.50.2	Constructor & Destructor Documentation	156
5.50.3	Member Function Documentation	156
5.50.4	Property Documentation	159
5.51	MB2D.Scenes.SceneStack Class Reference	160
5.51.1	Detailed Description	161
5.51.2	Constructor & Destructor Documentation	161
5.51.3	Member Function Documentation	162
5.51.4	Property Documentation	164
5.52	MB2D.SetASTNode Class Reference	165
5.52.1	Detailed Description	166
5.52.2	Constructor & Destructor Documentation	166

5.52.3	Member Function Documentation	167
5.53	MB2D.SoundTrigger Class Reference	167
5.53.1	Detailed Description	168
5.53.2	Constructor & Destructor Documentation	168
5.53.3	Member Function Documentation	168
5.53.4	Property Documentation	169
5.54	MB2D.EntityComponent.SpriteTransform Class Reference	170
5.54.1	Detailed Description	171
5.54.2	Constructor & Destructor Documentation	171
5.54.3	Property Documentation	172
5.55	MB2D.Testing.Test Class Reference	173
5.56	MB2D.Testing.TestSystem Class Reference	174
5.56.1	Member Function Documentation	175
5.57	MB2D.Testing.TestSystem2 Class Reference	176
5.57.1	Member Function Documentation	177
5.58	MB2D.Testing.TestUIView Class Reference	178
5.59	MB2D.IO.TextInputHandler Class Reference	179
5.59.1	Property Documentation	180
5.60	MB2D.Tile Class Reference	180
5.60.1	Detailed Description	181
5.60.2	Constructor & Destructor Documentation	181
5.60.3	Property Documentation	181
5.61	MB2D.Tiles.TileMap Class Reference	182
5.61.1	Detailed Description	183
5.61.2	Constructor & Destructor Documentation	183
5.61.3	Member Function Documentation	183
5.61.4	Property Documentation	184
5.62	MB2D.UI.UIContent Class Reference	185
5.62.1	Detailed Description	186
5.62.2	Constructor & Destructor Documentation	186

5.62.3	Property Documentation	186
5.63	MB2D.UI.UIControlElement Class Reference	187
5.63.1	Detailed Description	190
5.63.2	Constructor & Destructor Documentation	190
5.63.3	Member Function Documentation	191
5.63.4	Member Data Documentation	191
5.63.5	Property Documentation	191
5.64	MB2D.UI.UIElement Class Reference	193
5.64.1	Detailed Description	195
5.64.2	Constructor & Destructor Documentation	195
5.64.3	Member Function Documentation	196
5.64.4	Property Documentation	198
5.65	MB2D.Testing.UITest Class Reference	201
5.65.1	Member Function Documentation	203
5.66	MB2D.UI.UIView Class Reference	205
5.66.1	Detailed Description	206
5.66.2	Constructor & Destructor Documentation	206
5.66.3	Member Function Documentation	206
5.66.4	Property Documentation	209
5.67	MB2D.Testing.Unregistered Class Reference	210
5.68	MB2D.EntityComponent.UtilityController Class Reference	211
5.68.1	Detailed Description	212
5.68.2	Constructor & Destructor Documentation	212
5.68.3	Property Documentation	212
5.69	MB2D.VariableASTNode Class Reference	212
5.69.1	Detailed Description	213
5.69.2	Constructor & Destructor Documentation	213
5.69.3	Property Documentation	213
5.70	MB2D.Testing.Velocity Class Reference	214

1 Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

MB2D	8
MB2D.Collision	10
MB2D.EntityComponent	10
MB2D.Geometry	11
MB2D.IO	11
MB2D.Scenes	12
MB2D.Testing	13
MB2D.Tiles	13
MB2D.UI	13

2 Hierarchical Index

2.1 Class Hierarchy

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

MB2D.Collectable	18
MB2D.Collision.CollisionCell	21
MB2D.Collision.CollisionMap	26
MB2D.IO.Command	38
MB2D.IO.ConsoleCommand	41
MB2D.IO.MoveBackward	111
MB2D.IO.MoveDown	113
MB2D.IO.MoveForward	115
MB2D.IO.MoveLeft	117
MB2D.IO.MoveRight	124
MB2D.IO.MoveUp	127
MB2D.IO.RotateLeft	147
MB2D.IO.RotateRight	149
MB2D.EntityComponent.Entity	47

MB2D.Testing.EntityContainerTests	53
MB2D.EntityComponent.EntityMap	53
MB2D.EntityComponent.EntitySystem	60
MB2D.EntityComponent.CollisionSystem	33
MB2D.EntityComponent.DepthSystem	44
MB2D.EntityComponent.InputSystem	71
MB2D.EntityComponent.MovementSystem	122
MB2D.EntityComponent.PhysicsSystem	132
MB2D.EntityComponent.RenderSystem	141
MB2D.Testing.CollisionRenderSystem	31
MB2D.Testing.TestSystem	174
MB2D.Testing.TestSystem2	176
Game	
MB2D.MBGame	105
MB2D.Geometry.Grid	66
MB2D.EntityComponent.IComponent	68
MB2D.EntityComponent.CollisionComponent	24
MB2D.EntityComponent.Depth	43
MB2D.EntityComponent.Inventory	74
MB2D.EntityComponent.Movement	119
MB2D.EntityComponent.PhysicsComponent	129
MB2D.EntityComponent.PlayerController	135
MB2D.EntityComponent.SpriteTransform	170
MB2D.EntityComponent.UtilityController	211
MB2D.Testing.Position	136
MB2D.Testing.Test	173
MB2D.Testing.Unregistered	210
MB2D.Testing.Velocity	214
MB2D.IO.InputMap	69
MB2D.Geometry.Line	82
MB2D.MBConsole	90
MB2D.MBConsoleASTNode	99

MB2D.PrintASTNode	137
MB2D.QuitASTNode	140
MB2D.RootASTNode	144
MB2D.RunASTNode	151
MB2D.SetASTNode	165
MB2D.MBConsoleLexer	100
MB2D.MBConsoleParser	103
MB2D.EntityComponent.PhysicsEnvironment	131
MB2D.Scenes.Scene	153
MB2D.Testing.UITest	201
MB2D.Scenes.SceneStack	160
MB2D.SoundTrigger	167
MB2D.IO.TextInputHandler	179
MB2D.Tile	180
MB2D.Tiles.TileMap	182
MB2D.UI.UIContent	185
MB2D.UI.UIElement	193
MB2D.UI.Label	75
MB2D.UI.Layout	78
MB2D.UI.UIControlElement	187
MB2D.UI.Button	14
MB2D.UI.ListControl	85
MB2D.UI.UIView	205
MB2D.Testing.TestUIView	178
MB2D.VariableASTNode	212

3 Class Index

3.1 Class List

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

MB2D.UI.Button	
A pressable ui element with a single OnPress event	14

MB2D.Collectable	
Defines an object that can be contained within components and systems that operate on collectible items, such as Inventory.	18
MB2D.Collision.CollisionCell	
A cell used in a collision map to hold a linked list of all its contained entities	21
MB2D.EntityComponent.CollisionComponent	
Used for running collision detection on an Entity	24
MB2D.Collision.CollisionMap	
A 2D Grid that represents a particular space in the game world to check for collisions. Uses spatial indexing to determine where an entity will be located at any given time. For best results, the cellsize and overall size of the map should be tweaked for each individual game screen and environment.	26
MB2D.Testing.CollisionRenderSystem	31
MB2D.EntityComponent.CollisionSystem	
Checks collisions. Uses a spatial indexing grid for broad-phase collision checking and AABB checks for narrow phase	33
MB2D.IO.Command	
Executes an action associated with a specific key	38
MB2D.IO.ConsoleCommand	
Shows or hides the debug console	41
MB2D.EntityComponent.Depth	
A tag class used to define an entity that should be draw sorted according to its current z-index	43
MB2D.EntityComponent.DepthSystem	
Changes an entities z-index based on the y coordinate of the top of their sprite	44
MB2D.EntityComponent.Entity	
Represents a tagged and id'd container for components that can be operated on by systems.	47
MB2D.Testing.EntityContainerTests	53
MB2D.EntityComponent.EntityMap	
Maps entities, systems and components to one another and provides querying and updating access to all elements	53
MB2D.EntityComponent.EntitySystem	
Performs logic on an entity.	60
MB2D.Geometry.Grid	
Represents a grid structure. Can be drawn via a SpriteBatch	66
MB2D.EntityComponent.IComponent	
Tags any class as a valid component for use in the EntityMap . Derived classes should contain no logic, only data fields.	68
MB2D.IO.InputMap	
Maps commands to keys and trigger types	69
MB2D.EntityComponent.InputSystem	
Processes input for PlayerController and UtilityController entities. Can operate on an entity with either or both components	71

MB2D.EntityComponent.Inventory	
Defines a dictionary of Collectable types used for entities	74
MB2D.UI.Label	
A static UIElement with a TextContent , border and optional texture	75
MB2D.UI.Layout	
A container for UIElements used within a UIView . Used to divide the View into smaller segments and to move around a group of elements easily	78
MB2D.Geometry.Line	
A line structure, can be drawn via SpriteBatch	82
MB2D.UI.ListControl	
A scrollable list box. Items can be added and interacted with.	85
MB2D.MBConsole	
Midnight Blue debug console class. Executes attached methods and changes attached variables.	90
MB2D.MBConsoleASTNode	
Class all AST nodes inherit from	99
MB2D.MBConsoleLexer	
Breaks a string into a series of tokens to use for parsing the debug consoles command language	100
MB2D.MBConsoleParser	
Parses command string input and executes it using a debug console.	103
MB2D.MBGame	
This is the main type for your game.	105
MB2D.IO.MoveBackward	
Moves an entity backward. Only runs on entities with a physics component	111
MB2D.IO.MoveDown	
Moves an entity down	113
MB2D.IO.MoveForward	
Moves an entity forward. Only runs on entities with a physics component	115
MB2D.IO.MoveLeft	
Moves an entity left.	117
MB2D.EntityComponent.Movement	
Defines position, rotation and speed related data for moving an entity.	119
MB2D.EntityComponent.MovementSystem	
Processes the change in position, rotation, and sprite transform for an entity	122
MB2D.IO.MoveRight	
Moves an entity right	124
MB2D.IO.MoveUp	
Moves a player controller up	127
MB2D.EntityComponent.PhysicsComponent	
Physics component used to define acceleration and velocity.	129

MB2D.EntityComponent.PhysicsEnvironment	
Defines a new environment to feed into the physics system to alter the impact it has on an entity	131
MB2D.EntityComponent.PhysicsSystem	
Processes physics changes for a given entity	132
MB2D.EntityComponent.PlayerController	
Defines the attached entity as controllable	135
MB2D.Testing.Position	136
MB2D.PrintASTNode	
Prints a variable to the console.	137
MB2D.QuitASTNode	
Handles quitting the game	140
MB2D.EntityComponent.RenderSystem	
Renders culled entities with a SpriteTransform to the window	141
MB2D.RootASTNode	
The entry point for command execution with a single child.	144
MB2D.IO.RotateLeft	
Rotates an entity left	147
MB2D.IO.RotateRight	
Rotates an entity right	149
MB2D.RunASTNode	
AST node entry point for executing a run command	151
MB2D.Scenes.Scene	
Holds all logic and data for a single game screen	153
MB2D.Scenes.SceneStack	
Holds the games scenes in a stack structure running the top scene every frame. Handles switching state for scenes and popping/pushing new scenes on top of one another. Allows the current scene to access other scenes.	160
MB2D.SetASTNode	
AST node representing the entry point for a 'set' command with an identifier and a value child	165
MB2D.SoundTrigger	
Triggers a sound effect	167
MB2D.EntityComponent.SpriteTransform	
Defines a sprite component with control over its size, rotation, and scale	170
MB2D.Testing.Test	173
MB2D.Testing.TestSystem	174
MB2D.Testing.TestSystem2	176
MB2D.Testing.TestUIView	178
MB2D.IO.TextInputHandler	179
MB2D.Tile	
Represents a single tile in a tile map.	180

MB2D.Tiles.TileMap	
A grid of tiles with collision. Wraps coordinates when they fall out of bounds. Allows accessing tiles by index.	182
MB2D.UI.UIContent	
Holds content in a grid structure for a UIContext or Layout	185
MB2D.UI.UIControlElement	
An interactive and controllable UIElement	187
MB2D.UI.UIElement	
Defines a UI object that can be contained within Views and Layouts, drawn, updated, and moved about	193
MB2D.Testing.UITest	201
MB2D.UI.UIView	
A single context for all UI elements and layouts.	205
MB2D.Testing.Unregistered	210
MB2D.EntityComponent.UtilityController	
Declares the attached entity as able to control utility commands such as opening the debug console	211
MB2D.VariableASTNode	
Represents a variable with a type and a value	212
MB2D.Testing.Velocity	214

4 Namespace Documentation

4.1 MB2D Namespace Reference

Namespaces

Classes

- class [Collectable](#)
Defines an object that can be contained within components and systems that operate on collectible items, such as Inventory.
- class [MBConsole](#)
Midnight Blue debug console class. Executes attached methods and changes attached variables.
- class [MBConsoleASTNode](#)
Class all AST nodes inherit from
- class [MBConsoleLexer](#)
Breaks a string into a series of tokens to use for parsing the debug consoles command language
- class [MBConsoleParser](#)
Parses command string input and executes it using a debug console.
- class [MBGame](#)
This is the main type for your game.
- class **MBMath**
Math helper class
- class [PrintASTNode](#)

- Prints a variable to the console.*
- class [QuitASTNode](#)
 - Handles quitting the game*
- class [RootASTNode](#)
 - The entry point for command execution with a single child.*
- class [RunASTNode](#)
 - AST node entry point for executing a run command*
- class [SetASTNode](#)
 - AST node representing the entry point for a 'set' command with an identifier and a value child*
- class [SoundTrigger](#)
 - Triggers a sound effect*
- class **SpriteBatchExtensions**
 - Extends SpriteBatch with MidnightBlue data structures.*
- class [Tile](#)
 - Represents a single tile in a tile map.*
- class [VariableASTNode](#)
 - Represents a variable with a type and a value*
- class **Vector2Extensions**

Enumerations

- enum [Token](#) {
[Token.Unknown](#), [Token.String](#), [Token.Set](#), [Token.Run](#),
[Token.Print](#), [Token.Quit](#) }
Category of tokens created by the lexer
- enum [TileFlag](#) { [TileFlag.Passable](#), [TileFlag.Impassable](#) }
Flags a tile as passable or impassable, used in collision checking.

4.1.1 Enumeration Type Documentation

4.1.1.1 TileFlag

```
enum MB2D.TileFlag [strong]
```

Flags a tile as passable or impassable, used in collision checking.

Enumerator

Passable	Flags a tile as walkable
Impassable	Flags a tile as collidable and unable to be walked on.

4.1.1.2 Token

```
enum MB2D.Token [strong]
```

Category of tokens created by the lexer

Enumerator

Unknown	A value or an identifier
String	A string sequence
Set	A set command statement
Run	A run command statement
Print	A print command statement
Quit	A quit command statement

4.2 MB2D.Collision Namespace Reference

Classes

- class [CollisionCell](#)
A cell used in a collision map to hold a linked list of all its contained entities
- class [CollisionMap](#)
A 2D Grid that represents a particular space in the game world to check for collisions. Uses spatial indexing to determine where an entity will be located at any given time. For best results, the cellsize and overall size of the map should be tweaked for each individual game screen and environment.

4.3 MB2D.EntityComponent Namespace Reference

Classes

- class [CollisionComponent](#)
Used for running collision detection on an [Entity](#)
- class [CollisionSystem](#)
Checks collisions. Uses a spatial indexing grid for broad-phase collision checking and AABB checks for narrow phase
- class [Depth](#)
A tag class used to define an entity that should be draw sorted according to its current z-index
- class [DepthSystem](#)
Changes an entities z-index based on the y coordinate of the top of their sprite
- class [Entity](#)
Represents a tagged and id'd container for components that can be operated on by systems.
- class [EntityMap](#)
Maps entities, systems and components to one another and provides querying and updating access to all elements
- class [EntitySystem](#)
Performs logic on an entity.
- interface [IComponent](#)
Tags any class as a valid component for use in the [EntityMap](#). Derived classes should contain no logic, only data fields.
- class [InputSystem](#)
Processes input for [PlayerController](#) and [UtilityController](#) entities. Can operate on an entity with either or both components
- class [Inventory](#)
Defines a dictionary of [Collectable](#) types used for entities
- class [Movement](#)
Defines position, rotation and speed related data for moving an entity.
- class [MovementSystem](#)

- Processes the change in position, rotation, and sprite transform for an entity*
 - class [PhysicsComponent](#)
 - Physics component used to define acceleration and velocity.*
 - class [PhysicsEnvironment](#)
 - Defines a new environment to feed into the physics system to alter the impact it has on an entity*
 - class [PhysicsSystem](#)
 - Processes physics changes for a given entity*
 - class [PlayerController](#)
 - Defines the attached entity as controllable*
 - class [RenderSystem](#)
 - Renders culled entities with a [SpriteTransform](#) to the window*
 - class [SpriteTransform](#)
 - Defines a sprite component with control over its size, rotation, and scale*
 - class [UtilityController](#)
 - Declares the attached entity as able to control utility commands such as opening the debug console*

Enumerations

- enum **EntityAssociation** { **Strict**, **Loose** }

4.4 MB2D.Geometry Namespace Reference

Classes

- class [Grid](#)
 - Represents a grid structure. Can be drawn via a [SpriteBatch](#)*
- class [Line](#)
 - A line structure, can be drawn via [SpriteBatch](#)*

4.5 MB2D.IO Namespace Reference

Classes

- class [Command](#)
 - Executes an action associated with a specific key*
- class [ConsoleCommand](#)
 - Shows or hides the debug console*
- class [InputMap](#)
 - Maps commands to keys and trigger types*
- class **IOUtil**
 - Utility methods for working with the keyboard and mouse*
- class [MoveBackward](#)
 - Moves an entity backward. Only runs on entities with a physics component*
- class [MoveDown](#)
 - Moves an entity down*
- class [MoveForward](#)
 - Moves an entity forward. Only runs on entities with a physics component*
- class [MoveLeft](#)

- Moves an entity left.*
 - class [MoveRight](#)
 - Moves an entity right*
 - class [MoveUp](#)
 - Moves a player controller up*
 - class [RotateLeft](#)
 - Rotates an entity left*
 - class [RotateRight](#)
 - Rotates an entity right*
 - class [TextInputHandler](#)

Enumerations

- enum [CommandType](#) { [CommandType.Hold](#), [CommandType.Trigger](#) }
Represents either a trigger or hold command type

4.5.1 Enumeration Type Documentation

4.5.1.1 CommandType

```
enum MB2D.IO.CommandType [strong]
```

Represents either a trigger or hold command type

Enumerator

Hold	Execute command every frame its associated input key/button is detected
Trigger	Execute command only on the first frame its associated input key/button is detected and don't execute again until it's released and pressed again

4.6 MB2D.Scenes Namespace Reference

Classes

- class [Scene](#)
Holds all logic and data for a single game screen
- class [SceneStack](#)
Holds the games scenes in a stack structure running the top scene every frame. Handles switching state for scenes and popping/pushing new scenes on top of one another. Allows the current scene to access other scenes.

Enumerations

- enum [TransitionState](#) {
[TransitionState.Null](#), [TransitionState.None](#), [TransitionState.Pausing](#), [TransitionState.Resuming](#),
[TransitionState.Exiting](#), [TransitionState.Initializing](#) }
Defines a valid transition state to move into. Once set, the scene stack will automatically move the current scene into that state the next frame.

4.6.1 Enumeration Type Documentation

4.6.1.1 TransitionState

```
enum MB2D.Scenes.TransitionState [strong]
```

Defines a valid transition state to move into. Once set, the scene stack will automatically move the current scene into that state the next frame.

Enumerator

Null	The scene hasn't initialized yet
None	The normal state
Pausing	The scene is currently pausing. Set state to None to end transition.
Resuming	The scene is resuming from the paused state. Set state to None to end transition.
Exiting	The scene is exiting to be destroyed. Set state to Null to end transition.
Initializing	The scene is initializing from the an unconstructed state. Set state to None to end.

4.7 MB2D.Testing Namespace Reference

Classes

- class [CollisionRenderSystem](#)
- struct [EntityContainerTests](#)
- class [Position](#)
- class [Test](#)
- class [TestSystem](#)
- class [TestSystem2](#)
- class [TestUIView](#)
- class [UITest](#)
- class [Unregistered](#)
- class [Velocity](#)

4.8 MB2D.Tiles Namespace Reference

Classes

- class [TileMap](#)
A grid of tiles with collision. Wraps coordinates when they fall out of bounds. Allows accessing tiles by index.

4.9 MB2D.UI Namespace Reference

Classes

- class [Button](#)
A pressable ui element with a single OnPress event
- class [Label](#)
A static [UIElement](#) with a [TextContent](#), border and optional texture

- class [Layout](#)
A container for [UIElements](#) used within a [UIView](#). Used to divide the View into smaller segments and to move around a group of elements easily
- class [ListControl](#)
A scrollable list box. Items can be added and interacted with.
- class [UIContent](#)
Holds content in a grid structure for a [UIContext](#) or [Layout](#)
- class [UIControlElement](#)
An interactive and controllable [UIElement](#)
- class [UIElement](#)
Defines a [UI](#) object that can be contained within Views and Layouts, drawn, updated, and moved about
- class [UIView](#)
A single context for all [UI](#) elements and layouts.

Enumerations

- enum [UIState](#) { [UIState.Normal](#), [UIState.Selected](#), [UIState.Pressed](#) }
Represents the current state of a controllable [UIElement](#)

4.9.1 Enumeration Type Documentation

4.9.1.1 UIState

```
enum MB2D.UI.UIState [strong]
```

Represents the current state of a controllable [UIElement](#)

Enumerator

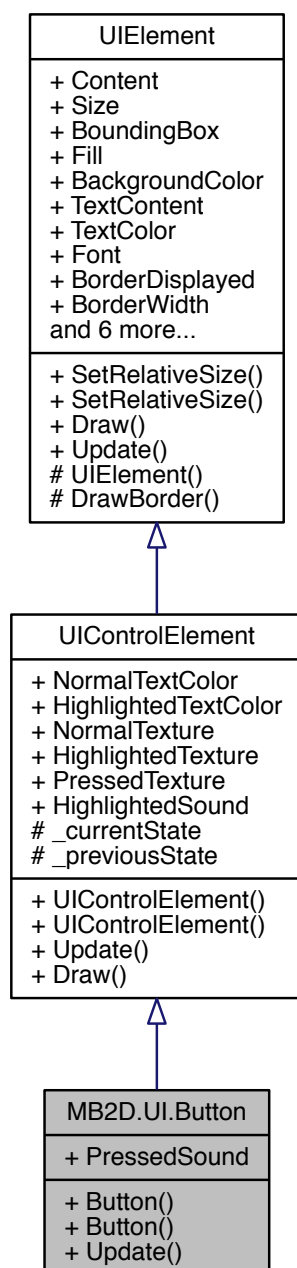
Normal	Unselected, unpressed state
Selected	Hovered or highlighted state
Pressed	Clicked or pressed state

5 Class Documentation

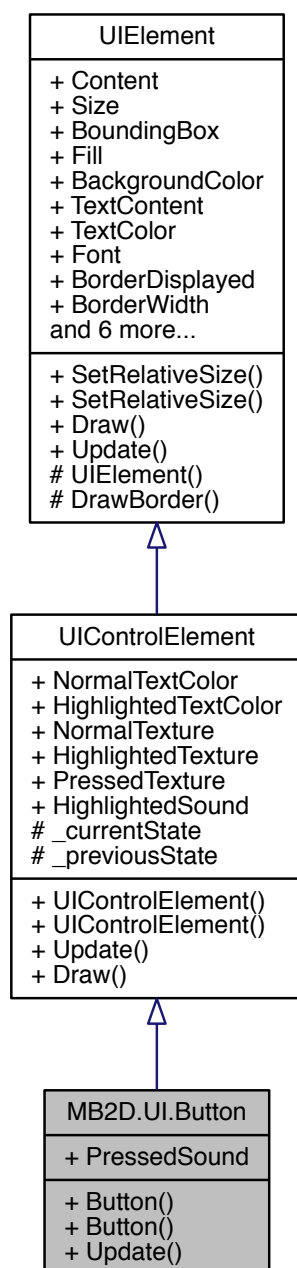
5.1 MB2D.UI.Button Class Reference

A pressable ui element with a single OnPress event

Inheritance diagram for MB2D.UI.Button:



Collaboration diagram for MB2D.UI.Button:



Public Member Functions

- **Button** (Texture2D normal, Texture2D selected, Texture2D pressed)
Initializes a new instance of the T:MB2D.UI.Button class.
- **Button** ()
Initializes a new instance of the T:MB2D.UI.Button class with no associated textures
- override void **Update** ()
Updates the button state.

Properties

- SoundEffectInstance [PressedSound](#) [get, set]
Gets or sets the sound fired when transitioning to the pressed state.

Events

- EventHandler [OnPress](#)
Occurs when the button has been clicked or pressed.

Additional Inherited Members

5.1.1 Detailed Description

A pressable ui element with a single OnPress event

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Button() [1/2]

```
MB2D.UI.Button.Button (
    Texture2D normal,
    Texture2D selected,
    Texture2D pressed ) [inline]
```

Initializes a new instance of the T:MB2D.UI.Button class.

Parameters

<i>normal</i>	Normal state texture
<i>selected</i>	Selected state texture.
<i>pressed</i>	Pressed state texture.

5.1.2.2 Button() [2/2]

```
MB2D.UI.Button.Button ( ) [inline]
```

Initializes a new instance of the T:MB2D.UI.Button class with no associated textures

5.1.3 Member Function Documentation

5.1.3.1 Update()

```
override void MB2D.UI.Button.Update ( ) [inline], [virtual]
```

Updates the button state.

Implements [MB2D.UI.UIElement](#).

5.1.4 Property Documentation

5.1.4.1 PressedSound

`SoundEffectInstance MB2D.UI.Button.PressedSound [get], [set]`

Gets or sets the sound fired when transitioning to the pressed state.

The pressed state sound.

5.1.5 Event Documentation

5.1.5.1 OnPress

`EventHandler MB2D.UI.Button.OnPress`

Occurs when the button has been clicked or pressed.

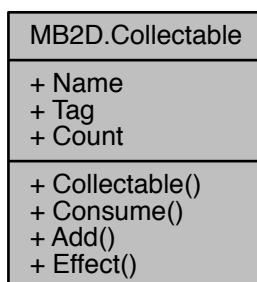
The documentation for this class was generated from the following file:

- MB2D/src/UI/Button.cs

5.2 MB2D.Collectable Class Reference

Defines an object that can be contained within components and systems that operate on collectible items, such as Inventory.

Collaboration diagram for MB2D.Collectable:



Public Member Functions

- [Collectable](#) (string name, string tag, int initialCount)
Initializes a new instance of the T:MB2D.Collectable class.
- void [Consume](#) (int amount=1)
Consumes a number of instances of the item
- void [Add](#) (int amount=1)
Adds a number of instances of this item to the container
- abstract void [Effect](#) ([Entity](#) entity)
The action to enact when the item is consumed or used

Properties

- string [Name](#) [get]
Gets the name of the item.
- string [Tag](#) [get]
Gets the items tag.
- int [Count](#) [get]
Gets the count of available instances of the item.

5.2.1 Detailed Description

Defines an object that can be contained within components and systems that operate on collectible items, such as Inventory.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 Collectable()

```
MB2D.Collectable.Collectable (
    string name,
    string tag,
    int initialCount ) [inline]
```

Initializes a new instance of the T:MB2D.Collectable class.

Parameters

<i>name</i>	Name to give to the item.
<i>tag</i>	Short tag to give to the item.
<i>initialCount</i>	Initial count to add to the container.

5.2.3 Member Function Documentation

5.2.3.1 Add()

```
void MB2D.Collectable.Add (
    int amount = 1 ) [inline]
```

Adds a number of instances of this item to the container

Parameters

<i>amount</i>	Amount to add.
---------------	----------------

Here is the call graph for this function:



5.2.3.2 Consume()

```
void MB2D.Collectable.Consume (
    int amount = 1 ) [inline]
```

Consumes a number of instances of the item

Parameters

<i>amount</i>	Amount to consume.
---------------	--------------------

5.2.3.3 Effect()

```
abstract void MB2D.Collectable.Effect (
    Entity entity ) [pure virtual]
```

The action to enact when the item is consumed or used

Parameters

<i>entity</i>	Entity to operate on.
---------------	-----------------------

Here is the caller graph for this function:



5.2.4 Property Documentation

5.2.4.1 Count

```
int MB2D.Collectable.Count [get]
```

Gets the count of available instances of the item.

The count.

5.2.4.2 Name

```
string MB2D.Collectable.Name [get]
```

Gets the name of the item.

The name.

5.2.4.3 Tag

```
string MB2D.Collectable.Tag [get]
```

Gets the items tag.

The tag.

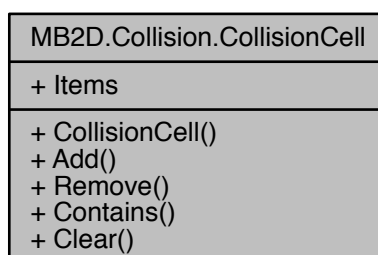
The documentation for this class was generated from the following file:

- MB2D/src/Inventory/Collectable.cs

5.3 MB2D.Collision.CollisionCell Class Reference

A cell used in a collision map to hold a linked list of all its contained entities

Collaboration diagram for MB2D.Collision.CollisionCell:



Public Member Functions

- [CollisionCell](#) ()
Initializes a new instance of the T:MB2D.Collision.CollisionCell class.
- void [Add](#) ([Entity](#) entity)
Adds an entity to the cell
- void [Remove](#) ([Entity](#) entity)
Removes a specific entity from the cell
- bool [Contains](#) ([Entity](#) entity)
Checks if the entity is inside the cell already
- void [Clear](#) ()
Clear the cell of all entities.

Properties

- [LinkedList< Entity > Items](#) [get]
Gets the list of this cells entities

5.3.1 Detailed Description

A cell used in a collision map to hold a linked list of all its contained entities

5.3.2 Constructor & Destructor Documentation

5.3.2.1 CollisionCell()

```
MB2D.Collision.CollisionCell.CollisionCell ( ) [inline]
```

Initializes a new instance of the T:MB2D.Collision.CollisionCell class.

5.3.3 Member Function Documentation

5.3.3.1 Add()

```
void MB2D.Collision.CollisionCell.Add (
    Entity entity ) [inline]
```

Adds an entity to the cell

Parameters

<i>entity</i>	Entity to add.
---------------	----------------

Here is the caller graph for this function:



5.3.3.2 Clear()

```
void MB2D.Collision.CollisionCell.Clear ( ) [inline]
```

Clear the cell of all entities.

5.3.3.3 Contains()

```
bool MB2D.Collision.CollisionCell.Contains (
    Entity entity ) [inline]
```

Checks if the entity is inside the cell already

Parameters

<i>entity</i>	Entity.
---------------	---------

5.3.3.4 Remove()

```
void MB2D.Collision.CollisionCell.Remove (
    Entity entity ) [inline]
```

Removes a specific entity from the cell

Parameters

<i>entity</i>	Entity to remove.
---------------	-------------------

5.3.4 Property Documentation

5.3.4.1 Items

```
LinkedList<Entity> MB2D.Collision.CollisionCell.Items [get]
```

Gets the list of this cells entities

The entities.

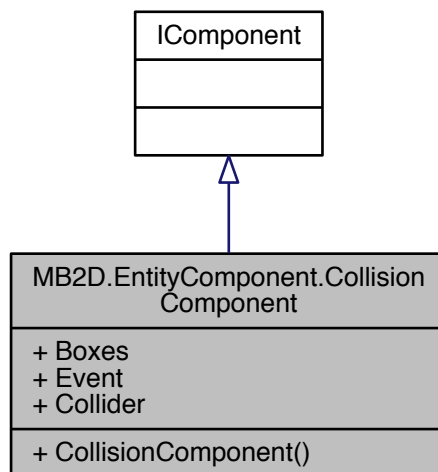
The documentation for this class was generated from the following file:

- MB2D/src/Collision/CollisionCell.cs

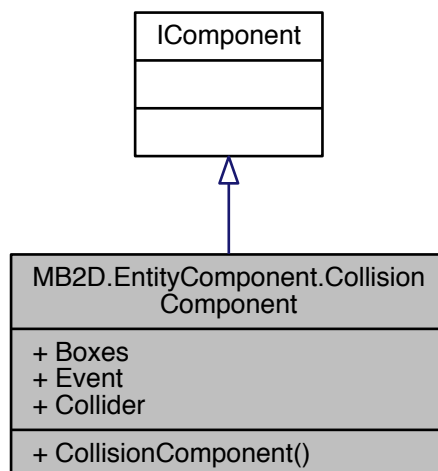
5.4 MB2D.EntityComponent.CollisionComponent Class Reference

Used for running collision detection on an [Entity](#)

Inheritance diagram for MB2D.EntityComponent.CollisionComponent:



Collaboration diagram for MB2D.EntityComponent.CollisionComponent:



Public Member Functions

- [CollisionComponent](#) (params RectangleF[] boxes)
Initializes a new instance of the T:MB2D.EntityComponent.CollisionComponent class with an array of its associated AABB's

Properties

- List< RectangleF > [Boxes](#) [get, set]
Gets or sets the list of bounding boxes used for collision detection.
- bool [Event](#) [get, set]
Gets or sets a value indicating whether this T:MB2D.EntityComponent.CollisionComponent has had a collision event this frame.
- [Entity Collider](#) [get, set]
Gets or sets the collider entity associated with the collision event.

5.4.1 Detailed Description

Used for running collision detection on an [Entity](#)

5.4.2 Constructor & Destructor Documentation

5.4.2.1 CollisionComponent()

```
MB2D.EntityComponent.CollisionComponent.CollisionComponent (
    params RectangleF [] boxes ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.CollisionComponent class with an array of its associated AABB's

Parameters

<i>boxes</i>	The bounding boxes used for detecting collisions.
--------------	---

5.4.3 Property Documentation

5.4.3.1 Boxes

```
List<RectangleF> MB2D.EntityComponent.CollisionComponent.Boxes [get], [set]
```

Gets or sets the list of bounding boxes used for collision detection.

The boxes.

5.4.3.2 Collider

```
Entity MB2D.EntityComponent.CollisionComponent.Collider [get], [set]
```

Gets or sets the collider entity associated with the collision event.

The collider.

5.4.3.3 Event

```
bool MB2D.EntityComponent.CollisionComponent.Event [get], [set]
```

Gets or sets a value indicating whether this T:MB2D.EntityComponent.CollisionComponent has had a collision event this frame.

true if an event occurred; otherwise, false.

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

- MB2D/src/EntityComponent/Components/CollisionComponent.cs

5.5 MB2D.Collision.CollisionMap Class Reference

A 2D Grid that represents a particular space in the game world to check for collisions. Uses spatial indexing to determine where an entity will be located at any given time. For best results, the cellsize and overall size of the map should be tweaked for each individual game screen and environment.

Collaboration diagram for MB2D.Collision.CollisionMap:

MB2D.Collision.CollisionMap
+ Grid + Position + Max
+ CollisionMap() + IndexOf() + IndexExists() + IndexExists() + Insert() + GetCollisions() + UpdatePosition() + Clear()

Public Member Functions

- [CollisionMap](#) (int xMin, int xMax, int yMin, int yMax, int cellSize)
Initializes a new instance of the T:MB2D.Collision.CollisionMap class.
- Point [IndexOf](#) (Point position)
Indexes a world-based coordinate into the collision grid
- bool [IndexExists](#) (int x, int y)
Checks if a particular index exists in the grid
- bool [IndexExists](#) (Point index)
Checks if a particular index exists in the grid
- void [Insert](#) ([Entity](#) entity, [CollisionComponent](#) collision)
Inserts an entity and its associated collision component into the grid
- List< [Entity](#) > [GetCollisions](#) ([Entity](#) entity, [CollisionComponent](#) collision)
Gets a list of all entities located in the same cell/s as a specific single entity
- void [UpdatePosition](#) (int x, int y)
Updates the position of the collision grid.
- void [Clear](#) ()
Clears all non-empty cells of the grid from their entities

Properties

- [Grid](#) [Grid](#) [get]
Gets the geometric representation of the grid
- [Vector2](#) [Position](#) [get]
Gets the current position of the grid.
- [Vector2](#) [Max](#) [get]
Gets the upper bounds of the x and y coordinates in the grid

5.5.1 Detailed Description

A 2D Grid that represents a particular space in the game world to check for collisions. Uses spatial indexing to determine where an entity will be located at any given time. For best results, the cellsize and overall size of the map should be tweaked for each individual game screen and environment.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 CollisionMap()

```
MB2D.Collision.CollisionMap.CollisionMap (
    int xMin,
    int xMax,
    int yMin,
    int yMax,
    int cellSize ) [inline]
```

Initializes a new instance of the T:MB2D.Collision.CollisionMap class.

Parameters

<i>xMin</i>	The grids left most x coordinate.
<i>xMax</i>	Right most x coordinae.
<i>yMin</i>	Top most y coordinate.
<i>yMax</i>	Bottom most y coordinate.
<i>cellSize</i>	The size of each cell in the grid.

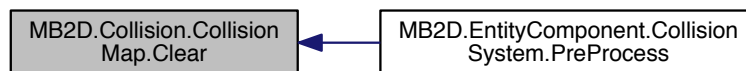
5.5.3 Member Function Documentation

5.5.3.1 Clear()

```
void MB2D.Collision.CollisionMap.Clear ( ) [inline]
```

Clears all non-empty cells of the grid from their entities

Here is the caller graph for this function:



5.5.3.2 GetCollisions()

```
List<Entity> MB2D.Collision.CollisionMap.GetCollisions (
    Entity entity,
    CollisionComponent collision ) [inline]
```

Gets a list of all entities located in the same cell/s as a specific single entity

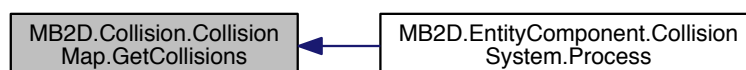
Returns

The entities neighbours.

Parameters

<i>entity</i>	Entity to get collisions for.
<i>collision</i>	Collision component to use in checking.

Here is the caller graph for this function:



5.5.3.3 IndexExists() [1/2]

```
bool MB2D.Collision.CollisionMap.IndexExists (
    int x,
    int y ) [inline]
```

Checks if a particular index exists in the grid

Returns

`true`, if index exists, `false` otherwise.

Parameters

<i>x</i>	The x coordinate.
<i>y</i>	The y coordinate.

5.5.3.4 IndexExists() [2/2]

```
bool MB2D.Collision.CollisionMap.IndexExists (  
    Point index ) [inline]
```

Checks if a particular index exists in the grid

Returns

true, if index exists, false otherwise.

Parameters

<i>index</i>	Index to check.
--------------	-----------------

5.5.3.5 IndexOf()

```
Point MB2D.Collision.CollisionMap.IndexOf (  
    Point position ) [inline]
```

Indexes a world-based coordinate into the collision grid

Returns

The grid-based position.

Parameters

<i>position</i>	World-based position to index.
-----------------	--------------------------------

5.5.3.6 Insert()

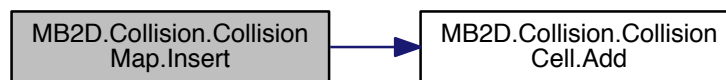
```
void MB2D.Collision.CollisionMap.Insert (  
    Entity entity,  
    CollisionComponent collision ) [inline]
```

Inserts an entity and its associated collision component into the grid

Parameters

<i>entity</i>	Entity to insert.
<i>collision</i>	The entities collision component.

Here is the call graph for this function:



Here is the caller graph for this function:



5.5.3.7 UpdatePosition()

```
void MB2D.Collision.CollisionMap.UpdatePosition (
    int x,
    int y ) [inline]
```

Updates the position of the collision grid.

Parameters

<i>x</i>	The x coordinate.
<i>y</i>	The y coordinate.

5.5.4 Property Documentation

5.5.4.1 Grid

```
Grid MB2D.Collision.CollisionMap.Grid [get]
```

Gets the geometric representation of the grid

The grid.

5.5.4.2 Max

```
Vector2 MB2D.Collision.CollisionMap.Max [get]
```

Gets the upper bounds of the x and y coordinates in the grid

The max coordinates.

5.5.4.3 Position

```
Vector2 MB2D.Collision.CollisionMap.Position [get]
```

Gets the current position of the grid.

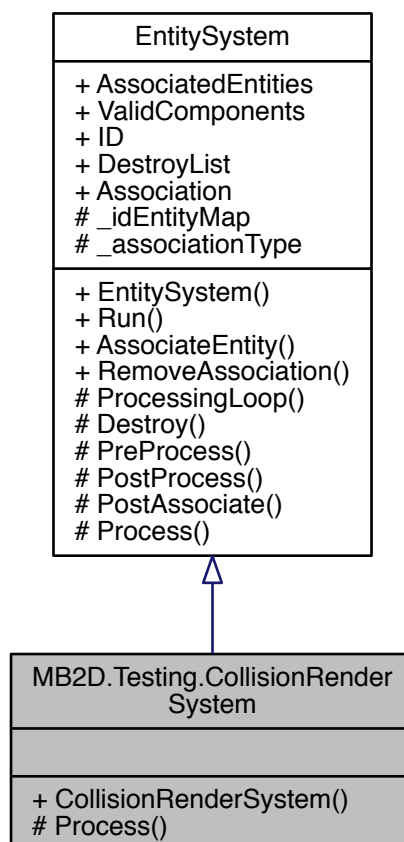
The position.

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

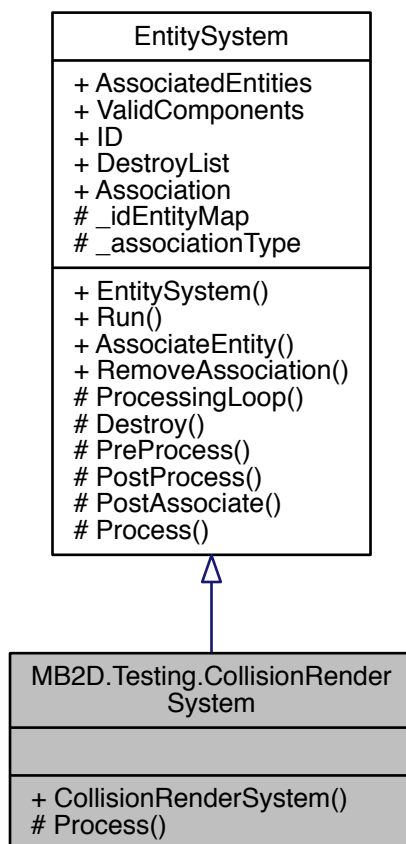
- MB2D/src/Collision/CollisionMap.cs

5.6 MB2D.Testing.CollisionRenderSystem Class Reference

Inheritance diagram for MB2D.Testing.CollisionRenderSystem:



Collaboration diagram for MB2D.Testing.CollisionRenderSystem:



Public Member Functions

- **CollisionRenderSystem** (SpriteBatch spriteBatch)

Protected Member Functions

- override void `Process` (`Entity` entity)
Executes this systems logic on a single entity

Additional Inherited Members

5.6.1 Member Function Documentation

5.6.1.1 `Process()`

```

override void MB2D.Testing.CollisionRenderSystem.Process (
    Entity entity ) [inline], [protected], [virtual]
  
```

Executes this systems logic on a single entity

Parameters

<i>entity</i>	Entity to operate on
---------------	----------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

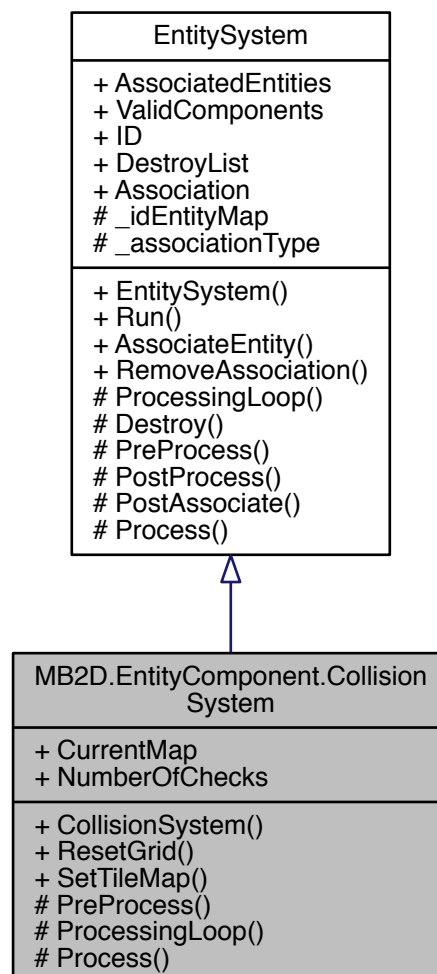
The documentation for this class was generated from the following file:

- MB2D/src/Test/CollisionRenderSystem.cs

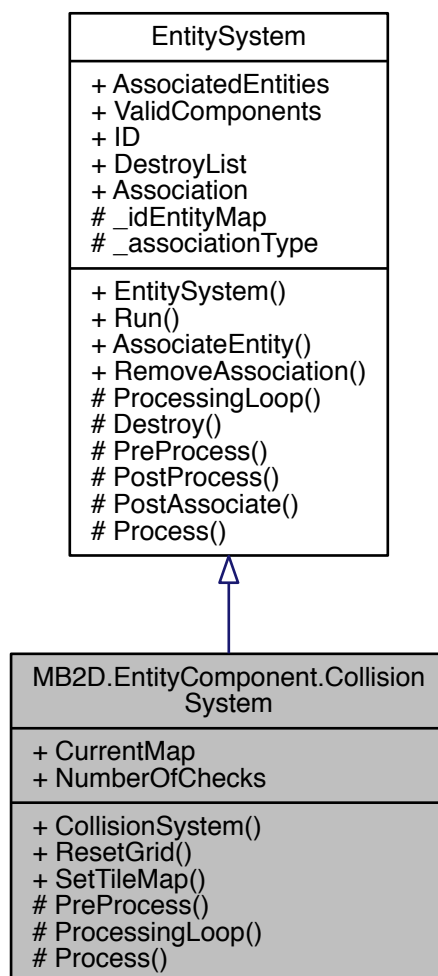
5.7 MB2D.EntityComponent.CollisionSystem Class Reference

Checks collisions. Uses a spatial indexing grid for broad-phase collision checking and AABB checks for narrow phase

Inheritance diagram for MB2D.EntityComponent.CollisionSystem:



Collaboration diagram for MB2D.EntityComponent.CollisionSystem:



Public Member Functions

- `CollisionSystem ()`
Initializes a new instance of the T:MB2D.EntityComponent.CollisionSystem class.
- `void ResetGrid (int xMin, int xMax, int yMin, int yMax, int cellSize)`
Resets the grid position in the world.
- `void SetTileMap (TileMap tileMap)`
Sets the current tile map to check for collisions

Protected Member Functions

- `override void PreProcess ()`
Clears the collision grid and inserts all entities before checking collisions
- `override void ProcessingLoop ()`

Override. Only processes entities with movement components. Still considers static entities, but only as possible neighbours.

- override void [Process](#) ([Entity](#) entity)
Checks all collisions within the entities known collision cells

Properties

- [CollisionMap](#) [CurrentMap](#) [get]
Gets the current collision map.
- int [NumberOfChecks](#) [get]
Gets the number of collision checks made last frame. Used for debugging.

Additional Inherited Members

5.7.1 Detailed Description

Checks collisions. Uses a spatial indexing grid for broad-phase collision checking and AABB checks for narrow phase

5.7.2 Constructor & Destructor Documentation

5.7.2.1 CollisionSystem()

```
MB2D.EntityComponent.CollisionSystem.CollisionSystem ( ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.CollisionSystem class.

5.7.3 Member Function Documentation

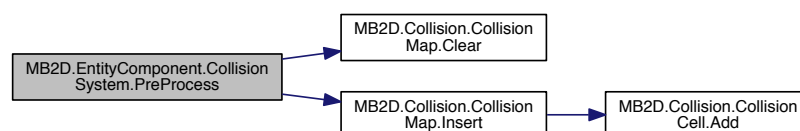
5.7.3.1 PreProcess()

```
override void MB2D.EntityComponent.CollisionSystem.PreProcess ( ) [inline], [protected],  
[virtual]
```

Clears the collision grid and inserts all entities before checking collisions

Reimplemented from [MB2D.EntityComponent.EntitySystem](#).

Here is the call graph for this function:



5.7.3.2 Process()

```
override void MB2D.EntityComponent.CollisionSystem.Process (
    Entity entity ) [inline], [protected], [virtual]
```

Checks all collisions within the entities known collision cells

Parameters

<i>entity</i>	Entity to check.
---------------	----------------------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

Here is the call graph for this function:



5.7.3.3 ProcessingLoop()

```
override void MB2D.EntityComponent.CollisionSystem.ProcessingLoop ( ) [inline], [protected], [virtual]
```

Override. Only processes entities with movement components. Still considers static entities, but only as possible neighbours.

Reimplemented from [MB2D.EntityComponent.EntitySystem](#).

5.7.3.4 ResetGrid()

```
void MB2D.EntityComponent.CollisionSystem.ResetGrid (
    int xMin,
    int xMax,
    int yMin,
    int yMax,
    int cellSize ) [inline]
```

Resets the grid position in the world.

Parameters

<i>xMin</i>	The grids left most x coordinate.
<i>xMax</i>	Right most x coordinae.
<i>yMin</i>	Top most y coordinate.
<i>yMax</i>	Bottom most y coordinate.
<i>cellSize</i>	The size of each cell in the grid.

5.7.3.5 SetTileMap()

```
void MB2D.EntityComponent.CollisionSystem.SetTileMap (
    TileMap tileMap ) [inline]
```

Sets the current tile map to check for collisions

Parameters

<code>tileMap</code>	Tile map.
----------------------	-----------

5.7.4 Property Documentation

5.7.4.1 CurrentMap

```
CollisionMap MB2D.EntityComponent.CollisionSystem.CurrentMap [get]
```

Gets the current collision map.

The current map.

5.7.4.2 NumberOfChecks

```
int MB2D.EntityComponent.CollisionSystem.NumberOfChecks [get]
```

Gets the number of collision checks made last frame. Used for debugging.

The number of collision checks.

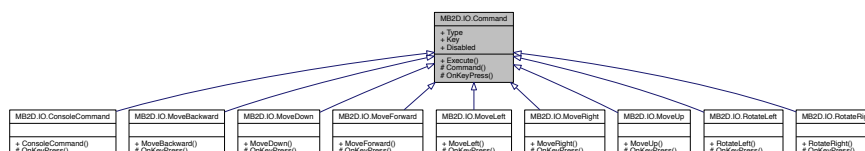
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Systems/CollisionSystem.cs

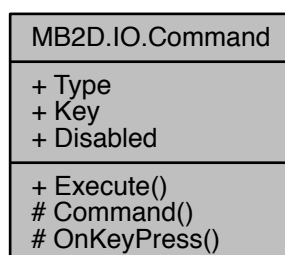
5.8 MB2D.IO.Command Class Reference

Executes an action associated with a specific key

Inheritance diagram for MB2D.IO.Command:



Collaboration diagram for MB2D.IO.Command:



Public Member Functions

- `bool Execute (Entity e=null)`
Executes the specific command on the entity parameter

Protected Member Functions

- `Command (Keys key, CommandType commandType)`
Initializes a new instance of the T:MidnightBlue.Command class.
- `abstract void OnKeyPress (Entity e=null)`
Defines the logic to perform when operating on a given entity

Properties

- `CommandType Type` [get, set]
Gets or sets the trigger type of the command.
- `Keys Key` [get]
Gets the keycode associated with the command.
- `bool Disabled` [get, set]

5.8.1 Detailed Description

Executes an action associated with a specific key

5.8.2 Constructor & Destructor Documentation

5.8.2.1 Command()

```
MB2D.IO.Command.Command (
    Keys key,
    CommandType commandType ) [inline], [protected]
```

Initializes a new instance of the T:MidnightBlue.Command class.

Parameters

<i>key</i>	Key to associate with the command
<i>commandType</i>	Trigger or hold command

5.8.3 Member Function Documentation

5.8.3.1 Execute()

```
bool MB2D.IO.Command.Execute (
    Entity e = null ) [inline]
```

Executes the specific command on the entity parameter

Parameters

<i>e</i>	Entity to operate on. Optional
----------	--------------------------------

5.8.3.2 OnKeyPress()

```
abstract void MB2D.IO.Command.OnKeyPress (
    Entity e = null ) [protected], [pure virtual]
```

Defines the logic to perform when operating on a given entity

Parameters

<i>e</i>	Entity to operate on
----------	----------------------

Implemented in [MB2D.IO.RotateLeft](#), [MB2D.IO.RotateRight](#), [MB2D.IO.MoveBackward](#), [MB2D.IO.MoveForward](#), [MB2D.IO.MoveLeft](#), [MB2D.IO.MoveDown](#), [MB2D.IO.MoveRight](#), [MB2D.IO.MoveUp](#), and [MB2D.IO.ConsoleCommand](#).

5.8.4 Property Documentation**5.8.4.1 Key**

```
Keys MB2D.IO.Command.Key [get]
```

Gets the keycode associated with the command.

The key code

5.8.4.2 Type

```
CommandType MB2D.IO.Command.Type [get], [set]
```

Gets or sets the trigger type of the command.

The command type

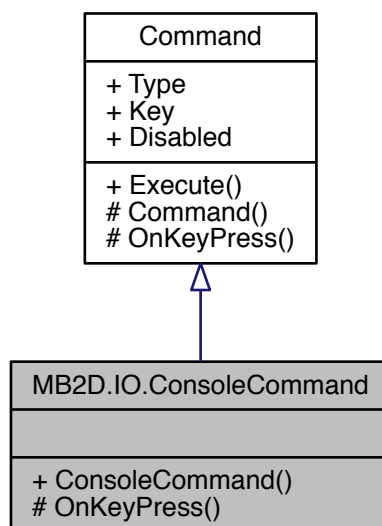
The documentation for this class was generated from the following file:

- MB2D/src/Input/Command.cs

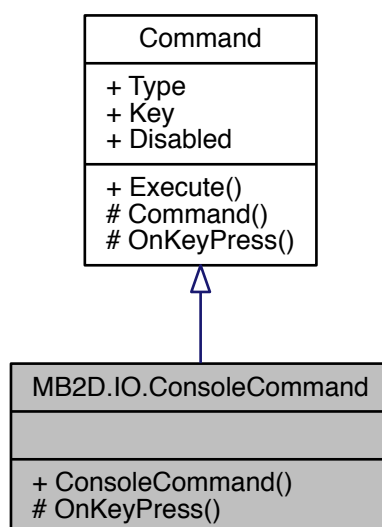
5.9 MB2D.IO.ConsoleCommand Class Reference

Shows or hides the debug console

Inheritance diagram for MB2D.IO.ConsoleCommand:



Collaboration diagram for MB2D.IO.ConsoleCommand:



Public Member Functions

- [ConsoleCommand](#) (Keys key, [CommandType](#) type)
Initializes a new instance of the T:MB2D.IO.ConsoleCommand class.

Protected Member Functions

- override void [OnKeyPress](#) ([Entity](#) e=null)
Toggles the debug console open/closed

Additional Inherited Members

5.9.1 Detailed Description

Shows or hides the debug console

5.9.2 Constructor & Destructor Documentation

5.9.2.1 ConsoleCommand()

```
MB2D.IO.ConsoleCommand.ConsoleCommand (
    Keys key,
    CommandType type ) [inline]
```

Initializes a new instance of the T:MB2D.IO.ConsoleCommand class.

Parameters

<i>key</i>	Key to assign the command to.
<i>type</i>	Type of command trigger.

5.9.3 Member Function Documentation

5.9.3.1 OnKeyPress()

```
override void MB2D.IO.ConsoleCommand.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

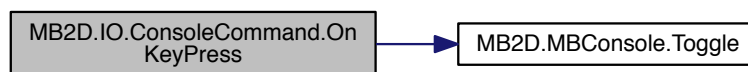
Toggles the debug console open/closed

Parameters

<i>e</i>	Entity with the controller component. Unused.
----------	---

Implements [MB2D.IO.Command](#).

Here is the call graph for this function:



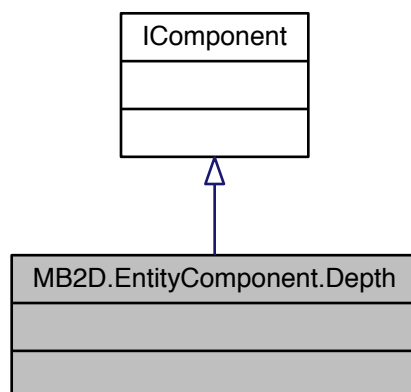
The documentation for this class was generated from the following file:

- MB2D/src/Input/ConsoleCommand.cs

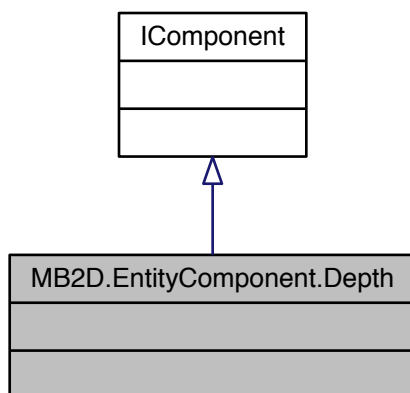
5.10 MB2D.EntityComponent.Depth Class Reference

A tag class used to define an entity that should be draw sorted according to its current z-index

Inheritance diagram for MB2D.EntityComponent.Depth:



Collaboration diagram for MB2D.EntityComponent.Depth:



5.10.1 Detailed Description

A tag class used to define an entity that should be draw sorted according to its current z-index

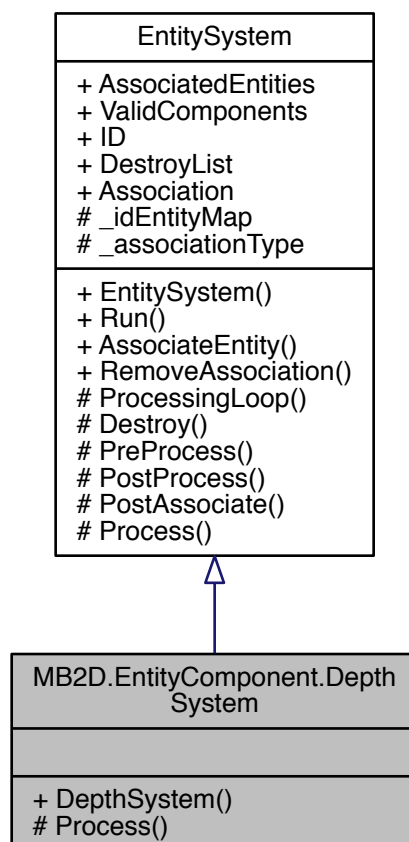
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Components/Depth.cs

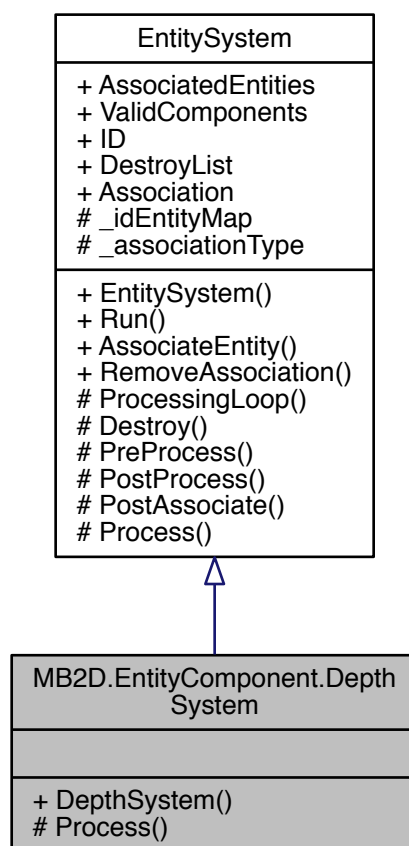
5.11 MB2D.EntityComponent.DepthSystem Class Reference

Changes an entities z-index based on the y coordinate of the top of their sprite

Inheritance diagram for MB2D.EntityComponent.DepthSystem:



Collaboration diagram for MB2D.EntityComponent.DepthSystem:



Public Member Functions

- [DepthSystem](#) ()

Initializes a new instance of the T:MB2D.EntityComponent.DepthSystem class.

Protected Member Functions

- override void [Process](#) ([Entity](#) entity)

Changes the z index based on the y coordinate of the entities bounds top

Additional Inherited Members

5.11.1 Detailed Description

Changes an entities z-index based on the y coordinate of the top of their sprite

5.11.2 Constructor & Destructor Documentation

5.11.2.1 DepthSystem()

```
MB2D.EntityComponent.DepthSystem.DepthSystem ( ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.DepthSystem class.

5.11.3 Member Function Documentation

5.11.3.1 Process()

```
override void MB2D.EntityComponent.DepthSystem.Process (
    Entity entity ) [inline], [protected], [virtual]
```

Changes the z index based on the y coordinate of the entities bounds top

Parameters

<i>entity</i>	Entity to process.
---------------	--------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

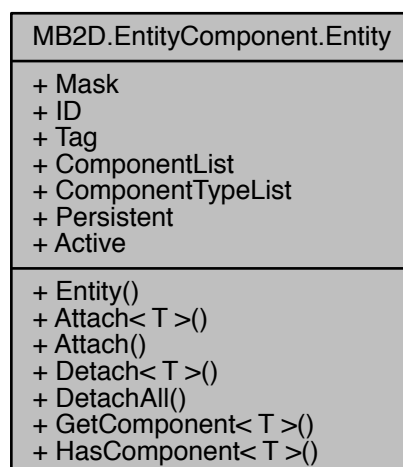
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Systems/DepthSystem.cs

5.12 MB2D.EntityComponent.Entity Class Reference

Represents a tagged and id'd container for components that can be operated on by systems.

Collaboration diagram for MB2D.EntityComponent.Entity:



Public Member Functions

- [Entity](#) ([EntityMap](#) container, string tag="")
Initializes a new instance of the T:MB2D.EntityComponent.Entity class.
- [IComponent Attach< T >](#) (params object[] args)
Attaches a new component to the entity.
- void [Attach](#) ([IComponent](#) component)
Attaches a new component to the entity.
- void [Detach< T >](#) ()
Detaches a specific component from the entity
- void [DetachAll](#) ()
Detachs all of the entities attached components.
- T [GetComponent< T >](#) ()
Queries the entity to see if it has a component attached and returns it if it does
- bool [HasComponent< T >](#) ()
Checks if an entity has a specific component attached

Properties

- ulong [Mask](#) [get, set]
Gets and sets the entities component mask.
- ulong [ID](#) [get, set]
Gets and sets the entities Globally Unique ID in the [EntityMap](#)
- string [Tag](#) [get]
Gets this entities tagname
- Dictionary< Type, [IComponent](#) >.ValueCollection [ComponentList](#) [get]
Gets the list if components attached to this entity.
- Dictionary< Type, [IComponent](#) >.KeyCollection [ComponentTypeList](#) [get]
Gets the types of components this entity has attached
- bool [Persistent](#) [get, set]
Gets or sets a value indicating whether this T:MB2D.EntityComponent.Entity is persistant in its parent T:MB2D.EntityComponent.EntityMap.
- bool [Active](#) [get, set]
Gets or sets a value indicating whether this T:MB2D.EntityComponent.Entity is active. Inactive entities are skipped over in each EntitySystems Process() method but aren't destroyed. Allowing semi-persistant entities.

5.12.1 Detailed Description

Represents a tagged and id'd container for components that can be operated on by systems.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 Entity()

```
MB2D.EntityComponent.Entity.Entity (
    EntityMap container,
    string tag = "" ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.Entity class.

Parameters

<i>container</i>	The entities parent EntityMap
<i>tag</i>	Tagname to give the entity

5.12.3 Member Function Documentation

5.12.3.1 Attach()

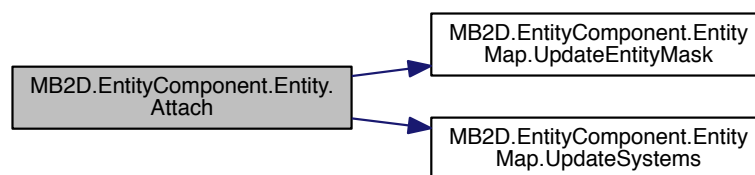
```
void MB2D.EntityComponent.Entity.Attach (
    IComponent component ) [inline]
```

Attaches a new component to the entity.

Parameters

<i>component</i>	Pre constructed component to add
------------------	----------------------------------

Here is the call graph for this function:



5.12.3.2 Attach< T >()

```
IComponent MB2D.EntityComponent.Entity.Attach< T > (
    params object [] args ) [inline]
```

Attaches a new component to the entity.

Parameters

<i>args</i>	The components constructor arguments
-------------	--------------------------------------

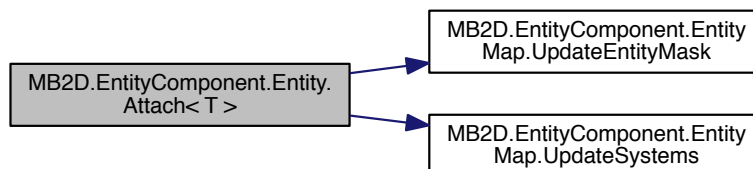
Template Parameters

<i>T</i>	Type of component to attach
----------	-----------------------------

Type Constraints

T : IComponent

Here is the call graph for this function:

**5.12.3.3 Detach< T >()**

```
void MB2D.EntityComponent.Entity.Detach< T > ( ) [inline]
```

Detaches a specific component from the entity

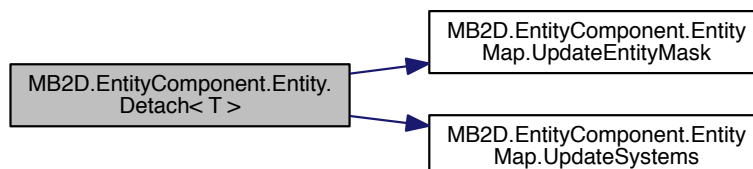
Template Parameters

<i>T</i>	The type of component to detach.
----------	----------------------------------

Type Constraints

T : IComponent

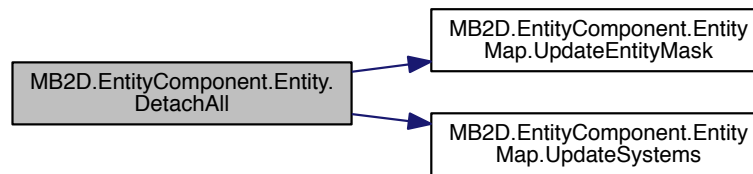
Here is the call graph for this function:

**5.12.3.4 DetachAll()**

```
void MB2D.EntityComponent.Entity.DetachAll ( ) [inline]
```

Detaches all of the entities attached components.

Here is the call graph for this function:



5.12.3.5 GetComponent< T >()

```
T MB2D.EntityComponent.Entity.GetComponent< T > ( ) [inline]
```

Queries the entity to see if it has a component attached and returns it if it does

Returns

The component if the entity has it attached, null otherwise

Template Parameters

<i>T</i>	Component to query the entity for.
----------	------------------------------------

Type Constraints

T : IComponent

5.12.3.6 HasComponent< T >()

```
bool MB2D.EntityComponent.Entity.HasComponent< T > ( ) [inline]
```

Checks if an entity has a specific component attached

Returns

`true`, if component is attached, `false` otherwise.

Template Parameters

<i>T</i>	The type of component to check.
----------	---------------------------------

Type Constraints

T : IComponent

5.12.4 Property Documentation

5.12.4.1 Active

```
bool MB2D.EntityComponent.Entity.Active [get], [set]
```

Gets or sets a value indicating whether this T:MB2D.EntityComponent.Entity is active. Inactive entities are skipped over in each EntitySystems Process() method but aren't destroyed. Allowing semi-persistent entities.

true if the entity is active; otherwise, false.

5.12.4.2 ComponentList

```
Dictionary<Type, IComponent>.ValueCollection MB2D.EntityComponent.Entity.ComponentList [get]
```

Gets the list of components attached to this entity.

The component list.

5.12.4.3 ComponentTypeList

```
Dictionary<Type, IComponent>.KeyCollection MB2D.EntityComponent.Entity.ComponentTypeList [get]
```

Gets the types of components this entity has attached

The component type list.

5.12.4.4 ID

```
ulong MB2D.EntityComponent.Entity.ID [get], [set]
```

Gets and sets the entities Globally Unique ID in the [EntityMap](#)

GUID

5.12.4.5 Mask

```
ulong MB2D.EntityComponent.Entity.Mask [get], [set]
```

Gets and sets the entities component mask.

The component mask.

5.12.4.6 Persistent

```
bool MB2D.EntityComponent.Entity.Persistent [get], [set]
```

Gets or sets a value indicating whether this T:MB2D.EntityComponent.Entity is persistent in its parent T:MB2D.EntityComponent.EntityMap.

true if persistent; otherwise, false.

5.12.4.7 Tag

```
string MB2D.EntityComponent.Entity.Tag [get]
```

Gets this entities tagname

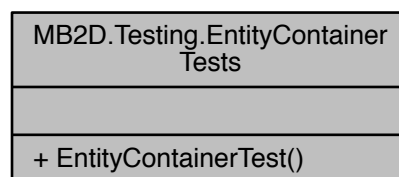
The tagname

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

- MB2D/src/EntityComponent/Entity.cs

5.13 MB2D.Testing.EntityContainerTests Struct Reference

Collaboration diagram for MB2D.Testing.EntityContainerTests:



Static Public Member Functions

- static void **EntityContainerTest** (params string[] args)

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

- MB2D/src/Test/EntityContainerTests.cs

5.14 MB2D.EntityComponent.EntityMap Class Reference

Maps entities, systems and components to one another and provides querying and updating access to all elements

Collaboration diagram for MB2D.EntityComponent.EntityMap:

MB2D.EntityComponent.EntityMap
+ this[string key] + EntityCount + NextID + NextMask
+ EntityMap() + EntityMap() + AddComponent< T >() + AddComponent() + AddSystem< T >() + AddEntity() + UpdateEntityMask() + UpdateSystems() + CreateEntity() + GetComponentID< T >() and 6 more...

Public Member Functions

- [EntityMap](#) ()
Initializes a new instance of the T:MB2D.EntityComponent.EntityMap class.
- [EntityMap](#) ([EntityMap](#) map)
Initializes a new instance of the T:MB2D.EntityComponent.EntityMap class. Uses an existing [EntityMap](#) to copy all registered systems and components as well as any persistant Entities.
- void [AddComponent](#)< T > ()
Registers a new component type to the [EntityMap](#)
- void [AddComponent](#) (Type componentType)
Registers a new component type to the [EntityMap](#)
- void [AddSystem](#)< T > (params object[] args)
Registers a new [EntitySystem](#) to the map
- void [AddEntity](#) ([Entity](#) entity)
Adds a created entity to this map
- void [UpdateEntityMask](#) ([Entity](#) entity)
Updates a specific entities component mask. Use after registering new components or systems.
- void [UpdateSystems](#) ([Entity](#) entity)
Updates each systems associated entity list, adding the specified [Entity](#). Use after creating a new [Entity](#) and adding it manually
- [Entity](#) [CreateEntity](#) (string tag="")
Creates a new [Entity](#) with the given tag in this map. Auto-Registers the entity with all systems and updates its mask.
- ulong [GetComponentID](#)< T > ()
Gets the id of a specified component type if it exists.
- [EntitySystem](#) [GetSystem](#)< T > ()
Retrieves a pre-registered system from the map
- List< [Entity](#) > [EntitiesWithComponent](#)< T > ()
- void [Clear](#) ()

Clears all entities from this map except for any marked as persistent.

- void **Reset** ()
- void **MakeBlueprint** (string id, Action< [Entity](#) > buildFunction)
- void **UseBlueprint** (string name, [Entity](#) entity)

Properties

- [Entity](#) **this[string key]** [get]
Gets the T:MB2D.EntityComponent.Entity with the specified tag if it exists; null otherwise
- int **EntityCount** [get]
Gets the number of entities in the map.
- ulong **NextID** [get]
Auto-increments the last generated GUID and retrieves the result

5.14.1 Detailed Description

Maps entities, systems and components to one another and provides querying and updating access to all elements

5.14.2 Constructor & Destructor Documentation

5.14.2.1 EntityMap() [1/2]

```
MB2D.EntityComponent.EntityMap.EntityMap ( ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.EntityMap class.

5.14.2.2 EntityMap() [2/2]

```
MB2D.EntityComponent.EntityMap.EntityMap (
    EntityMap map ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.EntityMap class. Uses an existing [EntityMap](#) to copy all registered systems and components as well as any persistent Entities.

Parameters

<i>map</i>	EntityMap to copy from
------------	--

5.14.3 Member Function Documentation

5.14.3.1 AddComponent()

```
void MB2D.EntityComponent.EntityMap.AddComponent (
    Type componentType ) [inline]
```

Registers a new component type to the [EntityMap](#)

Parameters

<i>componentType</i>	Type of component to register
----------------------	-------------------------------

5.14.3.2 AddComponent< T >()

```
void MB2D.EntityComponent.EntityMap.AddComponent< T > ( ) [inline]
```

Registers a new component type to the [EntityMap](#)

Template Parameters

<i>T</i>	Type of component to register
----------	-------------------------------

Type Constraints

T : IComponent

5.14.3.3 AddEntity()

```
void MB2D.EntityComponent.EntityMap.AddEntity (
    Entity entity ) [inline]
```

Adds a created entity to this map

Parameters

<i>entity</i>	Entity to add
---------------	-------------------------------

5.14.3.4 AddSystem< T >()

```
void MB2D.EntityComponent.EntityMap.AddSystem< T > (
    params object [] args ) [inline]
```

Registers a new [EntitySystem](#) to the map

Template Parameters

<i>T</i>	EntitySystem type to add
----------	--

Type Constraints

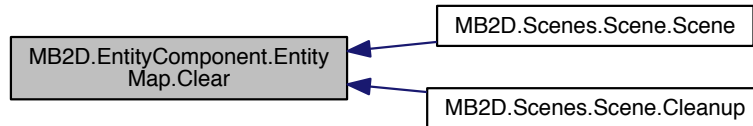
T : EntitySystem

5.14.3.5 Clear()

```
void MB2D.EntityComponent.EntityMap.Clear ( ) [inline]
```

Clears all entities from this map except for any marked as persistent.

Here is the caller graph for this function:



5.14.3.6 CreateEntity()

```
Entity MB2D.EntityComponent.EntityMap.CreateEntity (
    string tag = "" ) [inline]
```

Creates a new [Entity](#) with the given tag in this map. Auto-Registers the entity with all systems and updates its mask.

Returns

The created entity

Parameters

<i>tag</i>	Tagname to give the entity
------------	----------------------------

5.14.3.7 GetComponentID< T >()

```
ulong MB2D.EntityComponent.EntityMap.GetComponentID< T > ( ) [inline]
```

Gets the id of a specified component type if it exists.

Returns

The component id mask.

Template Parameters

<i>T</i>	Type of component to query for.
----------	---------------------------------

Type Constraints

T : *IComponent*

5.14.3.8 `GetSystem< T >()`

```
EntitySystem MB2D.EntityComponent.EntityMap.GetSystem< T > ( ) [inline]
```

Retrieves a pre-registered system from the map

Returns

The system if it exists; null otherwise

Template Parameters

<i>T</i>	Type of system to retrieve.
----------	-----------------------------

Type Constraints

T* : *EntitySystem

5.14.3.9 `UpdateEntityMask()`

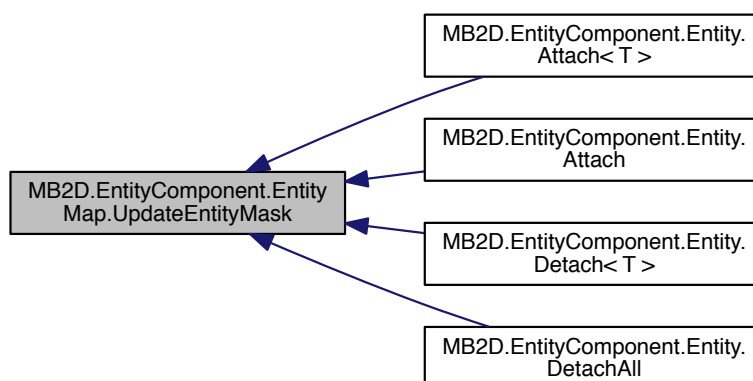
```
void MB2D.EntityComponent.EntityMap.UpdateEntityMask (
    Entity entity ) [inline]
```

Updates a specific entities component mask. Use after registering new components or systems.

Parameters

<i>entity</i>	Entity to update
---------------	------------------

Here is the caller graph for this function:



5.14.3.10 UpdateSystems()

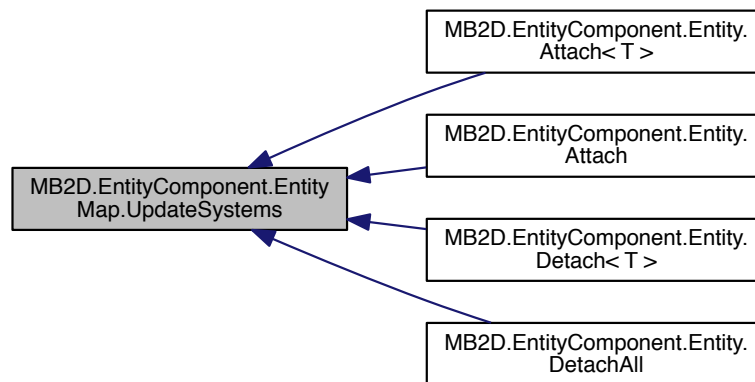
```
void MB2D.EntityComponent.EntityMap.UpdateSystems (
    Entity entity ) [inline]
```

Updates each systems associated entity list, adding the specified [Entity](#). Use after creating a new [Entity](#) and adding it manually

Parameters

<i>entity</i>	Entity to track in each system
---------------	--

Here is the caller graph for this function:



5.14.4 Property Documentation

5.14.4.1 EntityCount

```
int MB2D.EntityComponent.EntityMap.EntityCount [get]
```

Gets the number of entities in the map.

The entity count.

5.14.4.2 NextID

```
ulong MB2D.EntityComponent.EntityMap.NextID [get]
```

Auto-increments the last generated GUID and retrieves the result

The next identifier.

5.14.4.3 this[string key]

```
Entity MB2D.EntityComponent.EntityMap.this[string key] [get]
```

Gets the `T:MB2D.EntityComponent.Entity` with the specified tag if it exists; null otherwise

Parameters

key	Tagname of the entity to retireve.
-----	------------------------------------

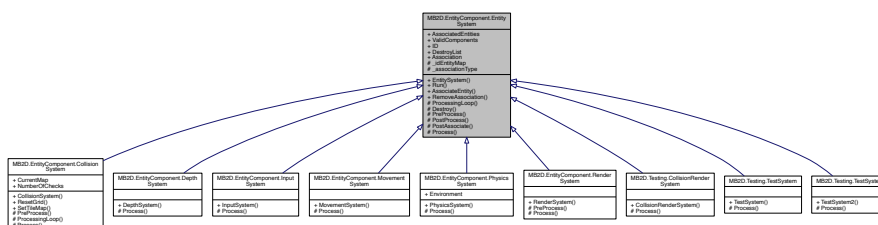
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/EntityMap.cs

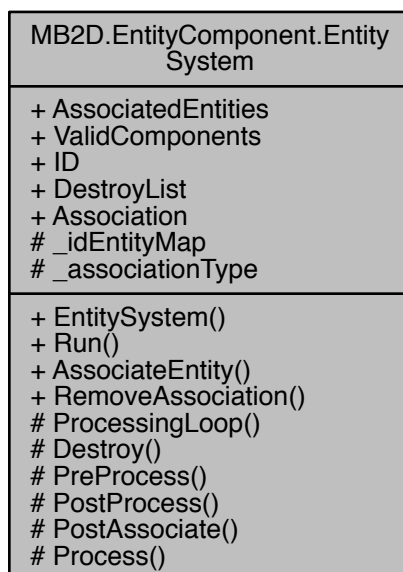
5.15 MB2D.EntityComponent.EntitySystem Class Reference

Performs logic on an entity.

Inheritance diagram for MB2D.EntityComponent.EntitySystem:



Collaboration diagram for MB2D.EntityComponent.EntitySystem:



Public Member Functions

- [EntitySystem](#) (params [Type](#)[] components)
Initializes a new instance of the `T:MB2D.EntityComponent.EntitySystem` class. Checks if the passed in types are valid components and if not, all components are deregistered leaving a system that knows about nothing and can't operate on any entities.
- void [Run](#) ()
Runs this systems [Process\(\)](#) method on all entities
- void [AssociateEntity](#) ([Entity](#) entity)
Associates a new entity with this system.
- void [RemoveAssociation](#) ([Entity](#) entity)
Decouples an association with this system

Protected Member Functions

- virtual void [ProcessingLoop](#) ()
Executes [Process\(\)](#) on all AssociatedEntities.
- void [Destroy](#) ([Entity](#) entity)
Adds the specific entity to the destroy list to be cleaned up the next time a system is run
- virtual void [PreProcess](#) ()
Used to setup any data needed before processing all entities, such as sorting a list ahead of time
- virtual void [PostProcess](#) ()
Used to cleanup or execute any teardown logic needed
- virtual void [PostAssociate](#) ([Entity](#) entity)
Used to define any logic or extra data needed after an entity is associated with the system.
- abstract void [Process](#) ([Entity](#) entity)
Executes this systems logic on a single entity

Protected Attributes

- Dictionary< ulong, [Entity](#) > [_idEntityMap](#)
All GUID's of entities this system knows about
- EntityAssociation [_associationType](#)
Defines if this system should be Loose or Strict

Properties

- List< [Entity](#) > [AssociatedEntities](#) [get, set]
Gets or sets the associated entities list.
- List< [Type](#) > [ValidComponents](#) [get]
Gets the list of components this system is interested in
- ulong [ID](#) [get, set]
Gets or sets this systems ID mask
- List< [Entity](#) > [DestroyList](#) [get]
Gets the destroy list.
- EntityAssociation [Association](#) [get]
Gets the systems association level.

5.15.1 Detailed Description

Performs logic on an entity.

5.15.2 Constructor & Destructor Documentation

5.15.2.1 EntitySystem()

```
MB2D.EntityComponent.EntitySystem.EntitySystem (
    params Type [] components ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.EntitySystem class. Checks if the passed in types are valid components and if not, all components are deregistered leaving a system that knows about nothing and can't operate on any entities.

Parameters

<i>components</i>	Components this system is interested in
-------------------	---

5.15.3 Member Function Documentation

5.15.3.1 AssociateEntity()

```
void MB2D.EntityComponent.EntitySystem.AssociateEntity (
    Entity entity ) [inline]
```

Associates a new entity with this system.

Parameters

<i>entity</i>	Entity to associate
---------------	---------------------

5.15.3.2 Destroy()

```
void MB2D.EntityComponent.EntitySystem.Destroy (
    Entity entity ) [inline], [protected]
```

Adds the specific entity to the destroy list to be cleaned up the next time a system is run

Parameters

<i>entity</i>	Entity to destroy.
---------------	--------------------

5.15.3.3 PostAssociate()

```
virtual void MB2D.EntityComponent.EntitySystem.PostAssociate (
    Entity entity ) [inline], [protected], [virtual]
```


Used to define any logic or extra data needed after an entity is associated with the system.

Parameters

<i>entity</i>	Entity .
---------------	--------------------------

5.15.3.4 PostProcess()

```
virtual void MB2D.EntityComponent.EntitySystem.PostProcess ( ) [inline], [protected], [virtual]
```

Used to cleanup or execute any teardown logic needed

5.15.3.5 PreProcess()

```
virtual void MB2D.EntityComponent.EntitySystem.PreProcess ( ) [inline], [protected], [virtual]
```

Used to setup any data needed before processing all entities, such as sorting a list ahead of time

Reimplemented in [MB2D.EntityComponent.CollisionSystem](#), and [MB2D.EntityComponent.RenderSystem](#).

5.15.3.6 Process()

```
abstract void MB2D.EntityComponent.EntitySystem.Process (
    Entity entity ) [protected], [pure virtual]
```

Executes this systems logic on a single entity

Parameters

<i>entity</i>	Entity to operate on
---------------	--------------------------------------

Implemented in [MB2D.EntityComponent.CollisionSystem](#), [MB2D.EntityComponent.RenderSystem](#), [MB2D.D.Testing.TestSystem2](#), [MB2D.EntityComponent.PhysicsSystem](#), [MB2D.Testing.TestSystem](#), [MB2D.EntityComponent.InputSystem](#), [MB2D.EntityComponent.MovementSystem](#), [MB2D.EntityComponent.DepthSystem](#), and [MB2D.Testing.CollisionRenderSystem](#).

5.15.3.7 ProcessingLoop()

```
virtual void MB2D.EntityComponent.EntitySystem.ProcessingLoop ( ) [inline], [protected], [virtual]
```

Executes [Process\(\)](#) on all AssociatedEntities.

Reimplemented in [MB2D.EntityComponent.CollisionSystem](#).

5.15.3.8 RemoveAssociation()

```
void MB2D.EntityComponent.EntitySystem.RemoveAssociation (
    Entity entity ) [inline]
```

Decouples an association with this system

Parameters

<i>entity</i>	Entity to decouple.
---------------	-------------------------------------

5.15.3.9 Run()

```
void MB2D.EntityComponent.EntitySystem.Run ( ) [inline]
```

Runs this systems [Process\(\)](#) method on all entities

5.15.4 Member Data Documentation**5.15.4.1 _associationType**

```
EntityAssociation MB2D.EntityComponent.EntitySystem._associationType [protected]
```

Defines if this system should be Loose or Strict

5.15.4.2 _idEntityMap

```
Dictionary<ulong, Entity> MB2D.EntityComponent.EntitySystem._idEntityMap [protected]
```

All GUID's of entities this system knows about

5.15.5 Property Documentation**5.15.5.1 AssociatedEntities**

```
List<Entity> MB2D.EntityComponent.EntitySystem.AssociatedEntities [get], [set]
```

Gets or sets the associated entities list.

The associated entities.

5.15.5.2 Association

```
EntityAssociation MB2D.EntityComponent.EntitySystem.Association [get]
```

Gets the systems association level.

The association level.

5.15.5.3 DestroyList

```
List<Entity> MB2D.EntityComponent.EntitySystem.DestroyList [get]
```

Gets the destroy list.

The destroy list.

5.15.5.4 ID

```
ulong MB2D.EntityComponent.EntitySystem.ID [get], [set]
```

Gets or sets this systems ID mask

The identifier mask.

5.15.5.5 ValidComponents

```
List<Type> MB2D.EntityComponent.EntitySystem.ValidComponents [get]
```

Gets the list of components this system is interested in

The valid components.

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

- MB2D/src/EntityComponent/EntitySystem.cs

5.16 MB2D.Geometry.Grid Class Reference

Represents a grid structure. Can be drawn via a SpriteBatch

Collaboration diagram for MB2D.Geometry.Grid:

MB2D.Geometry.Grid
+ RowCount + ColCount + RowSize + ColSize + CellSize
+ Grid()

Public Member Functions

- [Grid](#) (int rows, int cols, int rowSize, int colSize)
Initializes a new instance of the T:MB2D.Geometry.Grid class.

Properties

- int [RowCount](#) [get, set]
Gets or sets the number of rows.
- int [ColCount](#) [get, set]
Gets or sets the number of columns.
- int [RowSize](#) [get, set]
Gets or sets the size of each row.
- int [ColSize](#) [get, set]
Gets or sets the size of each column.
- Vector2 [CellSize](#) [get]
Gets the size of each cell.

5.16.1 Detailed Description

Represents a grid structure. Can be drawn via a SpriteBatch

5.16.2 Constructor & Destructor Documentation

5.16.2.1 Grid()

```
MB2D.Geometry.Grid.Grid (
    int rows,
    int cols,
    int rowSize,
    int colSize ) [inline]
```

Initializes a new instance of the T:MB2D.Geometry.Grid class.

Parameters

<i>rows</i>	Number of rows
<i>cols</i>	Number of columns
<i>rowSize</i>	Size of each row
<i>colSize</i>	Size of each column

5.16.3 Property Documentation

5.16.3.1 CellSize

```
Vector2 MB2D.Geometry.Grid.CellSize [get]
```

Gets the size of each cell.

The size of each cell.

5.16.3.2 ColCount

```
int MB2D.Geometry.Grid.ColCount [get], [set]
```

Gets or sets the number of columns.

The column count.

5.16.3.3 ColSize

```
int MB2D.Geometry.Grid.ColSize [get], [set]
```

Gets or sets the size of each column.

The size of each column.

5.16.3.4 RowCount

```
int MB2D.Geometry.Grid.RowCount [get], [set]
```

Gets or sets the number of rows.

The row count.

5.16.3.5 RowSize

```
int MB2D.Geometry.Grid.RowSize [get], [set]
```

Gets or sets the size of each row.

The size of each row.

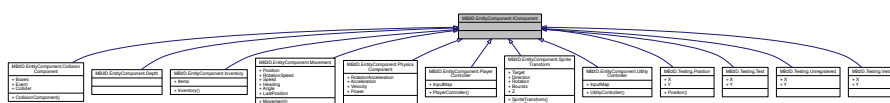
The documentation for this class was generated from the following file:

- MB2D/src/Geometry/Grid.cs

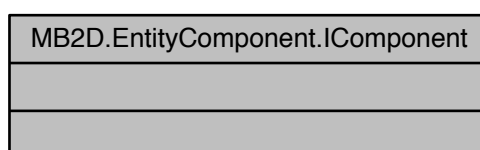
5.17 MB2D.EntityComponent.IComponent Interface Reference

Tags any class as a valid component for use in the [EntityMap](#). Derived classes should contain no logic, only data fields.

Inheritance diagram for MB2D.EntityComponent.IComponent:



Collaboration diagram for MB2D.EntityComponent.IComponent:



5.17.1 Detailed Description

Tags any class as a valid component for use in the [EntityMap](#). Derived classes should contain no logic, only data fields.

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

- MB2D/src/EntityComponent/IComponent.cs

5.18 MB2D.IO.InputMap Class Reference

Maps commands to keys and trigger types

Collaboration diagram for MB2D.IO.InputMap:

MB2D.IO.InputMap
+ this[Keys key] + Collection
+ InputMap() + Assign< T >() + Search< T >()

Public Member Functions

- [InputMap](#) ()
Initializes a new instance of the T:MB2D.IO.InputMap class.
- [Command Assign< T >](#) (Keys key, [CommandType](#) type, params object[] args)
Assign a [Command](#) to the specified key and CommandType .
- [T Search< T >](#) ()
Searches the map for a specific command

Properties

- List< [Command](#) > [this\[Keys key\]](#) [get]
Gets the T:MB2D.IO.Commnad mapped to the specified key.
- Dictionary< Keys, List< [Command](#) > > [Collection](#) [get]
Gets a key->[Command](#) dictionary

5.18.1 Detailed Description

Maps commands to keys and trigger types

5.18.2 Constructor & Destructor Documentation

5.18.2.1 InputMap()

```
MB2D.IO.InputMap.InputMap ( ) [inline]
```

Initializes a new instance of the T:MB2D.IO.InputMap class.

5.18.3 Member Function Documentation

5.18.3.1 Assign< T >()

```
Command MB2D.IO.InputMap.Assign< T > (
    Keys key,
    CommandType type,
    params object [] args ) [inline]
```

Assign a [Command](#) to the specified key and CommandType .

Parameters

<i>key</i>	Key to assign the command to
<i>type</i>	Type of trigger

Template Parameters

<i>T</i>	The command to assign
----------	-----------------------

Type Constraints

T : Command

5.18.3.2 Search< T >()

```
T MB2D.IO.InputMap.Search< T > ( ) [inline]
```

Searches the map for a specific command

Template Parameters

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

Type Constraints

T : Command

5.18.4 Property Documentation

5.18.4.1 Collection

Dictionary<Keys, List<Command> > MB2D.IO.InputMap.Collection [get]

Gets a key->Command dictionary

The collection of commands.

5.18.4.2 this[Keys key]

List<Command> MB2D.IO.InputMap.this[Keys key] [get]

Gets the T:MB2D.IO.Commnad mapped to the specified key.

Parameters

<i>key</i>	Key to query
------------	--------------

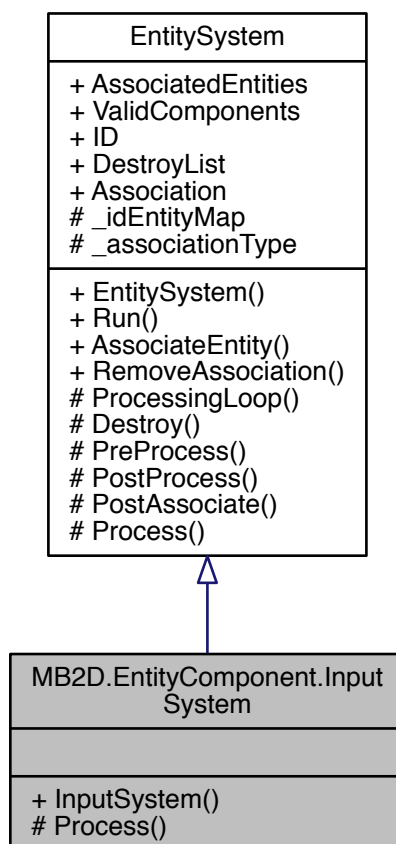
The documentation for this class was generated from the following file:

- MB2D/src/Input/InputMap.cs

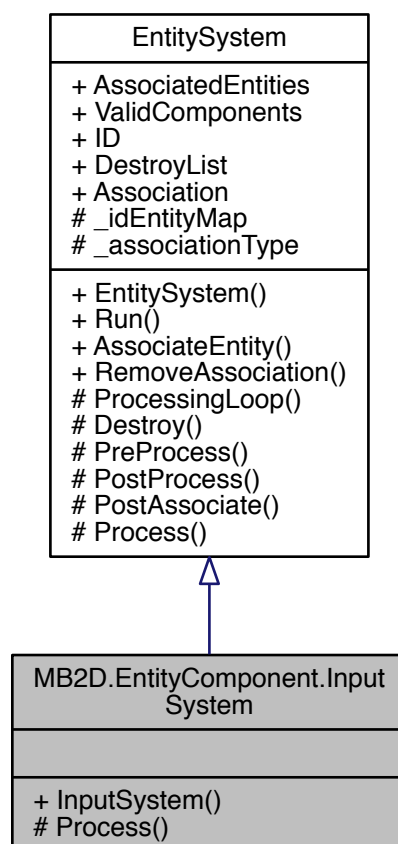
5.19 MB2D.EntityComponent.InputSystem Class Reference

Processes input for [PlayerController](#) and [UtilityController](#) entities. Can operate on an entity with either or both components

Inheritance diagram for MB2D.EntityComponent.InputSystem:



Collaboration diagram for MB2D.EntityComponent.InputSystem:



Protected Member Functions

- override void `Process` (`Entity` entity)
Processes the controllers inputs

Additional Inherited Members

5.19.1 Detailed Description

Processes input for `PlayerController` and `UtilityController` entities. Can operate on an entity with either or both components

5.19.2 Member Function Documentation

5.19.2.1 `Process()`

```

override void MB2D.EntityComponent.InputSystem.Process (
    Entity entity ) [inline], [protected], [virtual]
  
```

Processes the controllers inputs

Parameters

<i>entity</i>	Entity to process.
---------------	------------------------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

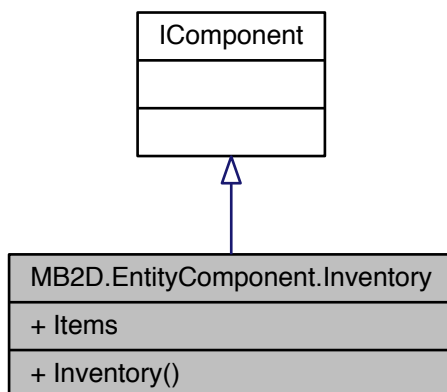
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Systems/InputSystem.cs

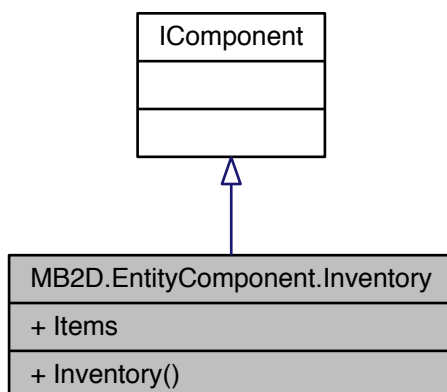
5.20 MB2D.EntityComponent.Inventory Class Reference

Defines a dictionary of [Collectable](#) types used for entities

Inheritance diagram for MB2D.EntityComponent.Inventory:



Collaboration diagram for MB2D.EntityComponent.Inventory:



Properties

- Dictionary< Type, [Collectable](#) > [Items](#) [get, set]

The items currently in the inventory

5.20.1 Detailed Description

Defines a dictionary of [Collectable](#) types used for entities

5.20.2 Property Documentation

5.20.2.1 Items

Dictionary<Type, [Collectable](#)> MB2D.EntityComponent.Inventory.Items [get], [set]

The items currently in the inventory

The items.

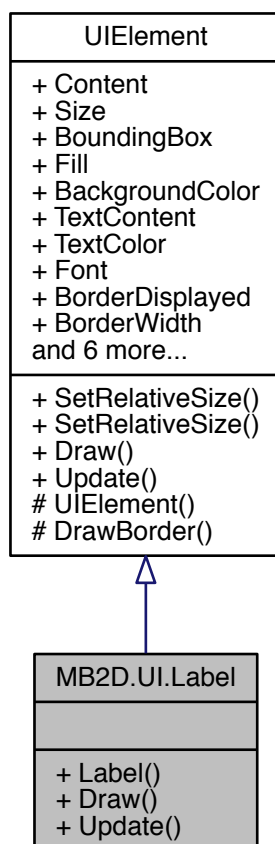
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Components/Inventory.cs

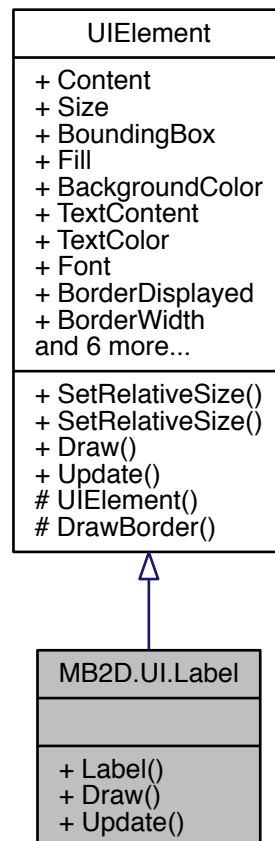
5.21 MB2D.UI.Label Class Reference

A static [UIElement](#) with a TextContent, border and optional texture

Inheritance diagram for MB2D.UI.Label:



Collaboration diagram for MB2D.UI.Label:



Public Member Functions

- [Label](#) ()
Initializes a new instance of the T:MB2D.UI.Label class.
- override void [Draw](#) (SpriteBatch spriteBatch)
Draw the label to the window.
- override void [Update](#) ()
Updates the labels state

Additional Inherited Members

5.21.1 Detailed Description

A static [UIElement](#) with a TextContent, border and optional texture

5.21.2 Constructor & Destructor Documentation

5.21.2.1 Label()

```
MB2D.UI.Label.Label ( ) [inline]
```

Initializes a new instance of the T:MB2D.UI.Label class.

5.21.3 Member Function Documentation

5.21.3.1 Draw()

```
override void MB2D.UI.Label.Draw (
    SpriteBatch spriteBatch ) [inline], [virtual]
```

Draw the label to the window.

Parameters

<i>spriteBatch</i>	Sprite batch to draw to.
--------------------	--------------------------

Reimplemented from [MB2D.UI.UIElement](#).

5.21.3.2 Update()

```
override void MB2D.UI.Label.Update ( ) [inline], [virtual]
```

Updates the labels state

Implements [MB2D.UI.UIElement](#).

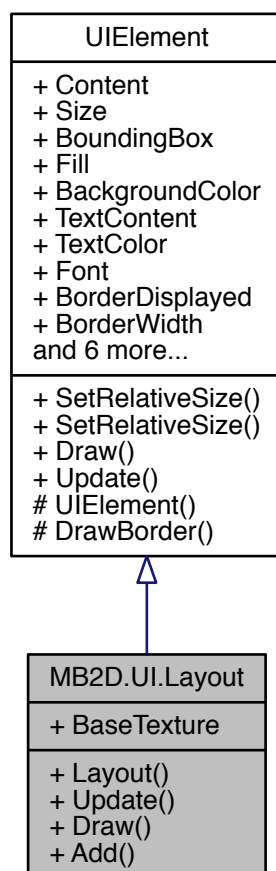
The documentation for this class was generated from the following file:

- MB2D/src/UI/Label.cs

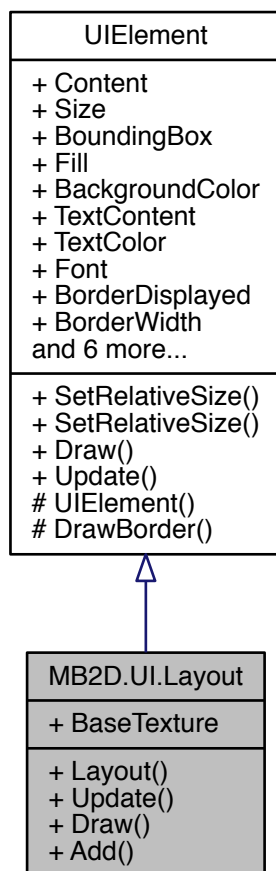
5.22 MB2D.UI.Layout Class Reference

A container for UIElements used within a [UIView](#). Used to divide the View into smaller segments and to move around a group of elements easily

Inheritance diagram for MB2D.UI.Layout:



Collaboration diagram for MB2D.UI.Layout:



Public Member Functions

- **Layout** (**UIView** parentView, int rows, int cols)
Initializes a new instance of the T:MB2D.UI.Layout class. Divides the layouts rows and columns into an even cell size based of its parents size.
- override void **Update** ()
Update all elements contained within the layout.
- override void **Draw** (SpriteBatch spriteBatch)
Draws all elements contained within the layout and the layouts border to the window.
- void **Add** (**UIElement** element, int atRow, int atCol, int rowSpan, int colSpan)
*Adds a new **UIElement** to the layout, setting its size relative to this layouts cell size*

Properties

- Texture2D **BaseTexture** [get, set]

Additional Inherited Members

5.22.1 Detailed Description

A container for UIElements used within a [UIView](#). Used to divide the View into smaller segments and to move around a group of elements easily

5.22.2 Constructor & Destructor Documentation

5.22.2.1 Layout()

```
MB2D.UI.Layout.Layout (
    UIView parentView,
    int rows,
    int cols ) [inline]
```

Initializes a new instance of the T:MB2D.UI.Layout class. Divides the layouts rows and columns into an even cell size based of its parents size.

Parameters

<i>rows</i>	Number of rows the layout has.
<i>cols</i>	Number of columns the layout has.

5.22.3 Member Function Documentation

5.22.3.1 Add()

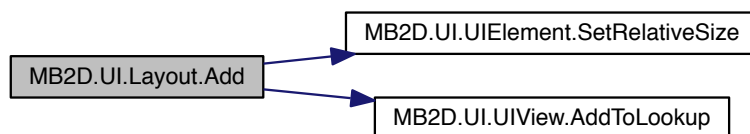
```
void MB2D.UI.Layout.Add (
    UIElement element,
    int atRow,
    int atCol,
    int rowSpan,
    int colSpan ) [inline]
```

Adds a new [UIElement](#) to the layout, setting its size relative to this layouts cell size

Parameters

<i>element</i>	Element to add.
<i>atRow</i>	Row position to add the element at.
<i>atCol</i>	Column position to add the element at.
<i>rowSpan</i>	Number of rows the element should span.
<i>colSpan</i>	Number of columns the element should span.

Here is the call graph for this function:



5.22.3.2 Draw()

```
override void MB2D.UI.Layout.Draw (
    SpriteBatch spriteBatch ) [inline], [virtual]
```

Draws all elements contained within the layout and the layouts border to the window.

Parameters

<i>spriteBatch</i>	Sprite batch to draw to.
--------------------	--------------------------

Reimplemented from [MB2D.UI.UIElement](#).

5.22.3.3 Update()

```
override void MB2D.UI.Layout.Update ( ) [inline], [virtual]
```

Update all elements contained within the layout.

Implements [MB2D.UI.UIElement](#).

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

- MB2D/src/UI/Layout.cs

5.23 MB2D.Geometry.Line Class Reference

A line structure, can be drawn via SpriteBatch

Collaboration diagram for MB2D.Geometry.Line:

MB2D.Geometry.Line
+ Start + End
+ Line() + Line() + PointDistance() + PointDistance()

Public Member Functions

- [Line](#) (Vector2 start, Vector2 end)
Initializes a new instance of the T:MB2D.Geometry.Line class.
- [Line](#) (Point start, Point end)
Initializes a new instance of the T:MB2D.Geometry.Line class.
- float [PointDistance](#) (Vector2 point)
Gets the distance from this line a given point is
- float [PointDistance](#) (Point point)
Gets the distance from this line a given point is

Properties

- Vector2 [Start](#) [get, set]
Gets or sets the start point.
- Vector2 [End](#) [get, set]
Gets or sets the end point.

5.23.1 Detailed Description

A line structure, can be drawn via SpriteBatch

5.23.2 Constructor & Destructor Documentation

5.23.2.1 [Line\(\)](#) [1/2]

```
MB2D.Geometry.Line.Line (
    Vector2 start,
    Vector2 end ) [inline]
```

Initializes a new instance of the T:MB2D.Geometry.Line class.

Parameters

<i>start</i>	Start point
<i>end</i>	End point

5.23.2.2 Line() [2/2]

```
MB2D.Geometry.Line.Line (
    Point start,
    Point end ) [inline]
```

Initializes a new instance of the T:MB2D.Geometry.Line class.

Parameters

<i>start</i>	Start point
<i>end</i>	End point

5.23.3 Member Function Documentation**5.23.3.1 PointDistance()** [1/2]

```
float MB2D.Geometry.Line.PointDistance (
    Vector2 point ) [inline]
```

Gets the distance from this line a given point is

Returns

The distance.

Parameters

<i>point</i>	Point to calculate.
--------------	---------------------

5.23.3.2 PointDistance() [2/2]

```
float MB2D.Geometry.Line.PointDistance (
    Point point ) [inline]
```

Gets the distance from this line a given point is

Returns

The distance.

Parameters

<i>point</i>	Point to calculate.
--------------	---------------------

5.23.4 Property Documentation

5.23.4.1 End

`Vector2 MB2D.Geometry.Line.End [get], [set]`

Gets or sets the end point.

The end point.

5.23.4.2 Start

`Vector2 MB2D.Geometry.Line.Start [get], [set]`

Gets or sets the start point.

The start point

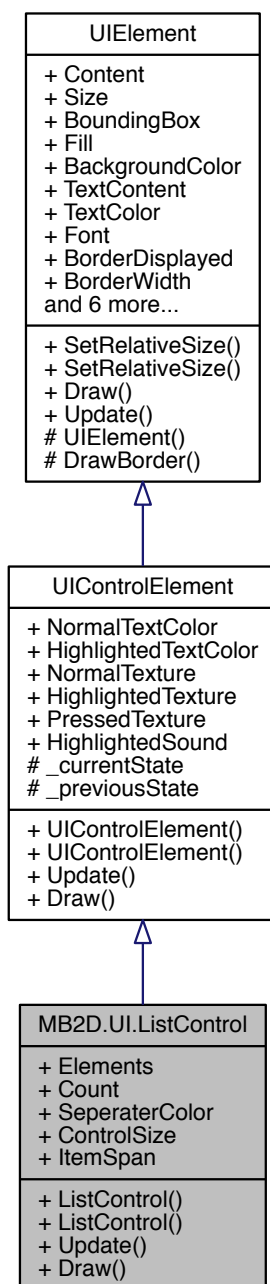
The documentation for this class was generated from the following file:

- MB2D/src/Geometry/Line.cs

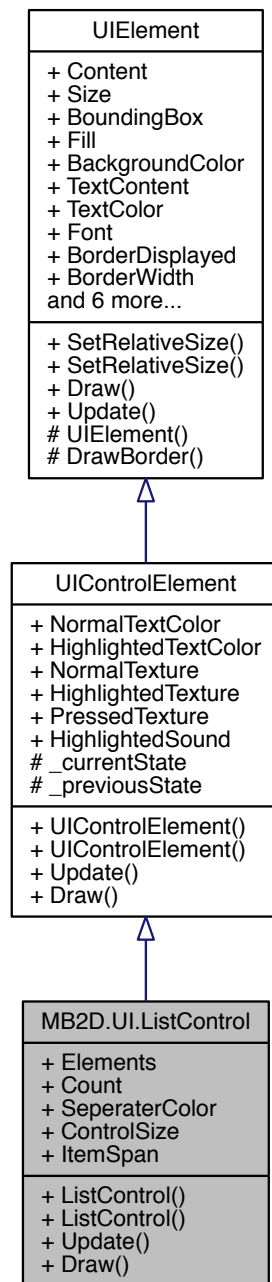
5.24 MB2D.UI.ListControl Class Reference

A scrollable list box. Items can be added and interactied with.

Inheritance diagram for MB2D.UI.ListControl:



Collaboration diagram for MB2D.UI.ListControl:



Public Member Functions

- [ListControl](#) (SpriteFont font)
Initializes a new instance of the T:MB2D.UI.ListControl class using the specified font to draw elements.
- [ListControl](#) (SpriteFont font, Texture2D normal, Texture2D selected, Texture2D pressed)
Initializes a new instance of the T:MB2D.UI.ListControl class with a font to use when drawing elements alongside state textures.

- override void [Update](#) ()
Updates the element to handle scrolling and clicking. Must be called once per frame.
- override void [Draw](#) (SpriteBatch spriteBatch)
Draws the list control to the window

Properties

- List< string > [Elements](#) [get, set]
Gets or sets a list of all of the list control elements
- int [Count](#) [get]
Gets the count of list control elements.
- Color [SeperatorColor](#) [get, set]
Gets or sets the color of the seperater between list elements.
- int [ControlSize](#) [get, set]
Gets or sets the size of the up and down arrow controls.
- int [ItemSpan](#) [get, set]
Gets or sets the span of each item vertically in px.

Additional Inherited Members

5.24.1 Detailed Description

A scrollable list box. Items can be added and interactied with.

5.24.2 Constructor & Destructor Documentation

5.24.2.1 ListControl() [1/2]

```
MB2D.UI.ListControl.ListControl (
    SpriteFont font ) [inline]
```

Initializes a new instance of the T:MB2D.UI.ListControl class using the specified font to draw elements.

Parameters

<i>font</i>	Font to use.
-------------	--------------

5.24.2.2 ListControl() [2/2]

```
MB2D.UI.ListControl.ListControl (
    SpriteFont font,
    Texture2D normal,
    Texture2D selected,
    Texture2D pressed ) [inline]
```

Initializes a new instance of the T:MB2D.UI.ListControl class with a font to use when drawing elements alongside state textures.

Parameters

<i>font</i>	Font to use.
<i>normal</i>	Normal state texture.
<i>selected</i>	Selected stae texture.
<i>pressed</i>	Pressed state texture.

5.24.3 Member Function Documentation**5.24.3.1 Draw()**

```
override void MB2D.UI.ListControl.Draw (
    SpriteBatch spriteBatch ) [inline], [virtual]
```

Draws the list control to the window

Parameters

<i>spriteBatch</i>	Sprite batch to draw to.
--------------------	--------------------------

Reimplemented from [MB2D.UI.UIElement](#).

5.24.3.2 Update()

```
override void MB2D.UI.ListControl.Update ( ) [inline], [virtual]
```

Updates the element to handle scrolling and clicking. Must be called once per frame.

Implements [MB2D.UI.UIElement](#).

5.24.4 Property Documentation**5.24.4.1 ControlSize**

```
int MB2D.UI.ListControl.ControlSize [get], [set]
```

Gets or sets the size of the up and down arrow controls.

The size of the controls.

5.24.4.2 Count

```
int MB2D.UI.ListControl.Count [get]
```

Gets the count of list control elements.

The count.

5.24.4.3 Elements

```
List<string> MB2D.UI.ListControl.Elements [get], [set]
```

Gets or sets a list of all of the list control elements

The elements.

5.24.4.4 ItemSpan

```
int MB2D.UI.ListControl.ItemSpan [get], [set]
```

Gets or sets the span of each item vertically in px.

The item span.

5.24.4.5 SeperatorColor

```
Color MB2D.UI.ListControl.SeperatorColor [get], [set]
```

Gets or sets the color of the seperater between list elements.

The color of the seperater.

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

- MB2D/src/UI/ListControl.cs

5.25 MB2D.MBConsole Class Reference

Midnight Blue debug console class. Executes attached methods and changes attached variables.

Collaboration diagram for MB2D.MBConsole:

MB2D.MBConsole
+ Display + BGColor + TextColor + Vars + Funcs + LastOutput
+ MBConsole() + InitWindow() + AddFunc() + AddVar() + Update() + Draw() + Write() + Write() + Write() + Debug() + Debug() + Debug() + Debug() + Toggle()

Public Member Functions

- **MBConsole** (Color bgColor, Color txtColor, SpriteFont font)
Initializes a new instance of the T:MidnightBlue.MBConsole class.
- void **InitWindow** (GraphicsDevice graphics)
Initializes a graphics target to render the console to
- void **AddFunc** (string name, Action< string[]> func)
Adds a new function to the console for executing in game
- void **AddVar** (string name, object variable)
Adds a new variable to the console for altering in game
- void **Update** ()
Updates any animation until no longer in an animation state. Otherwise calls ProcessInput()
- void **Draw** (SpriteBatch spriteBatch)
Draws the console and associated text to the attached window
- void **Write** (string line)
Writes a line to the console to display
- void **Write** (string line, params string[] args)
Writes a line to the console to display with specified string format information.
- void **Write** (string line, params object[] args)
Writes a line to the console to display with specified string format information.
- void **Debug** (string line, params object[] args)
Writes a debug line to the console with the specified string format information
- void **Debug** (int line, params object[] args)
Writes a debug line to the console with the specified string format information
- void **Debug** (uint line, params object[] args)
Writes a debug line to the console with the specified string format information
- void **Debug** (float line, params object[] args)
Writes a debug line to the console with the specified string format information
- void **Toggle** ()
Toggles the display/hidden state of the console

Properties

- bool **Display** [get, set]
Determines if the console is currently shown or hidden
- Color **BGColor** [get]
Gets the background color of the console
- Color **TextColor** [get]
Gets the color of the console text.
- Dictionary< string, object > **Vars** [get]
Gets the consoles game variables.
- Dictionary< string, Action< string[]> > **Funcs** [get]
Gets the consoles game functions.
- string **LastOutput** [get]

5.25.1 Detailed Description

Midnight Blue debug console class. Executes attached methods and changes attached variables.

5.25.2 Constructor & Destructor Documentation

5.25.2.1 MBConsole()

```
MB2D.MBConsole.MBConsole (
    Color bgColor,
    Color txtColor,
    SpriteFont font ) [inline]
```

Initializes a new instance of the T:MidnightBlue.MBConsole class.

Parameters

<i>bgColor</i>	Background color for rendering the console
<i>txtColor</i>	Text color

5.25.3 Member Function Documentation

5.25.3.1 AddFunc()

```
void MB2D.MBConsole.AddFunc (
    string name,
    Action< string[]> func ) [inline]
```

Adds a new function to the console for executing in game

Parameters

<i>name</i>	Name to use when calling the function in game
<i>func</i>	Function to attach

Here is the caller graph for this function:



5.25.3.2 AddVar()

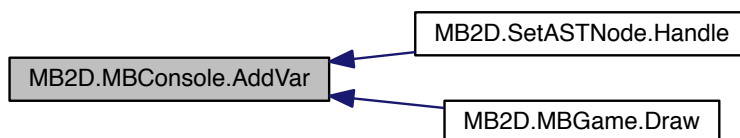
```
void MB2D.MBConsole.AddVar (
    string name,
    object variable ) [inline]
```

Adds a new variable to the console for altering in game

Parameters

<i>name</i>	Name to use when altering the variable in game
<i>variable</i>	Variable to attach

Here is the caller graph for this function:

5.25.3.3 `Debug()` [1/4]

```
void MB2D.MBConsole.Debug (
    string line,
    params object [] args ) [inline]
```

Writes a debug line to the console with the specified string format information

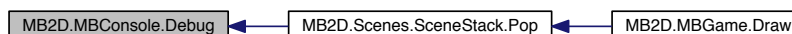
Parameters

<i>line</i>	Line to write
<i>args</i>	Arguments to format into string

Here is the call graph for this function:



Here is the caller graph for this function:



5.25.3.4 Debug() [2/4]

```
void MB2D.MBConsole.Debug (
    int line,
    params object [] args ) [inline]
```

Writes a debug line to the console with the specified string format information

Parameters

<i>line</i>	Line to write
<i>args</i>	Arguments to format into string

Here is the call graph for this function:



5.25.3.5 Debug() [3/4]

```
void MB2D.MBConsole.Debug (
    uint line,
    params object [] args ) [inline]
```

Writes a debug line to the console with the specified string format information

Parameters

<i>line</i>	Line to write
<i>args</i>	Arguments to format into string

Here is the call graph for this function:



5.25.3.6 Debug() [4/4]

```
void MB2D.MBConsole.Debug (
```



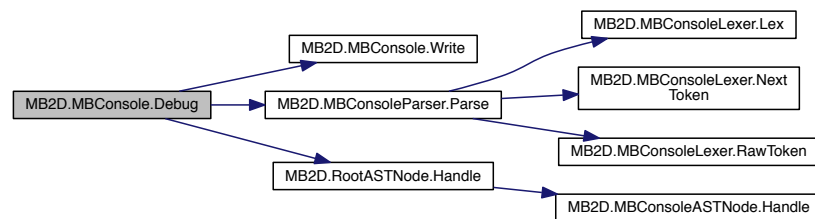
```
float line,
params object [] args ) [inline]
```

Writes a debug line to the console with the specified string format information

Parameters

<i>line</i>	Line to write
<i>args</i>	Arguments to format into string

Here is the call graph for this function:



5.25.3.7 Draw()

```
void MB2D.MBConsole.Draw (
    SpriteBatch spriteBatch ) [inline]
```

Draws the console and associated text to the attached window

Here is the caller graph for this function:



5.25.3.8 InitWindow()

```
void MB2D.MBConsole.InitWindow (
    GraphicsDevice graphics ) [inline]
```

Initializes a graphics target to render the console to

Parameters

<i>graphics</i>	GraphicsDevice to use for rendering
-----------------	-------------------------------------

Here is the caller graph for this function:



5.25.3.9 Toggle()

```
void MB2D.MBConsole.Toggle ( ) [inline]
```

Toggles the display/hide state of the console

Here is the caller graph for this function:



5.25.3.10 Update()

```
void MB2D.MBConsole.Update ( ) [inline]
```

Updates any animation until no longer in an animation state. Otherwise calls `ProcessInput()`

Here is the caller graph for this function:



5.25.3.11 Write() [1/3]

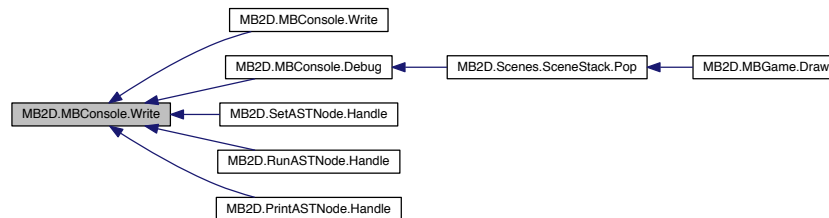
```
void MB2D.MBConsole.Write (
    string line ) [inline]
```

Writes a line to the console to display

Parameters

<i>line</i>	Line to write
-------------	---------------

Here is the caller graph for this function:



5.25.3.12 Write() [2/3]

```
void MB2D.MBConsole.Write (
    string line,
    params string [] args ) [inline]
```

Writes a line to the console to display with specified string format information.

Parameters

<i>line</i>	Line to write
<i>args</i>	Arguments to add to string format

Here is the call graph for this function:



5.25.3.13 Write() [3/3]

```
void MB2D.MBConsole.Write (
    string line,
    params object [] args ) [inline]
```

Writes a line to the console to display with specified string format information.

Parameters

<i>line</i>	Line to write
<i>args</i>	Arguments to format into a string

Here is the call graph for this function:

**5.25.4 Property Documentation****5.25.4.1 BGColor**

```
Color MB2D.MBConsole.BGColor [get]
```

Gets the background color of the console

The background color.

5.25.4.2 Display

```
bool MB2D.MBConsole.Display [get], [set]
```

Determines if the console is currently shown or hidden

true if shown; otherwise, false.

5.25.4.3 Funcs

```
Dictionary<string, Action<string[]>> > MB2D.MBConsole.Funcs [get]
```

Gets the consoles game functions.

The functions.

5.25.4.4 TextColor

```
Color MB2D.MBConsole.TextColor [get]
```

Gets the color of the console text.

The color of the text.

5.25.4.5 Vars

Dictionary<string, object> MB2D.MBConsole.Vars [get]

Gets the consoles game variables.

The variables.

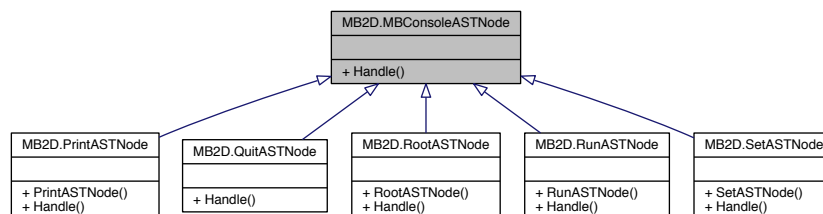
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsole.cs

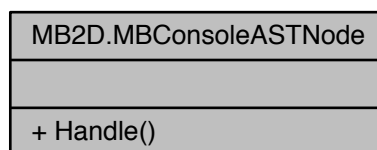
5.26 MB2D.MBConsoleASTNode Class Reference

Class all AST nodes inherit from

Inheritance diagram for MB2D.MBConsoleASTNode:



Collaboration diagram for MB2D.MBConsoleASTNode:



Public Member Functions

- abstract void [Handle](#) (MBConsole console)
Executes specific logic on the console

5.26.1 Detailed Description

Class all AST nodes inherit from

5.26.2 Member Function Documentation

5.26.2.1 Handle()

```
abstract void MB2D.MBConsoleASTNode.Handle (
    MBConsole console ) [pure virtual]
```

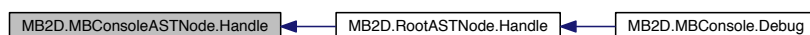
Executes specific logic on the console

Parameters

<i>console</i>	Game console.
----------------	---------------

Implemented in [MB2D.QuitASTNode](#), [MB2D.PrintASTNode](#), [MB2D.RunASTNode](#), [MB2D.SetASTNode](#), and [MB2D.RootASTNode](#).

Here is the caller graph for this function:



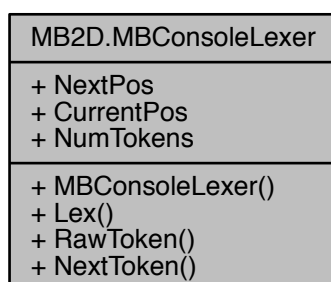
The documentation for this class was generated from the following file:

- `MB2D/src/MBConsole/MBConsoleAST.cs`

5.27 MB2D.MBConsoleLexer Class Reference

Breaks a string into a series of tokens to use for parsing the debug consoles command language

Collaboration diagram for `MB2D.MBConsoleLexer`:



Public Member Functions

- [MBConsoleLexer](#) ()
Initializes a new instance of the T:MidnightBlue.MBConsoleLexer class.
- void [Lex](#) (string command)
Lexes the command string breaking it up into token representation and a second raw string array for retrieving values.
- string [RawToken](#) (int index)
Gets an untokenized representation of a string at a specific index
- [Token](#) [NextToken](#) ()
Gets the next token in the tokenized representation of the command

Properties

- int [NextPos](#) [get]
Gets the next token index in the lexer
- int [CurrentPos](#) [get]
Gets the current token index in the lexer
- int [NumTokens](#) [get]
Gets the number of tokens scanned.

5.27.1 Detailed Description

Breaks a string into a series of tokens to use for parsing the debug consoles command language

5.27.2 Constructor & Destructor Documentation

5.27.2.1 MBConsoleLexer()

```
MB2D.MBConsoleLexer.MBConsoleLexer ( ) [inline]
```

Initializes a new instance of the T:MidnightBlue.MBConsoleLexer class.

5.27.3 Member Function Documentation

5.27.3.1 Lex()

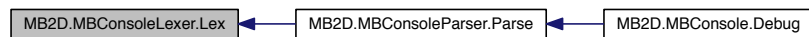
```
void MB2D.MBConsoleLexer.Lex (
    string command ) [inline]
```

Lexes the command string breaking it up into token representation and a second raw string array for retrieving values.

Parameters

<i>command</i>	Command string to scan.
----------------	-------------------------

Here is the caller graph for this function:

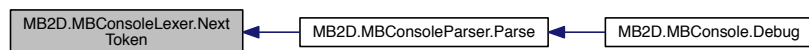


5.27.3.2 NextToken()

```
Token MB2D.MBConsoleLexer.NextToken ( ) [inline]
```

Gets the next token in the tokenized representation of the command

The next token. Here is the caller graph for this function:



5.27.3.3 RawToken()

```
string MB2D.MBConsoleLexer.RawToken (
    int index ) [inline]
```

Gets an untokenized representation of a string at a specific index

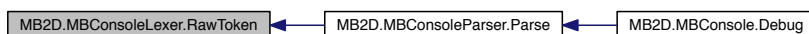
Returns

The string representation.

Parameters

<i>index</i>	Index.
--------------	--------

Here is the caller graph for this function:



5.27.4 Property Documentation

5.27.4.1 CurrentPos

```
int MB2D.MBConsoleLexer.CurrentPos [get]
```

Gets the current token index in the lexer

The current position.

5.27.4.2 NextPos

```
int MB2D.MBConsoleLexer.NextPos [get]
```

Gets the next token index in the lexer

The next position.

5.27.4.3 NumTokens

```
int MB2D.MBConsoleLexer.NumTokens [get]
```

Gets the number of tokens scanned.

The number of tokens.

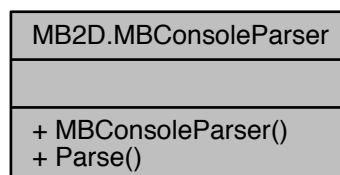
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsoleLexer.cs

5.28 MB2D.MBConsoleParser Class Reference

Parses command string input and executes it using a debug console.

Collaboration diagram for MB2D.MBConsoleParser:



Public Member Functions

- [MBConsoleParser \(\)](#)
Initializes a new instance of the T:MB2D.MBConsoleParser class.
- [RootASTNode Parse](#) (string command)
Processes the command string and executes to the given console.

5.28.1 Detailed Description

Parses command string input and executes it using a debug console.

5.28.2 Constructor & Destructor Documentation

5.28.2.1 MBConsoleParser()

```
MB2D.MBConsoleParser.MBConsoleParser ( ) [inline]
```

Initializes a new instance of the T:MB2D.MBConsoleParser class.

5.28.3 Member Function Documentation

5.28.3.1 Parse()

```
RootASTNode MB2D.MBConsoleParser.Parse (
    string command ) [inline]
```

Processes the command string and executes to the given console.

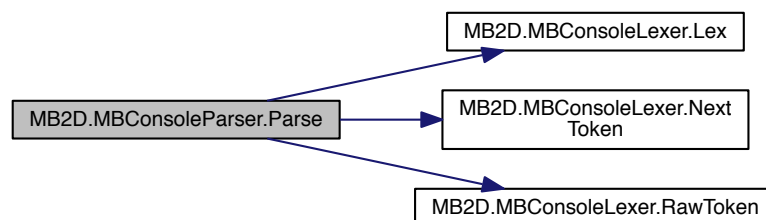
Parameters

<i>console</i>	Console to execute on.
<i>command</i>	Command to parse.

Returns

The entry point for the AST

Here is the call graph for this function:



Here is the caller graph for this function:



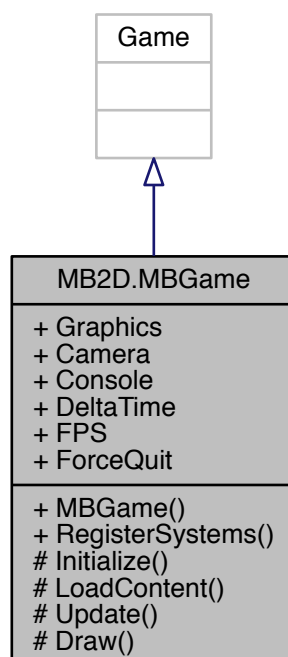
The documentation for this class was generated from the following file:

- `MB2D/src/MBConsole/MBConsoleParser.cs`

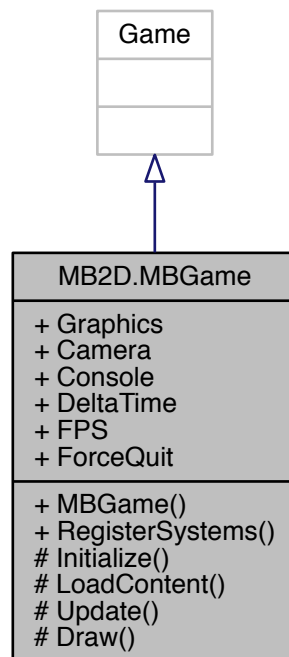
5.29 MB2D.MBGame Class Reference

This is the main type for your game.

Inheritance diagram for MB2D.MBGame:



Collaboration diagram for MB2D.MBGame:



Public Member Functions

- **MBGame** (Type initSceneType)
Initializes a new instance of the T:MB2D.MBGame class and defines essential graphics settings.
- void **RegisterSystems** ()
Registers all EntitySystems used in the engine

Protected Member Functions

- override void **Initialize** ()
Allows the game to perform any initialization it needs to before starting to run. This is where it can query for any required services and load any non-graphic related content. Calling base.Initialize will enumerate through any components and initialize them as well.
- override void **LoadContent** ()
Loads content at the beginning of the game
- override void **Update** (GameTime gameTime)
Allows the game to run logic such as updating the world, checking for collisions, gathering input, and playing audio.
- override void **Draw** (GameTime gameTime)
Draws the current scene to the window

Properties

- static GraphicsDevice [Graphics](#) [get]
Gets the main graphics device.
- static Camera2D [Camera](#) [get]
Gets the main camera.
- static MBConsole [Console](#) [get]
Gets the debug console for reading and writing to. There should only ever be one of these
- static float [DeltaTime](#) [get]
Gets time it took to complete the last frame
- static float [FPS](#) [get]
Gets the current average frames per second
- static bool [ForceQuit](#) [get, set]
Gets or sets a value indicating whether this T:MB2D.MBGame should quit.

5.29.1 Detailed Description

This is the main type for your game.

5.29.2 Constructor & Destructor Documentation

5.29.2.1 MBGame()

```
MB2D.MBGame.MBGame (
    Type initSceneType ) [inline]
```

Initializes a new instance of the T:MB2D.MBGame class and defines essential graphics settings.

5.29.3 Member Function Documentation

5.29.3.1 Draw()

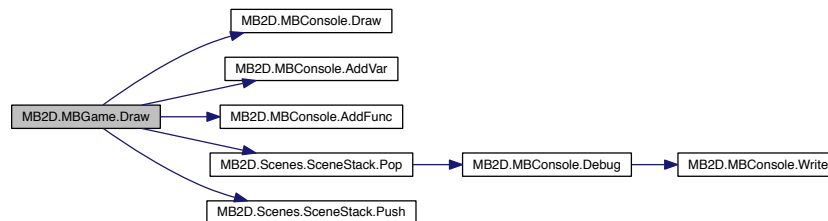
```
override void MB2D.MBGame.Draw (
    GameTime gameTime ) [inline], [protected]
```

Draws the current scene to the window

Parameters

<i>gameTime</i>	Provides a snapshot of timing values.
-----------------	---------------------------------------

Here is the call graph for this function:

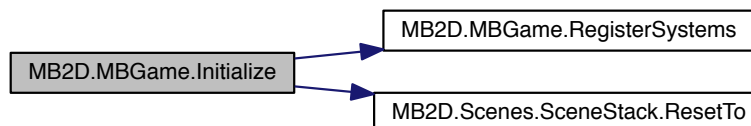


5.29.3.2 Initialize()

```
override void MB2D.MBGame.Initialize ( ) [inline], [protected]
```

Allows the game to perform any initialization it needs to before starting to run. This is where it can query for any required services and load any non-graphic related content. Calling `base.Initialize` will enumerate through any components and initialize them as well.

Here is the call graph for this function:



5.29.3.3 LoadContent()

```
override void MB2D.MBGame.LoadContent ( ) [inline], [protected]
```

Loads content at the beginning of the game

Here is the call graph for this function:



5.29.3.4 RegisterSystems()

```
void MB2D.MBGame.RegisterSystems ( ) [inline]
```

Registers all EntitySystems used in the engine

Here is the caller graph for this function:



5.29.3.5 Update()

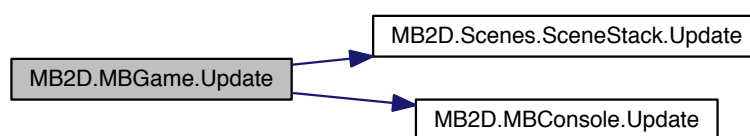
```
override void MB2D.MBGame.Update (
    GameTime gameTime ) [inline], [protected]
```

Allows the game to run logic such as updating the world, checking for collisions, gathering input, and playing audio.

Parameters

<i>gameTime</i>	Provides a snapshot of timing values.
-----------------	---------------------------------------

Here is the call graph for this function:



5.29.4 Property Documentation

5.29.4.1 Camera

```
Camera2D MB2D.MBGame.Camera [static], [get]
```

Gets the main camera.

The main camera.

5.29.4.2 Console

```
MBConsole MB2D.MBGame.Console [static], [get]
```

Gets the debug console for reading and writing to. There should only ever be one of these

The debug console.

5.29.4.3 DeltaTime

```
float MB2D.MBGame.DeltaTime [static], [get]
```

Gets time it took to complete the last frame

The delta time.

5.29.4.4 ForceQuit

```
bool MB2D.MBGame.ForceQuit [static], [get], [set]
```

Gets or sets a value indicating whether this T:MB2D.MBGame should quit.

true if the game should quit; otherwise, false.

5.29.4.5 FPS

```
float MB2D.MBGame.FPS [static], [get]
```

Gets the current average frames per second

The fps.

5.29.4.6 Graphics

```
GraphicsDevice MB2D.MBGame.Graphics [static], [get]
```

Gets the main graphics device.

The graphics device.

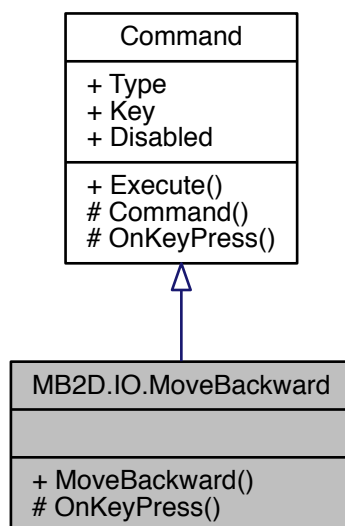
The documentation for this class was generated from the following file:

- MB2D/src/MBGame.cs

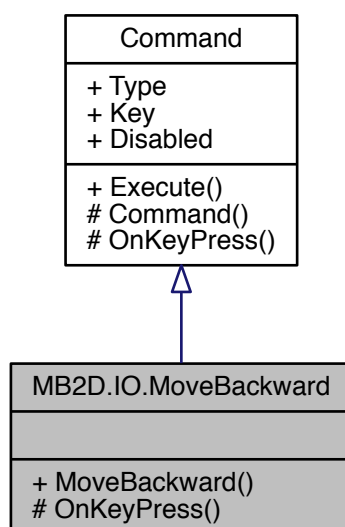
5.30 MB2D.IO.MoveBackward Class Reference

Moves an entity backward. Only runs on entities with a physics component

Inheritance diagram for MB2D.IO.MoveBackward:



Collaboration diagram for MB2D.IO.MoveBackward:



Public Member Functions

- [MoveBackward](#) (Keys key, [CommandType](#) type)
Initializes a new instance of the T:MB2D.IO.MoveBackward class.

Protected Member Functions

- override void [OnKeyPress](#) ([Entity](#) e=null)
Move an entity forward based on their velocity

Additional Inherited Members

5.30.1 Detailed Description

Moves an entity backward. Only runs on entities with a physics component

5.30.2 Constructor & Destructor Documentation

5.30.2.1 MoveBackward()

```
MB2D.IO.MoveBackward.MoveBackward (
    Keys key,
    CommandType type ) [inline]
```

Initializes a new instance of the T:MB2D.IO.MoveBackward class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.30.3 Member Function Documentation

5.30.3.1 OnKeyPress()

```
override void MB2D.IO.MoveBackward.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Move an entity forward based on their velocity

Parameters

<i>e</i>	Entity to move.
----------	-----------------

Implements [MB2D.IO.Command](#).

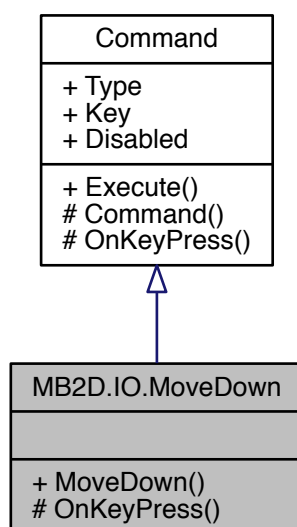
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

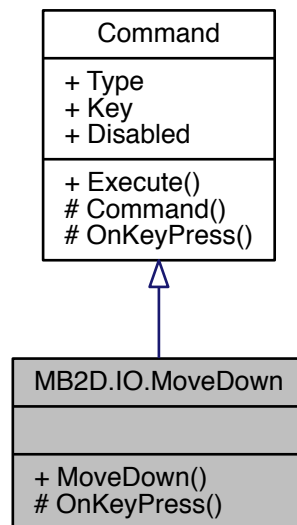
5.31 MB2D.IO.MoveDown Class Reference

Moves an entity down

Inheritance diagram for MB2D.IO.MoveDown:



Collaboration diagram for MB2D.IO.MoveDown:



Public Member Functions

- `MoveDown` (Keys key, `CommandType` type)
Initializes a new instance of the T:MB2D.IO.MoveDown class.

Protected Member Functions

- override void `OnKeyPress` (Entity e=null)
Move an entity down

Additional Inherited Members

5.31.1 Detailed Description

Moves an entity down

5.31.2 Constructor & Destructor Documentation

5.31.2.1 MoveDown()

```

MB2D.IO.MoveDown.MoveDown (
    Keys key,
    CommandType type ) [inline]
  
```

Initializes a new instance of the T:MB2D.IO.MoveDown class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.31.3 Member Function Documentation

5.31.3.1 OnKeyPress()

```
override void MB2D.IO.MoveDown.OnKeyPress (  
    Entity e = null ) [inline], [protected], [virtual]
```

Move an entity down

Parameters

<i>e</i>	Entity to move.
----------	-----------------

Implements [MB2D.IO.Command](#).

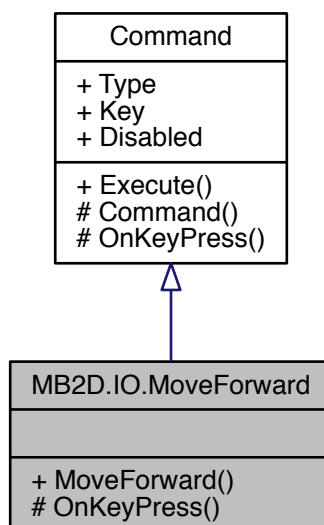
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

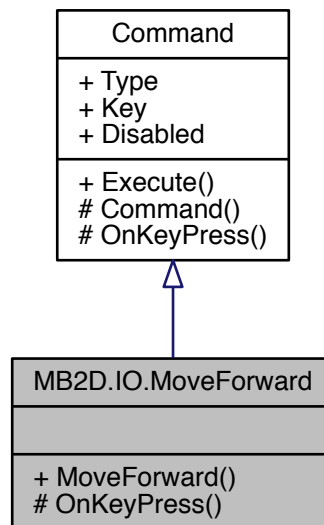
5.32 MB2D.IO.MoveForward Class Reference

Moves an entity forward. Only runs on entities with a physics component

Inheritance diagram for MB2D.IO.MoveForward:



Collaboration diagram for MB2D.IO.MoveForward:



Public Member Functions

- [MoveForward](#) (Keys key, [CommandType](#) type)
Initializes a new instance of the T:MB2D.IO.MoveForward class.

Protected Member Functions

- override void [OnKeyPress](#) ([Entity](#) e=null)
Move an entity forward based on their velocity

Additional Inherited Members

5.32.1 Detailed Description

Moves an entity forward. Only runs on entities with a physics component

5.32.2 Constructor & Destructor Documentation

5.32.2.1 MoveForward()

```

MB2D.IO.MoveForward.MoveForward (
    Keys key,
    CommandType type ) [inline]
  
```

Initializes a new instance of the T:MB2D.IO.MoveForward class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.32.3 Member Function Documentation

5.32.3.1 OnKeyPress()

```
override void MB2D.IO.MoveForward.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Move an entity forward based on their velocity

Parameters

<i>e</i>	Entity to move.
----------	-----------------

Implements [MB2D.IO.Command](#).

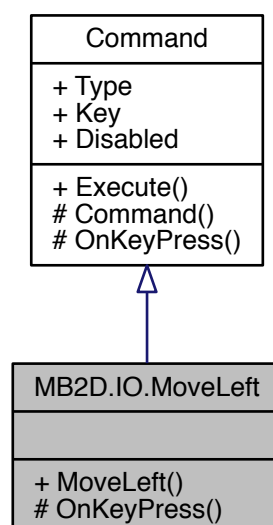
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

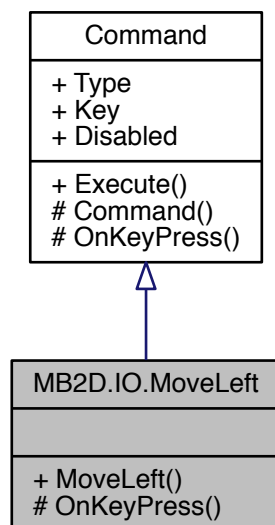
5.33 MB2D.IO.MoveLeft Class Reference

Moves an entity left.

Inheritance diagram for MB2D.IO.MoveLeft:



Collaboration diagram for MB2D.IO.MoveLeft:



Public Member Functions

- [MoveLeft](#) (Keys key, [CommandType](#) type)
Initializes a new instance of the T:MB2D.IO.MoveLeft class.

Protected Member Functions

- override void [OnKeyPress](#) (Entity e=null)
Move an entity up

Additional Inherited Members

5.33.1 Detailed Description

Moves an entity left.

5.33.2 Constructor & Destructor Documentation

5.33.2.1 MoveLeft()

```

MB2D.IO.MoveLeft.MoveLeft (
    Keys key,
    CommandType type ) [inline]
  
```

Initializes a new instance of the T:MB2D.IO.MoveLeft class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.33.3 Member Function Documentation

5.33.3.1 OnKeyPress()

```
override void MB2D.IO.MoveLeft.OnKeyPress (  
    Entity e = null ) [inline], [protected], [virtual]
```

Move an entity up

Parameters

<i>e</i>	Entity to move.
----------	-----------------

Implements [MB2D.IO.Command](#).

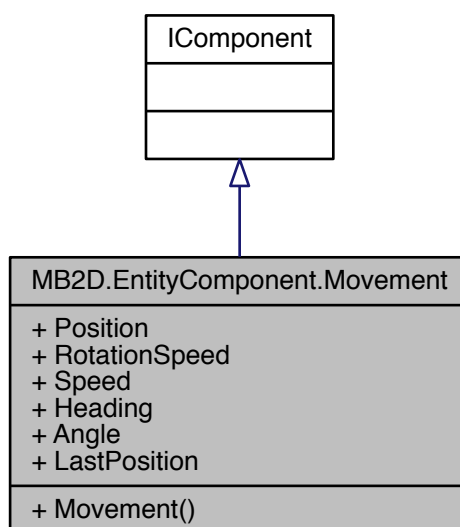
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

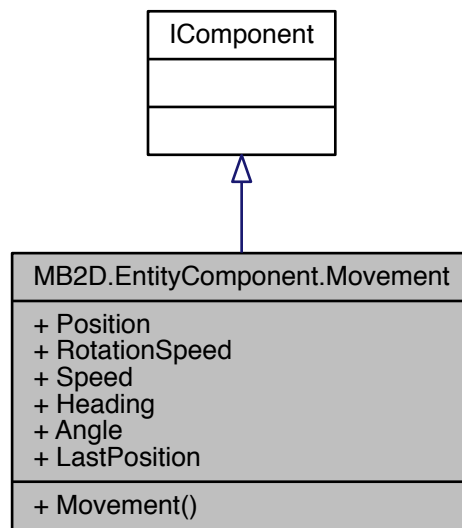
5.34 MB2D.EntityComponent.Movement Class Reference

Defines position, rotation and speed related data for moving an entity.

Inheritance diagram for MB2D.EntityComponent.Movement:



Collaboration diagram for MB2D.EntityComponent.Movement:



Public Member Functions

- [Movement](#) (float speed=0.0f, float rotationSpeed=0.0f)
Initializes a new instance of the T:MB2D.EntityComponent.Movement class.

Properties

- Vector2 [Position](#) [get, set]
Gets or sets the world position.
- float [RotationSpeed](#) [get, set]
Gets or sets the rotation speed.
- float [Speed](#) [get, set]
Gets or sets the movement speed.
- Vector2 [Heading](#) [get, set]
Gets or sets the current heading.
- float [Angle](#) [get, set]
Gets or sets the angle.
- Vector2 [LastPosition](#) [get, set]
Gets or sets the last known position.

5.34.1 Detailed Description

Defines position, rotation and speed related data for moving an entity.

5.34.2 Constructor & Destructor Documentation

5.34.2.1 Movement()

```
MB2D.EntityComponent.Movement.Movement (
    float speed = 0.0f,
    float rotationSpeed = 0.0f ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.Movement class.

Parameters

<i>speed</i>	Initial speed value.
<i>rotationSpeed</i>	Initial rotation speed value.

5.34.3 Property Documentation

5.34.3.1 Angle

```
float MB2D.EntityComponent.Movement.Angle [get], [set]
```

Gets or sets the angle.

The angle in radians.

5.34.3.2 Heading

```
Vector2 MB2D.EntityComponent.Movement.Heading [get], [set]
```

Gets or sets the current heading.

The heading.

5.34.3.3 LastPosition

```
Vector2 MB2D.EntityComponent.Movement.LastPosition [get], [set]
```

Gets or sets the last known position.

The last position.

5.34.3.4 Position

```
Vector2 MB2D.EntityComponent.Movement.Position [get], [set]
```

Gets or sets the world position.

The position.

5.34.3.5 RotationSpeed

```
float MB2D.EntityComponent.Movement.RotationSpeed [get], [set]
```

Gets or sets the rotation speed.

The rotation speed.

5.34.3.6 Speed

```
float MB2D.EntityComponent.Movement.Speed [get], [set]
```

Gets or sets the movement speed.

The speed.

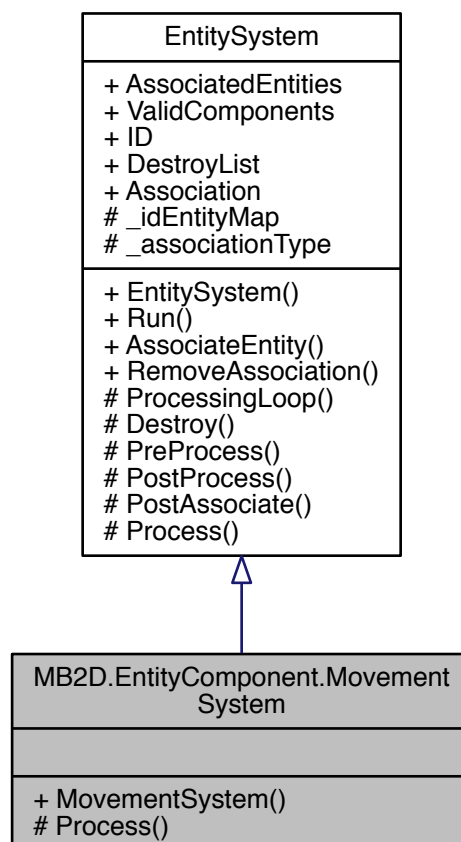
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Components/Movement.cs

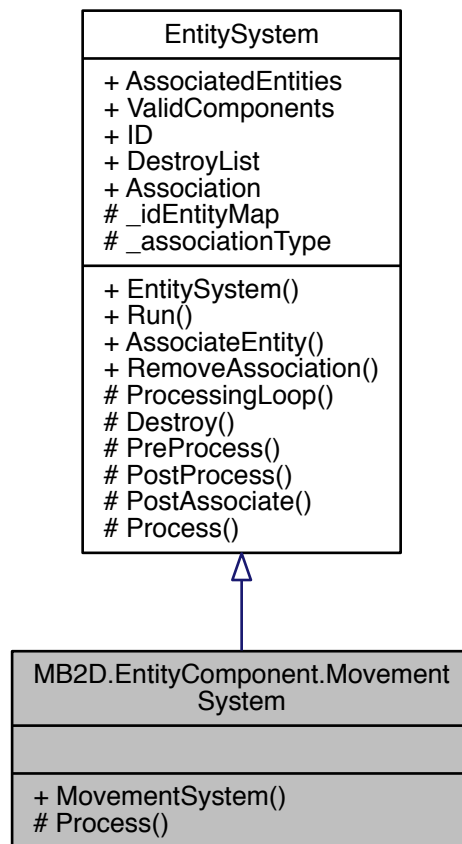
5.35 MB2D.EntityComponent.MovementSystem Class Reference

Processes the change in position, rotation, and sprite transform for an entity

Inheritance diagram for MB2D.EntityComponent.MovementSystem:



Collaboration diagram for MB2D.EntityComponent.MovementSystem:



Public Member Functions

- [MovementSystem](#) ()

Initializes a new instance of the T:MB2D.EntityComponent.MovementSystem class.

Protected Member Functions

- override void [Process](#) ([Entity](#) entity)

Processes the movement for the specific entity

Additional Inherited Members

5.35.1 Detailed Description

Processes the change in position, rotation, and sprite transform for an entity

5.35.2 Constructor & Destructor Documentation

5.35.2.1 MovementSystem()

```
MB2D.EntityComponent.MovementSystem.MovementSystem ( ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.MovementSystem class.

5.35.3 Member Function Documentation

5.35.3.1 Process()

```
override void MB2D.EntityComponent.MovementSystem.Process (
    Entity entity ) [inline], [protected], [virtual]
```

Processes the movement for the specific entity

Parameters

<i>entity</i>	Entity to operate on.
---------------	-----------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

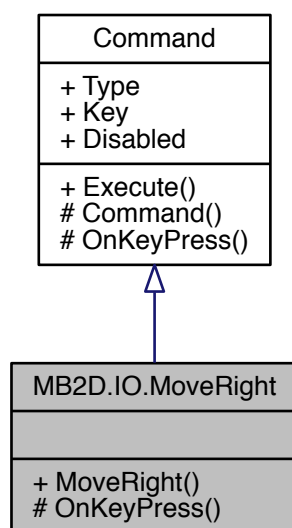
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Systems/MovementSystem.cs

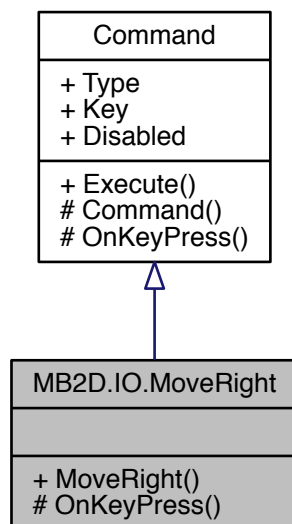
5.36 MB2D.IO.MoveRight Class Reference

Moves an entity right

Inheritance diagram for MB2D.IO.MoveRight:



Collaboration diagram for MB2D.IO.MoveRight:



Public Member Functions

- [MoveRight](#) (Keys key, [CommandType](#) type)

Initializes a new instance of the T:MB2D.IO.MoveRight class.

Protected Member Functions

- override void [OnKeyPress](#) ([Entity](#) e=null)
Move an entity right

Additional Inherited Members

5.36.1 Detailed Description

Moves an entity right

5.36.2 Constructor & Destructor Documentation

5.36.2.1 MoveRight()

```
MB2D.IO.MoveRight.MoveRight (
    Keys key,
    CommandType type ) [inline]
```

Initializes a new instance of the T:MB2D.IO.MoveRight class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.36.3 Member Function Documentation

5.36.3.1 OnKeyPress()

```
override void MB2D.IO.MoveRight.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Move an entity right

Parameters

<i>e</i>	Entity to move.
----------	-----------------

Implements [MB2D.IO.Command](#).

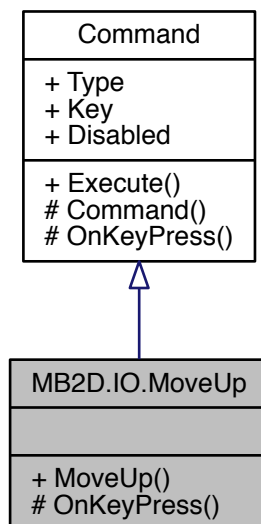
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

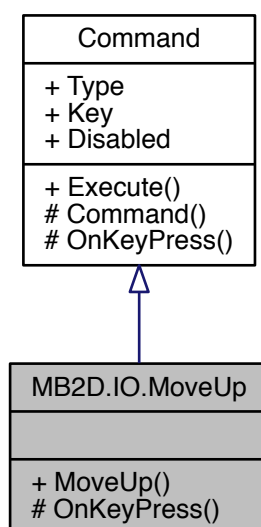
5.37 MB2D.IO.MoveUp Class Reference

Moves a player controller up

Inheritance diagram for MB2D.IO.MoveUp:



Collaboration diagram for MB2D.IO.MoveUp:



Public Member Functions

- [MoveUp](#) (Keys key, [CommandType](#) type)
Initializes a new instance of the T:MB2D.IO.MoveUp class.

Protected Member Functions

- override void [OnKeyPress](#) ([Entity](#) e=null)
Move an entity up

Additional Inherited Members

5.37.1 Detailed Description

Moves a player controller up

5.37.2 Constructor & Destructor Documentation

5.37.2.1 MoveUp()

```
MB2D.IO.MoveUp.MoveUp (
    Keys key,
    CommandType type ) [inline]
```

Initializes a new instance of the T:MB2D.IO.MoveUp class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.37.3 Member Function Documentation

5.37.3.1 OnKeyPress()

```
override void MB2D.IO.MoveUp.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Move an entity up

Parameters

<i>e</i>	Entity to move.
----------	-----------------

Implements [MB2D.IO.Command](#).

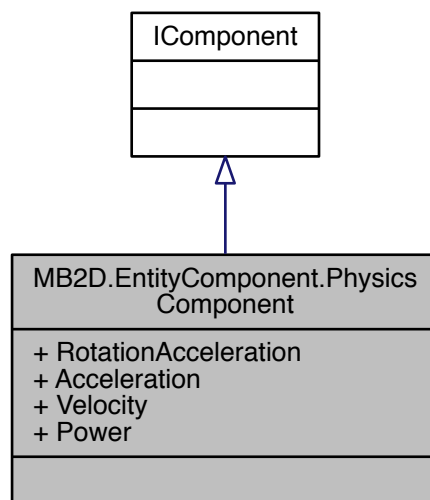
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

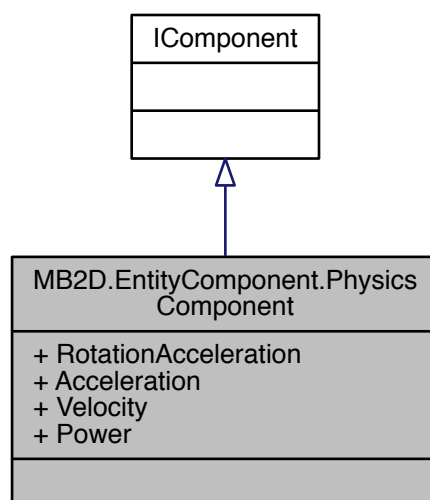
5.38 MB2D.EntityComponent.PhysicsComponent Class Reference

Physics component used to define acceleration and velocity.

Inheritance diagram for MB2D.EntityComponent.PhysicsComponent:



Collaboration diagram for MB2D.EntityComponent.PhysicsComponent:



Properties

- float [RotationAcceleration](#) [get, set]
Gets or sets the current acceleration of the rotation.
- Vector2 [Acceleration](#) [get, set]
Gets or sets the current acceleration.
- Vector2 [Velocity](#) [get, set]
Gets or sets the current positional velocity.
- float [Power](#) [get, set]
Gets or sets the current power applied.

5.38.1 Detailed Description

Physics component used to define acceleration and velocity.

5.38.2 Property Documentation

5.38.2.1 Acceleration

```
Vector2 MB2D.EntityComponent.PhysicsComponent.Acceleration [get], [set]
```

Gets or sets the current acceleration.

The acceleration.

5.38.2.2 Power

```
float MB2D.EntityComponent.PhysicsComponent.Power [get], [set]
```

Gets or sets the current power applied.

The power applied.

5.38.2.3 RotationAcceleration

```
float MB2D.EntityComponent.PhysicsComponent.RotationAcceleration [get], [set]
```

Gets or sets the current acceleration of the rotation.

The rotation acceleration.

5.38.2.4 Velocity

```
Vector2 MB2D.EntityComponent.PhysicsComponent.Velocity [get], [set]
```

Gets or sets the current positional velocity.

The velocity.

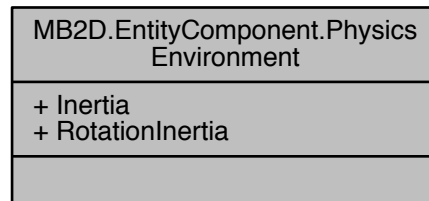
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Components/PhysicsComponent.cs

5.39 MB2D.EntityComponent.PhysicsEnvironment Class Reference

Defines a new environment to feed into the physics system to alter the impact it has on an entity

Collaboration diagram for MB2D.EntityComponent.PhysicsEnvironment:



Properties

- float [Inertia](#) [get, set]
Gets or sets the inertia of the environment.
- float [RotationInertia](#) [get, set]
Gets or sets the rotation inertia of the environment.

5.39.1 Detailed Description

Defines a new environment to feed into the physics system to alter the impact it has on an entity

5.39.2 Property Documentation

5.39.2.1 Inertia

```
float MB2D.EntityComponent.PhysicsEnvironment.Inertia [get], [set]
```

Gets or sets the inertia of the environment.

The inertia.

5.39.2.2 RotationInertia

```
float MB2D.EntityComponent.PhysicsEnvironment.RotationInertia [get], [set]
```

Gets or sets the rotation inertia of the environment.

The rotation inertia.

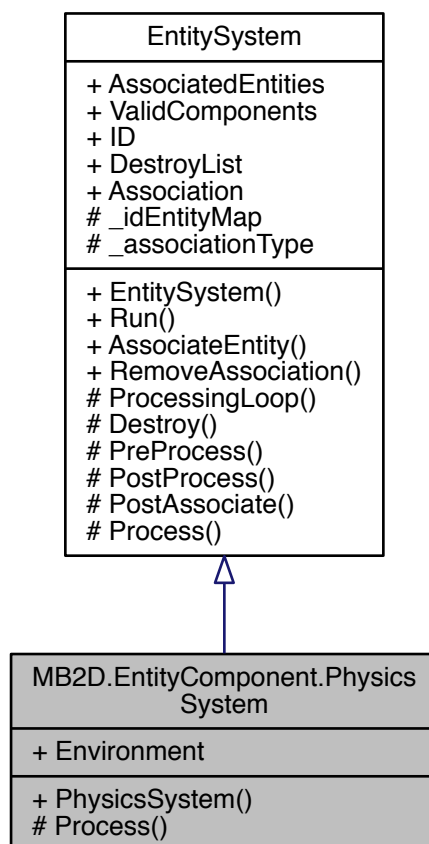
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Systems/PhysicsSystem.cs

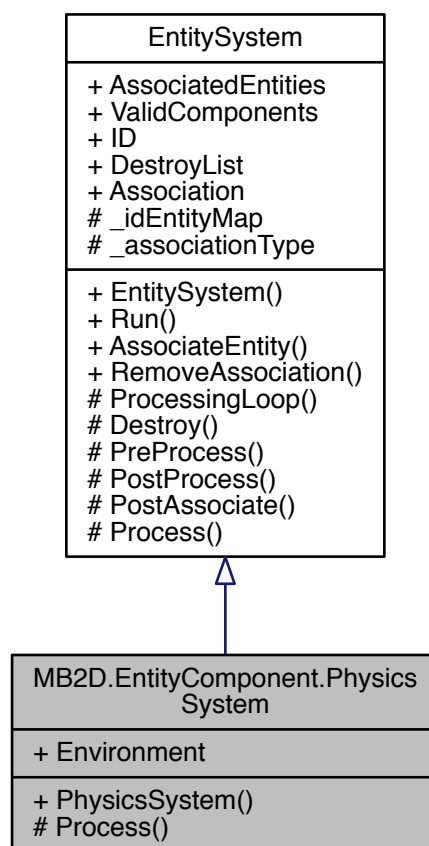
5.40 MB2D.EntityComponent.PhysicsSystem Class Reference

Processes physics changes for a given entity

Inheritance diagram for MB2D.EntityComponent.PhysicsSystem:



Collaboration diagram for MB2D.EntityComponent.PhysicsSystem:



Public Member Functions

- [PhysicsSystem \(\)](#)

Initializes a new instance of the T:MB2D.EntityComponent.PhysicsSystem class.

Protected Member Functions

- override void [Process \(Entity entity\)](#)

Updates the entities movement and velocity values based on the current physics environment

Properties

- [PhysicsEnvironment Environment](#) [get, set]

Gets or sets the current physics environment.

Additional Inherited Members

5.40.1 Detailed Description

Processes physics changes for a given entity

5.40.2 Constructor & Destructor Documentation

5.40.2.1 PhysicsSystem()

```
MB2D.EntityComponent.PhysicsSystem.PhysicsSystem ( ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.PhysicsSystem class.

5.40.3 Member Function Documentation

5.40.3.1 Process()

```
override void MB2D.EntityComponent.PhysicsSystem.Process (
    Entity entity ) [inline], [protected], [virtual]
```

Updates the entities movement and velocity values based on the current physics environment

Parameters

<i>entity</i>	Entity to process.
---------------	------------------------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

5.40.4 Property Documentation

5.40.4.1 Environment

```
PhysicsEnvironment MB2D.EntityComponent.PhysicsSystem.Environment [get], [set]
```

Gets or sets the current physics environment.

The environment.

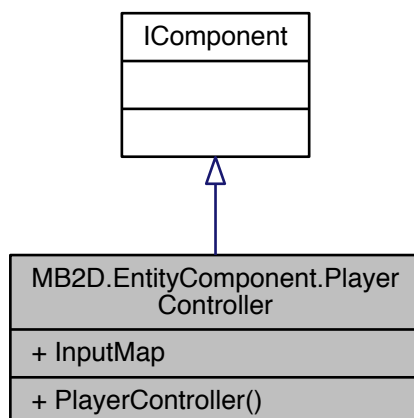
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Systems/PhysicsSystem.cs

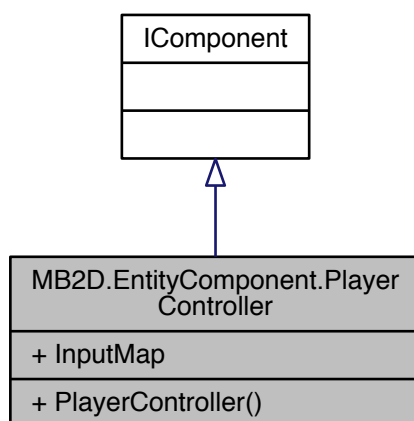
5.41 MB2D.EntityComponent.PlayerController Class Reference

Defines the attached entity as controllable

Inheritance diagram for MB2D.EntityComponent.PlayerController:



Collaboration diagram for MB2D.EntityComponent.PlayerController:



Public Member Functions

- [PlayerController](#) ()

Initializes a new instance of the T:MidnightBlue.PlayerController component with default input assignment

Properties

- [InputMap InputMap](#) [get]
Gets the input map.

5.41.1 Detailed Description

Defines the attached entity as controllable

5.41.2 Constructor & Destructor Documentation

5.41.2.1 PlayerController()

```
MB2D.EntityComponent.PlayerController.PlayerController ( ) [inline]
```

Initializes a new instance of the T:MidnightBlue.PlayerController component with default input assignment

5.41.3 Property Documentation

5.41.3.1 InputMap

```
InputMap MB2D.EntityComponent.PlayerController.InputMap [get]
```

Gets the input map.

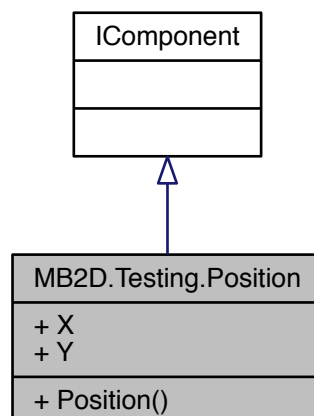
The input map.

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

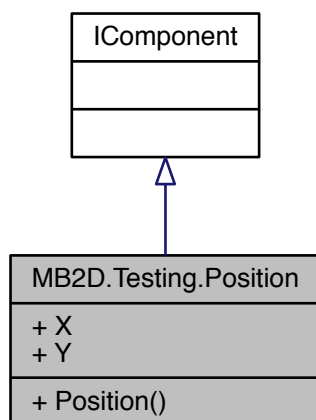
- MB2D/src/EntityComponent/Components/PlayerController.cs

5.42 MB2D.Testing.Position Class Reference

Inheritance diagram for MB2D.Testing.Position:



Collaboration diagram for MB2D.Testing.Position:



Public Member Functions

- **Position** (int x=0, int y=0)

Properties

- int **X** [get, set]
- int **Y** [get, set]

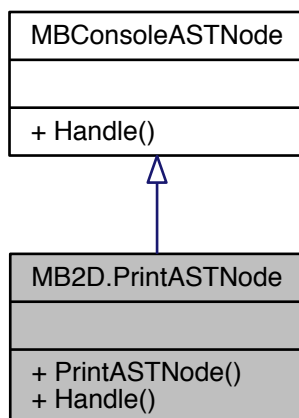
The documentation for this class was generated from the following file:

- MB2D/src/Test/TestSystem.cs

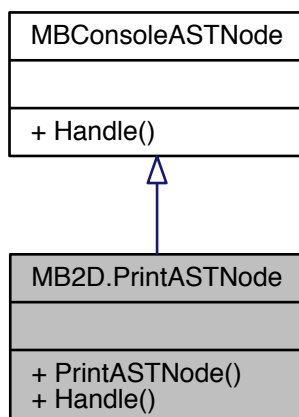
5.43 MB2D.PrintASTNode Class Reference

Prints a variable to the console.

Inheritance diagram for MB2D.PrintASTNode:



Collaboration diagram for MB2D.PrintASTNode:



Public Member Functions

- [PrintASTNode](#) (string var, [Token](#) type)

Initializes a new instance of the T:MB2D.PrintASTNode class.

- override void [Handle](#) ([MBConsole](#) console)

Prints the variable to the console. If it's not a previously assigned variable, it will print the variable name itself as if it's an immediate value.

5.43.1 Detailed Description

Prints a variable to the console.

5.43.2 Constructor & Destructor Documentation

5.43.2.1 PrintASTNode()

```
MB2D.PrintASTNode.PrintASTNode (
    string var,
    Token type ) [inline]
```

Initializes a new instance of the T:MB2D.PrintASTNode class.

Parameters

<i>var</i>	The variables identifier or, if not found, the print statements argument.
<i>type</i>	The variables token type

5.43.3 Member Function Documentation

5.43.3.1 Handle()

```
override void MB2D.PrintASTNode.Handle (
    MBConsole console ) [inline], [virtual]
```

Prints the variable to the console. If it's not a previously assigned variable, it will print the variable name itself as if it's an immediate value.

Parameters

<i>console</i>	Console to print to.
----------------	----------------------

Implements [MB2D.MBConsoleASTNode](#).

Here is the call graph for this function:



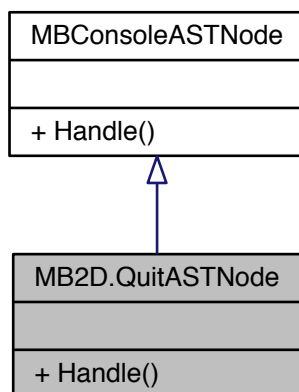
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsoleAST.cs

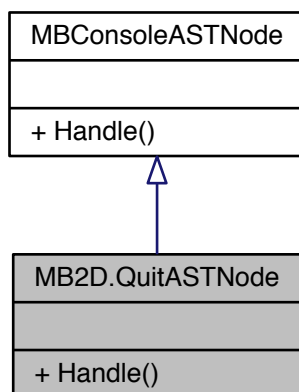
5.44 MB2D.QuitASTNode Class Reference

Handles quitting the game

Inheritance diagram for MB2D.QuitASTNode:



Collaboration diagram for MB2D.QuitASTNode:



Public Member Functions

- override void [Handle](#) ([MBConsole](#) console)
Quits the game

5.44.1 Detailed Description

Handles quitting the game

5.44.2 Member Function Documentation

5.44.2.1 Handle()

```
override void MB2D.QuitASTNode.Handle (  
    MBConsole console ) [inline], [virtual]
```

Quits the game

Parameters

<i>console</i>	Console to handle.
----------------	--------------------

Implements [MB2D.MBConsoleASTNode](#).

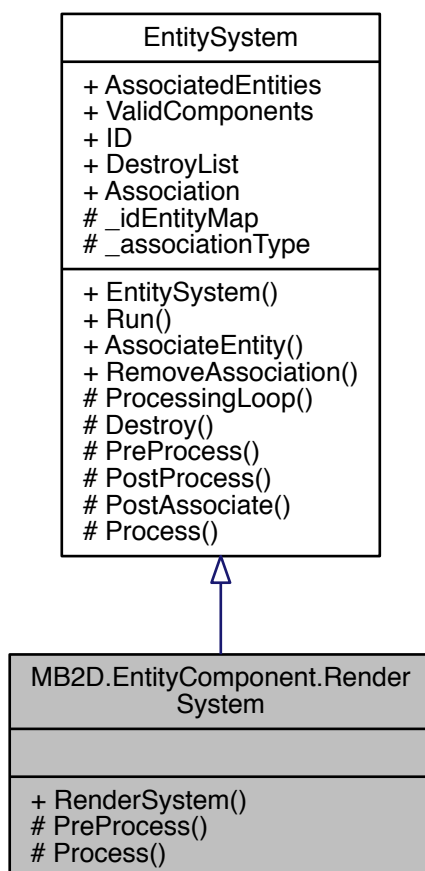
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsoleAST.cs

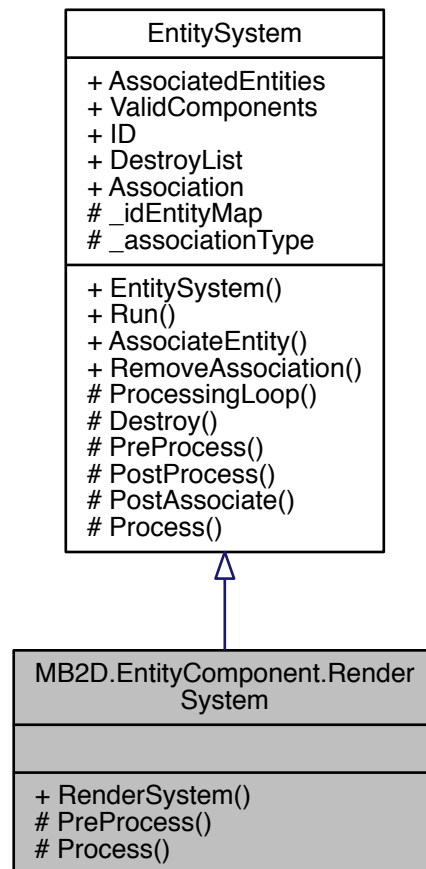
5.45 MB2D.EntityComponent.RenderSystem Class Reference

Renders culled entities with a [SpriteTransform](#) to the window

Inheritance diagram for MB2D.EntityComponent.RenderSystem:



Collaboration diagram for MB2D.EntityComponent.RenderSystem:



Public Member Functions

- [RenderSystem](#) (SpriteBatch spriteBatch)
Initializes a new instance of the T:MB2D.EntityComponent.RenderSystem class.

Protected Member Functions

- override void [PreProcess](#) ()
Re-orders the list of AssociatedEntities based on their current z-index
- override void [Process](#) (Entity entity)
Culls and then draws an entity to the window

Additional Inherited Members

5.45.1 Detailed Description

Renders culled entities with a [SpriteTransform](#) to the window

5.45.2 Constructor & Destructor Documentation

5.45.2.1 RenderSystem()

```
MB2D.EntityComponent.RenderSystem.RenderSystem (
    SpriteBatch spriteBatch ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.RenderSystem class.

Parameters

<i>spriteBatch</i>	Sprite batch to draw to.
--------------------	--------------------------

5.45.3 Member Function Documentation

5.45.3.1 PreProcess()

```
override void MB2D.EntityComponent.RenderSystem.PreProcess ( ) [inline], [protected], [virtual]
```

Re-orders the list of AssociatedEntities based on their current z-index

Reimplemented from [MB2D.EntityComponent.EntitySystem](#).

5.45.3.2 Process()

```
override void MB2D.EntityComponent.RenderSystem.Process (
    Entity entity ) [inline], [protected], [virtual]
```

Culls and then draws an entity to the window

Parameters

<i>entity</i>	Entity .
---------------	--------------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

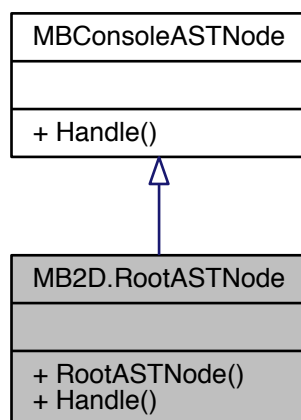
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Systems/RenderSystem.cs

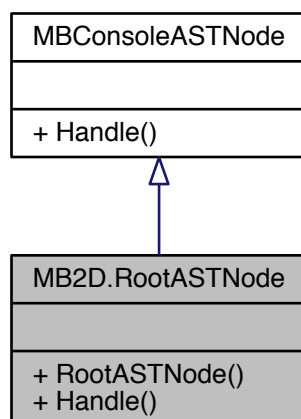
5.46 MB2D.RootASTNode Class Reference

The entry point for command execution with a single child.

Inheritance diagram for MB2D.RootASTNode:



Collaboration diagram for MB2D.RootASTNode:



Public Member Functions

- [RootASTNode](#) ([MBConsoleASTNode](#) child)
Initializes a new instance of the T:MB2D.RootASTNode class.
- override void [Handle](#) ([MBConsole](#) console)
Calls the child commands handle method

5.46.1 Detailed Description

The entry point for command execution with a single child.

5.46.2 Constructor & Destructor Documentation

5.46.2.1 RootASTNode()

```
MB2D.RootASTNode.RootASTNode (
    MBCConsoleASTNode child ) [inline]
```

Initializes a new instance of the T:MB2D.RootASTNode class.

Parameters

<i>child</i>	Command AST node to handle.
--------------	-----------------------------

5.46.3 Member Function Documentation

5.46.3.1 Handle()

```
override void MB2D.RootASTNode.Handle (
    MBCConsole console ) [inline], [virtual]
```

Calls the child commands handle method

Parameters

<i>console</i>	Console to handle.
----------------	--------------------

Implements [MB2D.MBCConsoleASTNode](#).

Here is the call graph for this function:



Here is the caller graph for this function:



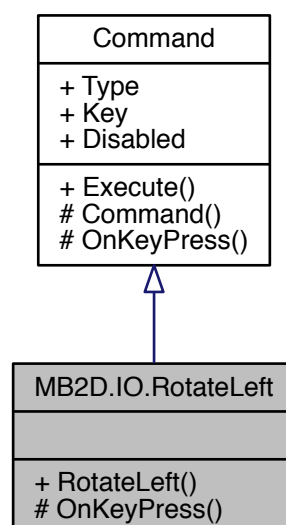
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsoleAST.cs

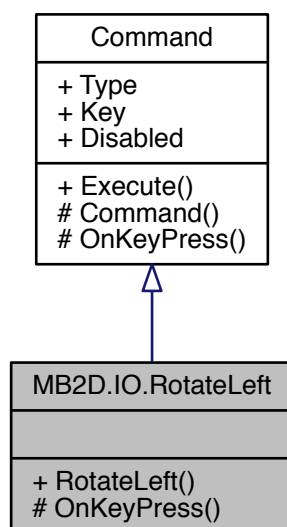
5.47 MB2D.IO.RotateLeft Class Reference

Rotates an entity left

Inheritance diagram for MB2D.IO.RotateLeft:



Collaboration diagram for MB2D.IO.RotateLeft:



Public Member Functions

- [RotateLeft](#) (Keys key, [CommandType](#) type)
Initializes a new instance of the T:MB2D.IO.RotateLeft class.

Protected Member Functions

- override void [OnKeyPress](#) ([Entity](#) e=null)
Rotates an entity left

Additional Inherited Members

5.47.1 Detailed Description

Rotates an entity left

5.47.2 Constructor & Destructor Documentation

5.47.2.1 RotateLeft()

```

MB2D.IO.RotateLeft.RotateLeft (
    Keys key,
    CommandType type ) [inline]
  
```

Initializes a new instance of the T:MB2D.IO.RotateLeft class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.47.3 Member Function Documentation

5.47.3.1 OnKeyPress()

```
override void MB2D.IO.RotateLeft.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Rotates an entity left

Parameters

<i>e</i>	Entity to rotate.
----------	-------------------

Implements [MB2D.IO.Command](#).

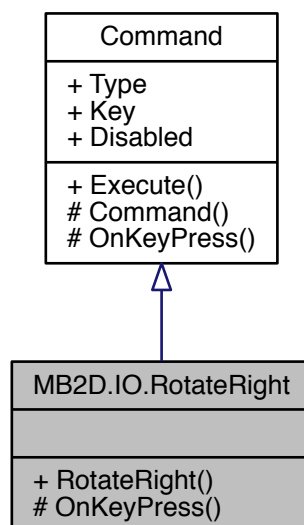
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

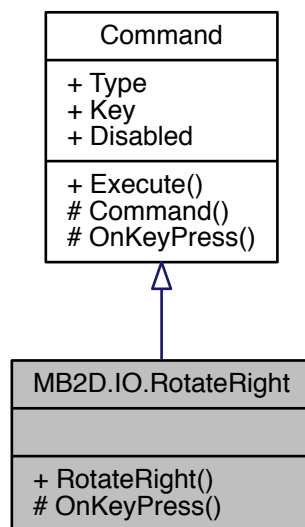
5.48 MB2D.IO.RotateRight Class Reference

Rotates an entity right

Inheritance diagram for MB2D.IO.RotateRight:



Collaboration diagram for MB2D.IO.RotateRight:



Public Member Functions

- [RotateRight](#) (Keys key, [CommandType](#) type)
Initializes a new instance of the T:MB2D.IO.RotateRight class.

Protected Member Functions

- override void [OnKeyPress](#) (Entity e=null)
Rotates an entity right

Additional Inherited Members

5.48.1 Detailed Description

Rotates an entity right

5.48.2 Constructor & Destructor Documentation

5.48.2.1 RotateRight()

```

MB2D.IO.RotateRight.RotateRight (
    Keys key,
    CommandType type ) [inline]
  
```

Initializes a new instance of the T:MB2D.IO.RotateRight class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.48.3 Member Function Documentation

5.48.3.1 OnKeyPress()

```
override void MB2D.IO.RotateRight.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Rotates an entity right

Parameters

<i>e</i>	Entity to rotate.
----------	-------------------

Implements [MB2D.IO.Command](#).

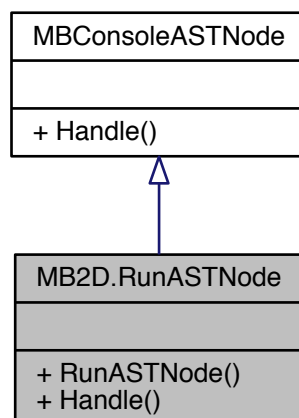
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

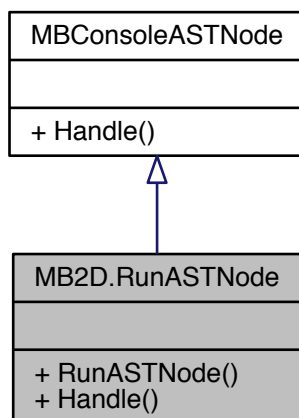
5.49 MB2D.RunASTNode Class Reference

AST node entry point for executing a run command

Inheritance diagram for MB2D.RunASTNode:



Collaboration diagram for MB2D.RunASTNode:



Public Member Functions

- [RunASTNode](#) (string ident, params string[] args)
Initializes a new instance of the T:MB2D.RunASTNode class.
- override void [Handle](#) (MBConsole console)
Checks for valid identifier and executes the given function from the console if correctly defined.

5.49.1 Detailed Description

AST node entry point for executing a run command

5.49.2 Constructor & Destructor Documentation

5.49.2.1 RunASTNode()

```

MB2D.RunASTNode.RunASTNode (
    string ident,
    params string [] args ) [inline]
  
```

Initializes a new instance of the T:MB2D.RunASTNode class.

Parameters

<i>ident</i>	Identifier of the function.
<i>args</i>	Arguments to pass to the function.

5.49.3 Member Function Documentation

5.49.3.1 Handle()

```
override void MB2D.RunASTNode.Handle (  
    MBConsole console ) [inline], [virtual]
```

Checks for valid identifier and executes the given function from the console if correctly defined.

Parameters

<i>console</i>	Console to handle.
----------------	--------------------

Implements [MB2D.MBConsoleASTNode](#).

Here is the call graph for this function:



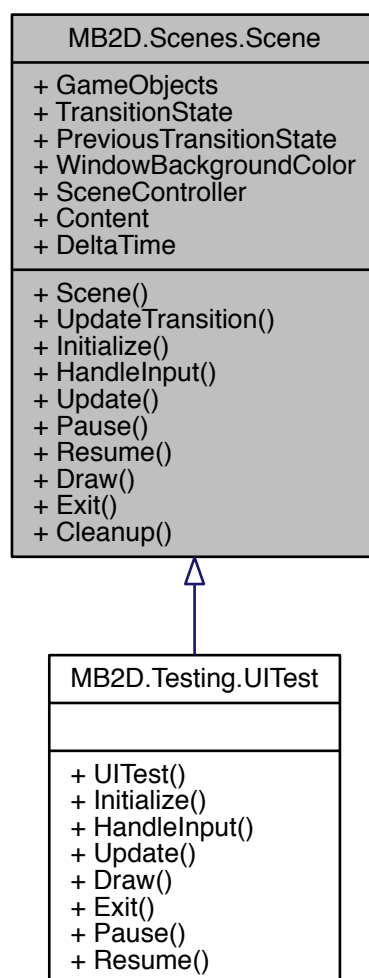
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsoleAST.cs

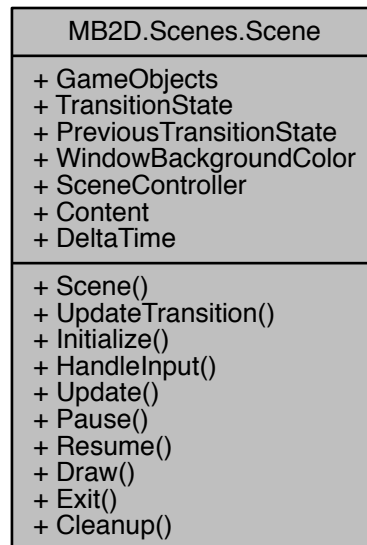
5.50 MB2D.Scenes.Scene Class Reference

Holds all logic and data for a single game screen

Inheritance diagram for MB2D.Scenes.Scene:



Collaboration diagram for MB2D.Scenes.Scene:



Public Member Functions

- [Scene](#) ([EntityMap](#) gameObjects, [ContentManager](#) content)
Initializes a new instance of the T:MB2D.Scenes.Scene class with a pre-existing EntityMap
- void **UpdateTransition** ()
- abstract void [Initialize](#) ()
Initialize this scene and loads all resources. Runs logic to execute during the Initializing state. Set state to None to end.
- abstract void [HandleInput](#) ()
Handles all input for the scene
- abstract void [Update](#) ()
Updates game logic and changes scene state.
- abstract void [Pause](#) ()
Runs logic to execute while the scene is in the Pausing state. Set state to None to end.
- abstract void [Resume](#) ()
Runs logic to execute while the scene is in the Resuming state. Set state to None to end.
- abstract void [Draw](#) ([SpriteBatch](#) spriteBatch, [SpriteBatch](#) uiSpriteBatch)
Draws entities and [UI](#) elements to the specified SpriteBatches
- abstract void [Exit](#) ()
Runs logic to execute while the scene is in the Exiting state. Set state to Null to end.
- void [Cleanup](#) ()
Cleans up the scene and unloads content.

Properties

- [EntityMap GameObjects](#) [get]
Gets all entities allocated to the scene
- [TransitionState TransitionState](#) [get, set]
Gets or sets the current transition state of the scene. This causes the scene stack to change the scenes state on the next frame.
- [TransitionState PreviousTransitionState](#) [get]
Gets the state the scene was in during the last frame.
- Color [WindowBackgroundColor](#) [get, set]
Gets or sets the color of the window background for this scene.
- [SceneStack SceneController](#) [get, set]
Gets or sets the scene controller.
- ContentManager [Content](#) [get]
Gets the content manager for loading and unloading resources.
- float [DeltaTime](#) [get, set]
Gets or sets the delta time value.

5.50.1 Detailed Description

Holds all logic and data for a single game screen

5.50.2 Constructor & Destructor Documentation

5.50.2.1 Scene()

```
MB2D.Scenes.Scene.Scene (
    EntityMap gameObject,
    ContentManager content ) [inline]
```

Initializes a new instance of the T:MB2D.Scenes.Scene class with a pre-existing EntityMap

Parameters

<i>gameObject</i>	EntityMap to assign to the scene.
-------------------	-----------------------------------

Here is the call graph for this function:



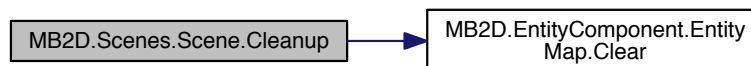
5.50.3 Member Function Documentation

5.50.3.1 Cleanup()

```
void MB2D.Scenes.Scene.Cleanup ( ) [inline]
```

Cleans up the scene and unloads content.

Here is the call graph for this function:



5.50.3.2 Draw()

```
abstract void MB2D.Scenes.Scene.Draw (
    SpriteBatch spriteBatch,
    SpriteBatch uiSpriteBatch ) [pure virtual]
```

Draws entities and [UI](#) elements to the specified SpriteBatches

Parameters

<i>spriteBatch</i>	World-coordinate based sprite batch.
<i>uiSpriteBatch</i>	Camera-based User Interface sprite batch.

Implemented in [MB2D.Testing.UITest](#).

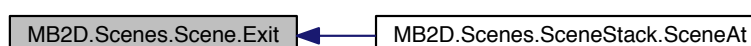
5.50.3.3 Exit()

```
abstract void MB2D.Scenes.Scene.Exit ( ) [pure virtual]
```

Runs logic to execute while the scene is in the Exiting state. Set state to Null to end.

Implemented in [MB2D.Testing.UITest](#).

Here is the caller graph for this function:



5.50.3.4 HandleInput()

```
abstract void MB2D.Scenes.Scene.HandleInput ( ) [pure virtual]
```

Handles all input for the scene

Implemented in [MB2D.Testing.UITest](#).

5.50.3.5 Initialize()

```
abstract void MB2D.Scenes.Scene.Initialize ( ) [pure virtual]
```

Initialize this scene and loads all resources. Runs logic to execute during the Initializing state. Set state to None to end.

Implemented in [MB2D.Testing.UITest](#).

Here is the caller graph for this function:



5.50.3.6 Pause()

```
abstract void MB2D.Scenes.Scene.Pause ( ) [pure virtual]
```

Runs logic to execute while the scene is in the Pausing state. Set state to None to end.

Implemented in [MB2D.Testing.UITest](#).

Here is the caller graph for this function:



5.50.3.7 Resume()

```
abstract void MB2D.Scenes.Scene.Resume ( ) [pure virtual]
```

Runs logic to execute while the scene is in the Resuming state. Set state to None to end.

Implemented in [MB2D.Testing.UITest](#).

Here is the caller graph for this function:



5.50.3.8 Update()

```
abstract void MB2D.Scenes.Scene.Update ( ) [pure virtual]
```

Updates game logic and changes scene state.

Implemented in [MB2D.Testing.UITest](#).

5.50.4 Property Documentation

5.50.4.1 Content

```
ContentManager MB2D.Scenes.Scene.Content [get], [protected]
```

Gets the content manager for loading and unloading resources.

The content manager.

5.50.4.2 DeltaTime

```
float MB2D.Scenes.Scene.DeltaTime [get], [set]
```

Gets or sets the delta time value.

The delta time.

5.50.4.3 GameObjects

```
EntityMap MB2D.Scenes.Scene.GameObjects [get]
```

Gets all entities allocated to the scene

The game objects.

5.50.4.4 PreviousTransitionState

`TransitionState` MB2D.Scenes.Scene.PreviousTransitionState [get]

Gets the state the scene was in during the last frame.

The state of the previous transition.

5.50.4.5 SceneController

`SceneStack` MB2D.Scenes.Scene.SceneController [get], [set]

Gets or sets the scene controller.

The scene controller.

5.50.4.6 TransitionState

`TransitionState` MB2D.Scenes.Scene.TransitionState [get], [set]

Gets or sets the current transition state of the scene. This causes the scene stack to change the scenes state on the next frame.

The transition state.

5.50.4.7 WindowBackgroundColor

`Color` MB2D.Scenes.Scene.WindowBackgroundColor [get], [set]

Gets or sets the color of the window background for this scene.

The color of the window background.

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

- MB2D/src/Scene/Scene.cs

5.51 MB2D.Scenes.SceneStack Class Reference

Holds the games scenes in a stack structure running the top scene every frame. Handles switching state for scenes and popping/pushing new scenes on top of one another. Allows the current scene to access other scenes.

Collaboration diagram for MB2D.Scenes.SceneStack:

MB2D.Scenes.SceneStack
+ Top + Bottom + Size + LastIndex + Next + LastSceneType
+ SceneStack() + Push() + Draw() + Update() + Pop() + ResetTo() + SceneAt()

Public Member Functions

- [SceneStack](#) ()
Initializes a new instance of the T:MB2D.Scenes.SceneStack class.
- void [Push](#) ([Scene](#) scene)
Pushes a new scene to the top of the stack. Calls the new scenes Initialize method and the previous scenes Pause method
- void **Draw** (SpriteBatch spriteBatch, SpriteBatch uiSpriteBatch)
- void [Update](#) ()
Updates the scene at the top of the stack and handles any state transitions if they've been called. For any transitions or scene logic to function correctly this must be called once per frame.
- void [Pop](#) ()
Pops the top scene off the stack, calling its Exit method and calls the Resume method of the next scene on the stack if it exists.
- void [ResetTo](#) ([Scene](#) scene)
Resets the scene stack to the specified scene, clearing all other scenes from the stack. Use this in most scenarios instead of Push to save memory by not keeping scenes allocated if unnecessary.
- [Scene](#) [SceneAt](#) (int index)
Gets the scene located at the specific index in the stack

Properties

- [Scene Top](#) [get]
Gets the scene at the top of the stack.
- [Scene Bottom](#) [get]
Gets the scene at the bottom of the stack.
- int [Size](#) [get]
Gets the current size of the stack
- int [LastIndex](#) [get]
Gets the upper bounds of the indexes of the stack
- [Scene Next](#) [get]
- Type [LastSceneType](#) [get]

5.51.1 Detailed Description

Holds the games scenes in a stack structure running the top scene every frame. Handles switching state for scenes and popping/pushing new scenes on top of one another. Allows the current scene to access other scenes.

5.51.2 Constructor & Destructor Documentation

5.51.2.1 SceneStack()

```
MB2D.Scenes.SceneStack.SceneStack ( ) [inline]
```

Initializes a new instance of the T:MB2D.Scenes.SceneStack class.

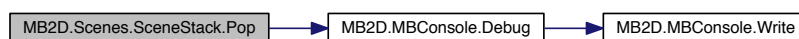
5.51.3 Member Function Documentation

5.51.3.1 Pop()

```
void MB2D.Scenes.SceneStack.Pop ( ) [inline]
```

Pops the top scene off the stack, calling its Exit method and calls the Resume method of the next scene on the stack if it exists.

Here is the call graph for this function:



Here is the caller graph for this function:



5.51.3.2 Push()

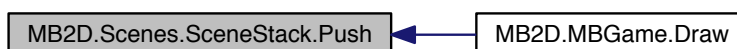
```
void MB2D.Scenes.SceneStack.Push (
    Scene scene ) [inline]
```

Pushes a new scene to the top of the stack. Calls the new scenes Initialize method and the previous scenes Pause method

Parameters

<i>scene</i>	Scene to push.
--------------	----------------

Here is the caller graph for this function:



5.51.3.3 ResetTo()

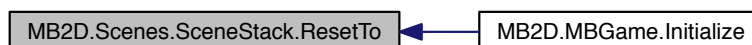
```
void MB2D.Scenes.SceneStack.ResetTo (  
    Scene scene ) [inline]
```

Resets the scene stack to the specified scene, clearing all other scenes from the stack. Use this in most scenarios instead of Push to save memory by not keeping scenes allocated if unnecessary.

Parameters

<i>scene</i>	Scene to reset to.
--------------	--------------------

Here is the caller graph for this function:



5.51.3.4 SceneAt()

```
Scene MB2D.Scenes.SceneStack.SceneAt (  
    int index ) [inline]
```

Gets the scene located at the specific index in the stack

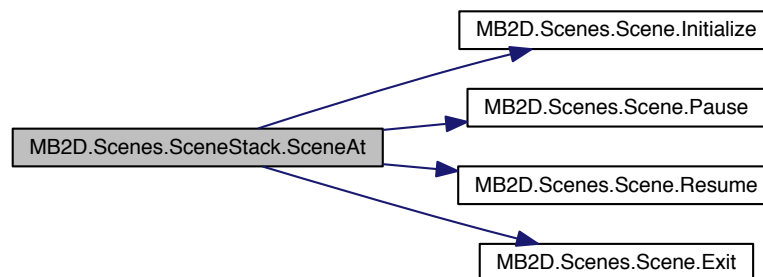
Returns

The T:MB2D.Scenes.Scene.

Parameters

<i>index</i>	Index to get.
--------------	---------------

Here is the call graph for this function:

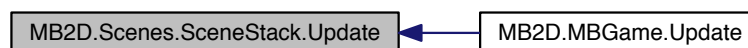


5.51.3.5 Update()

```
void MB2D.Scenes.SceneStack.Update ( ) [inline]
```

Updates the scene at the top of the stack and handles any state transitions if they've been called. For any transitions or scene logic to function correctly this must be called once per frame.

Here is the caller graph for this function:



5.51.4 Property Documentation

5.51.4.1 Bottom

```
Scene MB2D.Scenes.SceneStack.Bottom [get]
```

Gets the scene at the bottom of the stack.

The bottom scene.

5.51.4.2 LastIndex

```
int MB2D.Scenes.SceneStack.LastIndex [get]
```

Gets the upper bounds of the indexes of the stack

The last index.

5.51.4.3 Size

```
int MB2D.Scenes.SceneStack.Size [get]
```

Gets the current size of the stack

The size.

5.51.4.4 Top

```
Scene MB2D.Scenes.SceneStack.Top [get]
```

Gets the scene at the top of the stack.

The scene at the top of the stack.

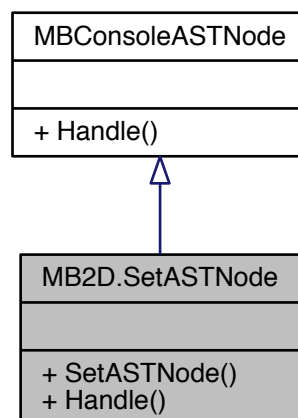
The documentation for this class was generated from the following file:

- MB2D/src/Scene/SceneStack.cs

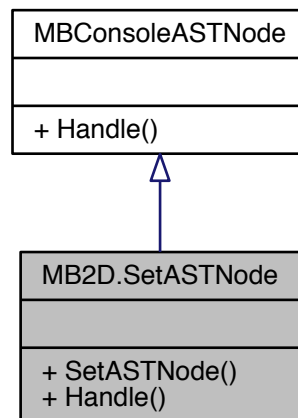
5.52 MB2D.SetASTNode Class Reference

AST node representing the entry point for a 'set' command with an identifier and a value child

Inheritance diagram for MB2D.SetASTNode:



Collaboration diagram for MB2D.SetASTNode:



Public Member Functions

- [SetASTNode](#) (string ident, [VariableASTNode](#) val)
Initializes a new instance of the T:MB2D.SetASTNode class.
- override void [Handle](#) ([MBConsole](#) console)
Handles setting the variable using the consoles Vars property. Checks if the identifier only starts with an alpha or underscore and handles any type checking or parse errors.

5.52.1 Detailed Description

AST node representing the entry point for a 'set' command with an identifier and a value child

5.52.2 Constructor & Destructor Documentation

5.52.2.1 SetASTNode()

```

MB2D.SetASTNode.SetASTNode (
    string ident,
    VariableASTNode val ) [inline]
  
```

Initializes a new instance of the T:MB2D.SetASTNode class.

Parameters

<i>ident</i>	Identifier of the variable.
<i>val</i>	Value to assign.

5.52.3 Member Function Documentation

5.52.3.1 Handle()

```
override void MB2D.SetASTNode.Handle (
    MBConsole console ) [inline], [virtual]
```

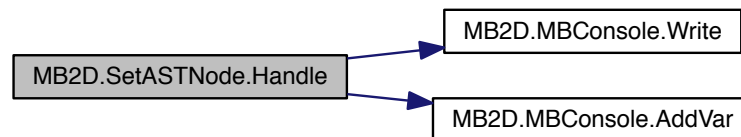
Handles setting the variable using the consoles Vars property. Checks if the identifier only starts with an alpha or underscore and handles any type checking or parse errors.

Parameters

<i>console</i>	Console to handle.
----------------	--------------------

Implements [MB2D.MBConsoleASTNode](#).

Here is the call graph for this function:



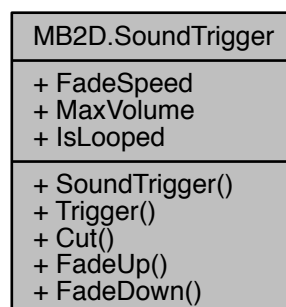
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsoleAST.cs

5.53 MB2D.SoundTrigger Class Reference

Triggers a sound effect

Collaboration diagram for MB2D.SoundTrigger:



Public Member Functions

- [SoundTrigger](#) (SoundEffect sound)
Initializes a new instance of the T:MB2D.SoundTrigger class using the specified SoundEffect
- void [Trigger](#) ()
Plays the sound if it's not already playing
- void [Cut](#) ()
Stops the sound if it's playing
- void [FadeUp](#) ()
Increases the sounds volume one step based on the specified FadeSpeed
- void [FadeDown](#) ()
Decreases the volume based on the specified FadeSpeed. Stops the sound once the volume reaches 0

Properties

- float [FadeSpeed](#) [get, set]
Gets or sets the fade speed.
- float [MaxVolume](#) [get, set]
Determines the maximum volume to FadeUp
- bool [IsLooped](#) [get, set]
Gets or sets a value indicating whether this T:MB2D.SoundTrigger is looped or one-shot.

5.53.1 Detailed Description

Triggers a sound effect

5.53.2 Constructor & Destructor Documentation

5.53.2.1 SoundTrigger()

```
MB2D.SoundTrigger.SoundTrigger (
    SoundEffect sound ) [inline]
```

Initializes a new instance of the T:MB2D.SoundTrigger class using the specified SoundEffect

Parameters

<i>sound</i>	Sound to use.
--------------	---------------

5.53.3 Member Function Documentation

5.53.3.1 Cut()

```
void MB2D.SoundTrigger.Cut ( ) [inline]
```

Stops the sound if it's playing

5.53.3.2 FadeDown()

```
void MB2D.SoundTrigger.FadeDown ( ) [inline]
```

Decreases the volume based on the specified FadeSpeed. Stops the sound once the volume reaches 0

5.53.3.3 FadeUp()

```
void MB2D.SoundTrigger.FadeUp ( ) [inline]
```

Increases the sounds volume one step based on the specified FadeSpeed

5.53.3.4 Trigger()

```
void MB2D.SoundTrigger.Trigger ( ) [inline]
```

Plays the sound if it's not already playing

5.53.4 Property Documentation

5.53.4.1 FadeSpeed

```
float MB2D.SoundTrigger.FadeSpeed [get], [set]
```

Gets or sets the fade speed.

The fade speed.

5.53.4.2 IsLooped

```
bool MB2D.SoundTrigger.IsLooped [get], [set]
```

Gets or sets a value indicating whether this T:MB2D.SoundTrigger is looped or one-shot.

true if is looped; otherwise, false.

5.53.4.3 MaxVolume

```
float MB2D.SoundTrigger.MaxVolume [get], [set]
```

Determines the maximum volume to FadeUp

The max volume.

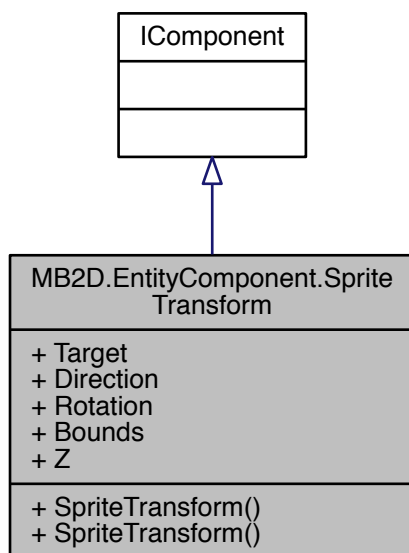
The documentation for this class was generated from the following file:

- MB2D/src/Audio/SoundTrigger.cs

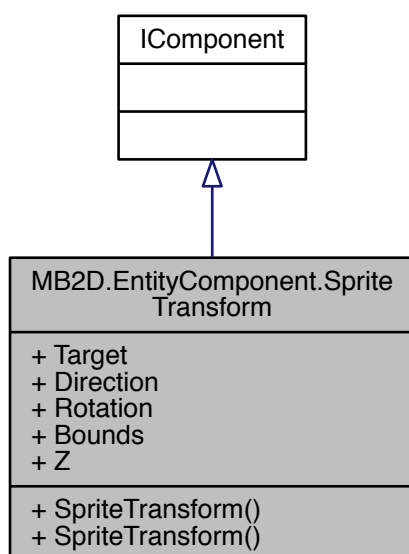
5.54 MB2D.EntityComponent.SpriteTransform Class Reference

Defines a sprite component with control over its size, rotation, and scale

Inheritance diagram for MB2D.EntityComponent.SpriteTransform:



Collaboration diagram for MB2D.EntityComponent.SpriteTransform:



Public Member Functions

- [SpriteTransform](#) (Texture2D texture, Vector2 position, Vector2 scale)
Initializes a new instance of the T:MB2D.EntityComponent.SpriteTransform class.
- [SpriteTransform](#) (TextureRegion2D texture, Vector2 position, Vector2 scale)
Initializes a new instance of the T:MB2D.EntityComponent.SpriteTransform class.

Properties

- [Sprite Target](#) [get, set]
Gets or sets the sprites target containing all of the applied data and logicd.
- [Vector2 Direction](#) [get]
Gets the sprites direction.
- [float Rotation](#) [get, set]
Gets or sets the rotation in radians.
- [RectangleF Bounds](#) [get, set]
Gets or sets the sprites bounding box.
- [float Z](#) [get, set]
Gets or sets the sprites z index. Used in depth systems.

5.54.1 Detailed Description

Defines a sprite component with control over its size, rotation, and scale

5.54.2 Constructor & Destructor Documentation

5.54.2.1 [SpriteTransform\(\)](#) [1/2]

```
MB2D.EntityComponent.SpriteTransform.SpriteTransform (
    Texture2D texture,
    Vector2 position,
    Vector2 scale ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.SpriteTransform class.

Parameters

<i>texture</i>	Texture to assign to the sprite.
<i>position</i>	Initial position of the sprite. Should be the entities position for best practice.
<i>scale</i>	Initial scale of the sprite.

5.54.2.2 [SpriteTransform\(\)](#) [2/2]

```
MB2D.EntityComponent.SpriteTransform.SpriteTransform (
    TextureRegion2D texture,
    Vector2 position,
    Vector2 scale ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.SpriteTransform class.

Parameters

<i>texture</i>	Texture region in a texture atlas to assign to the sprite.
<i>position</i>	Initial position of the sprite. Should be the entities position for best practice.
<i>scale</i>	Initial scale of the sprite.

5.54.3 Property Documentation**5.54.3.1 Bounds**

`RectangleF MB2D.EntityComponent.SpriteTransform.Bounds [get], [set]`

Gets or sets the sprites bounding box.

The bounds.

5.54.3.2 Direction

`Vector2 MB2D.EntityComponent.SpriteTransform.Direction [get]`

Gets the sprites direction.

The direction.

5.54.3.3 Rotation

`float MB2D.EntityComponent.SpriteTransform.Rotation [get], [set]`

Gets or sets the rotation in radians.

The rotation in radians.

5.54.3.4 Target

`Sprite MB2D.EntityComponent.SpriteTransform.Target [get], [set]`

Gets or sets the sprites target containing all of the applied data and logicd.

The sprite target.

5.54.3.5 Z

`float MB2D.EntityComponent.SpriteTransform.Z [get], [set]`

Gets or sets the sprites z index. Used in depth systems.

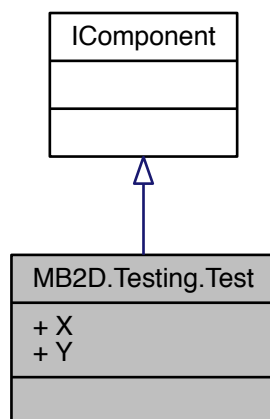
The z index.

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

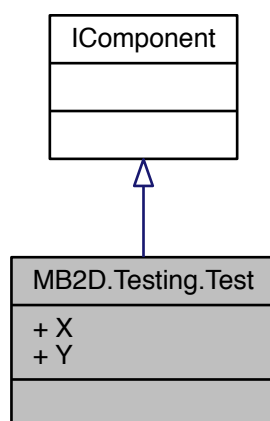
- MB2D/src/EntityComponent/Components/SpriteTransform.cs

5.55 MB2D.Testing.Test Class Reference

Inheritance diagram for MB2D.Testing.Test:



Collaboration diagram for MB2D.Testing.Test:



Properties

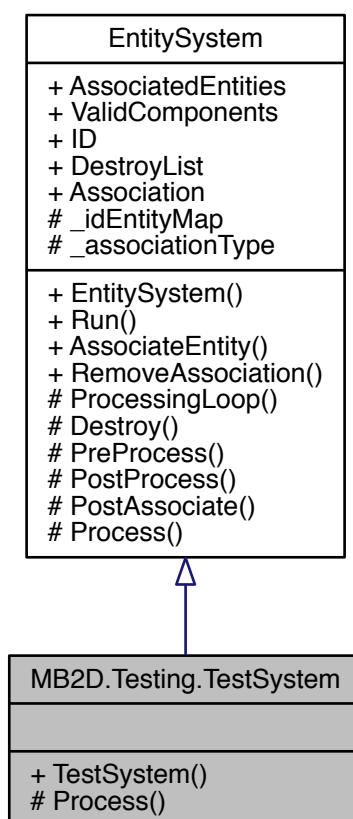
- **int X** [get, set]
- **int Y** [get, set]

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

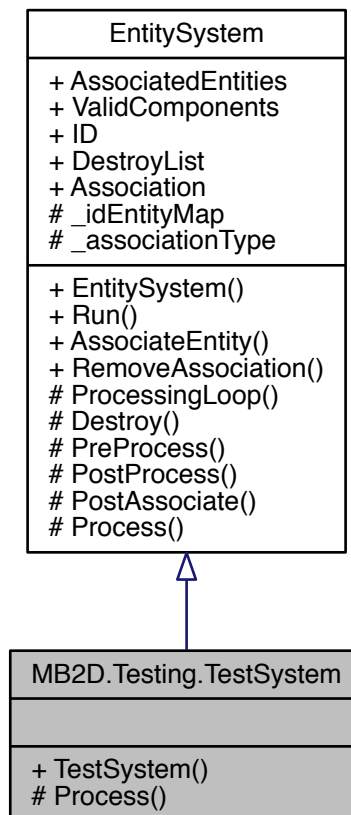
- MB2D/src/Test/TestSystem.cs

5.56 MB2D.Testing.TestSystem Class Reference

Inheritance diagram for MB2D.Testing.TestSystem:



Collaboration diagram for MB2D.Testing.TestSystem:



Protected Member Functions

- override void `Process` (`Entity` entity)
Executes this systems logic on a single entity

Additional Inherited Members

5.56.1 Member Function Documentation

5.56.1.1 Process()

```

override void MB2D.Testing.TestSystem.Process (
    Entity entity ) [inline], [protected], [virtual]
  
```

Executes this systems logic on a single entity

Parameters

<i>entity</i>	Entity to operate on
---------------	----------------------

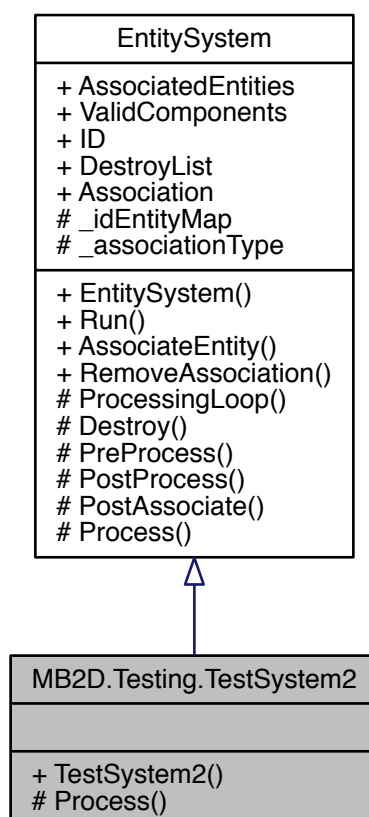
Implements [MB2D.EntityComponent.EntitySystem](#).

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

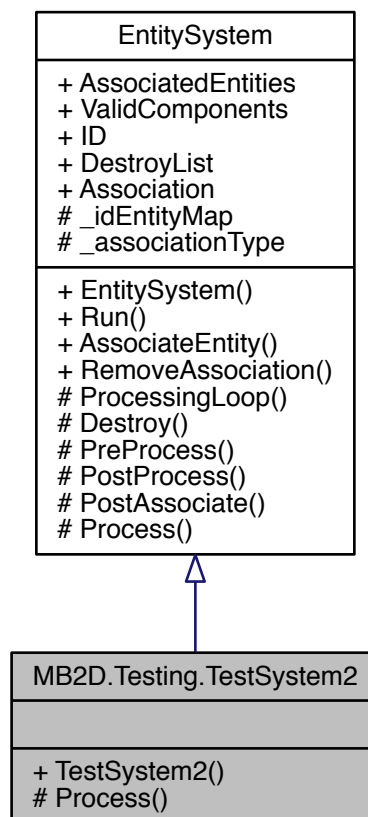
- MB2D/src/Test/TestSystem.cs

5.57 MB2D.Testing.TestSystem2 Class Reference

Inheritance diagram for MB2D.Testing.TestSystem2:



Collaboration diagram for MB2D.Testing.TestSystem2:



Protected Member Functions

- override void `Process (Entity entity)`
Executes this systems logic on a single entity

Additional Inherited Members

5.57.1 Member Function Documentation

5.57.1.1 Process()

```

override void MB2D.Testing.TestSystem2.Process (
    Entity entity ) [inline], [protected], [virtual]
  
```

Executes this systems logic on a single entity

Parameters

<i>entity</i>	Entity to operate on
---------------	----------------------

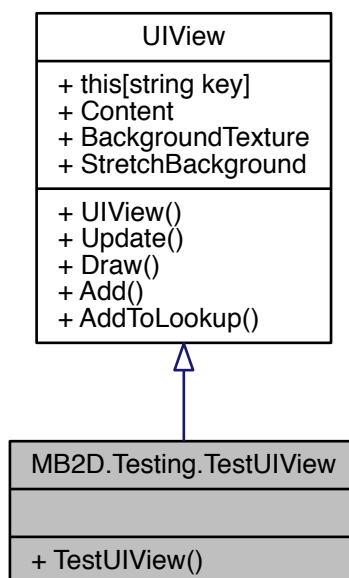
Implements [MB2D.EntityComponent.EntitySystem](#).

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

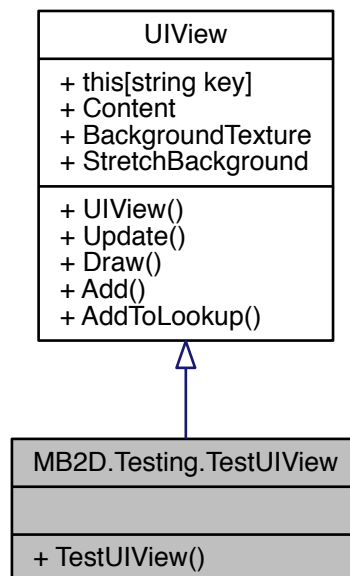
- MB2D/src/Test/TestSystem.cs

5.58 MB2D.Testing.TestUIView Class Reference

Inheritance diagram for MB2D.Testing.TestUIView:



Collaboration diagram for MB2D.Testing.TestUIView:



Public Member Functions

- **TestUIView** (ContentManager content)

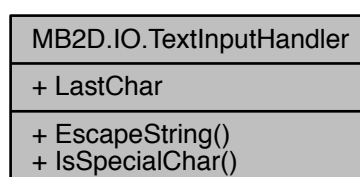
Additional Inherited Members

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

- MB2D/src/Test/TestUIView.cs

5.59 MB2D.IO.TextInputHandler Class Reference

Collaboration diagram for MB2D.IO.TextInputHandler:



Public Member Functions

- string **EscapeString** (char c)
- bool **IsSpecialChar** (char c)

Properties

- char **LastChar** [get]

Returns the last character entered by the user. Translates the keycode to an ASCII character taking into account space, backspace, and special character modifiers.

5.59.1 Property Documentation

5.59.1.1 LastChar

```
char MB2D.IO.TextInputHandler.LastChar [get]
```

Returns the last character entered by the user. Translates the keycode to an ASCII character taking into account space, backspace, and special character modifiers.

Returns

The char.

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

- MB2D/src/Input/TextInputHandler.cs

5.60 MB2D.Tile Class Reference

Represents a single tile in a tile map.

Collaboration diagram for MB2D.Tile:



Public Member Functions

- [Tile](#) (int textureID, Color color)
Initializes a new instance of the T:MidnightBlue.Tile class
- [Tile](#) ()
Initializes a new instance of the T:MidnightBlue.Tile class.

Properties

- int [ID](#) [get, protected set]
Gets or sets the tile map region ID to use for this tile.
- Color [TintColor](#) [get, protected set]
Gets or sets the color of the tint.
- [TileFlag](#) [Flag](#) [get, set]
Gets or sets the tile flag for collision detection.

5.60.1 Detailed Description

Represents a single tile in a tile map.

5.60.2 Constructor & Destructor Documentation

5.60.2.1 [Tile\(\)](#) [1/2]

```
MB2D.Tile.Tile (
    int textureID,
    Color color ) [inline]
```

Initializes a new instance of the T:MidnightBlue.Tile class

Parameters

<i>textureID</i>	ID of the texture region in the tile map to use for this tile, i.e. Grass or water.
<i>color</i>	Color tint to apply to the tile.

5.60.2.2 [Tile\(\)](#) [2/2]

```
MB2D.Tile.Tile ( ) [inline]
```

Initializes a new instance of the T:MidnightBlue.Tile class.

5.60.3 Property Documentation

5.60.3.1 [Flag](#)

[TileFlag](#) MB2D.Tile.Flag [get], [set]

Gets or sets the tile flag for collision detection.

The flag.

5.60.3.2 ID

```
int MB2D.Tile.ID [get], [protected set]
```

Gets or sets the tile map region ID to use for this tile.

The tile map region ID.

5.60.3.3 TintColor

```
Color MB2D.Tile.TintColor [get], [protected set]
```

Gets or sets the color of the tint.

The color of the tint.

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

- MB2D/src/Tiles/Tile.cs

5.61 MB2D.Tiles.TileMap Class Reference

A grid of tiles with collision. Wraps coordinates when they fall out of bounds. Allows accessing tiles by index.

Collaboration diagram for MB2D.Tiles.TileMap:

MB2D.Tiles.TileMap
+ this[int x, int y] + Texture + TileSize + MapSize
+ Fill() + TileMap() + GetTile() + Draw() + HandleWrapping()

Public Member Functions

- void [Fill](#) ([Tile](#)[,] tiles)
Uses a 2D Array of previously defined tile information to fill a tile map with collision data and other information. Must be called in order for the [TileMap](#) to function.
- [TileMap](#) (Texture2D texture, int cellSize, int margin=0, int spacing=0, int offset=0, float scale=1.0f)
Initializes a new instance of the T:MB2D.Tiles.TileMap class. Seperates the texture into a series of regions.
- Rectangle [GetTile](#) (int id)
Retrieves the bounding rectangle of a tile texture from the tilemap
- void [Draw](#) (SpriteBatch spriteBatch)
Draws the tile map to the specified SpriteBatch, wrapping the rendering when the camera reaches the bounds of the map.
- void [HandleWrapping](#) ([Movement](#) movement)
Handles wrapping an entity around the map if their movement falls out of bounds - gives the illusion of an infinitely looping map.

Properties

- [Tile](#) `this[int x, int y]` [get]
Gets the T:MB2D.Tiles.Tile at the specified x y.
- `Texture2D` [Texture](#) [get]
Gets the texture atlases undivided texture.
- `Point` [TileSize](#) [get]
Gets the size of each tile in the world.
- `Point` [MapSize](#) [get]
Gets the size of the map.

5.61.1 Detailed Description

A grid of tiles with collision. Wraps coordinates when they fall out of bounds. Allows accessing tiles by index.

5.61.2 Constructor & Destructor Documentation

5.61.2.1 TileMap()

```
MB2D.Tiles.TileMap.TileMap (
    Texture2D texture,
    int cellSize,
    int margin = 0,
    int spacing = 0,
    int offset = 0,
    float scale = 1.0f ) [inline]
```

Initializes a new instance of the T:MB2D.Tiles.TileMap class. Seperates the texture into a series of regions.

Parameters

<i>texture</i>	Texture to use in the texture atlas.
<i>cellSize</i>	The size of each cell in the texture atlas.
<i>margin</i>	Margin to apply to each rendered tile.
<i>spacing</i>	Spacing to apply to each rendered tile.
<i>offset</i>	Offset to apply to the x and y coordinates of each tile when rendering.
<i>scale</i>	Scale vector to apply to each cell when rendering.

5.61.3 Member Function Documentation

5.61.3.1 Draw()

```
void MB2D.Tiles.TileMap.Draw (
    SpriteBatch spriteBatch ) [inline]
```

Draws the tile map to the specified SpriteBatch, wrapping the rendering when the camera reaches the bounds of the map.

Parameters

<i>spriteBatch</i>	Sprite batch to draw to.
--------------------	--------------------------

5.61.3.2 Fill()

```
void MB2D.Tiles.TileMap.Fill (
    Tile tiles[,] ) [inline]
```

Uses a 2D Array of previously defined tile information to fill a tile map with collision data and other information. Must be called in order for the [TileMap](#) to function.

Parameters

<i>tiles</i>	Tiles.
--------------	------------------------

5.61.3.3 GetTile()

```
Rectangle MB2D.Tiles.TileMap.GetTile (
    int id ) [inline]
```

Retrieves the bounding rectangle of a tile texture from the tilemap

Returns

The tile ID's bounding rectangle.

Parameters

<i>id</i>	The ID to get.
-----------	----------------

5.61.3.4 HandleWrapping()

```
void MB2D.Tiles.TileMap.HandleWrapping (
    Movement movement ) [inline]
```

Handles wrapping an entity around the map if their movement falls out of bounds - gives the illusion of an infinitely looping map.

Parameters

<i>movement</i>	Movement component to operate on.
-----------------	-----------------------------------

5.61.4 Property Documentation**5.61.4.1 MapSize**

```
Point MB2D.Tiles.TileMap.MapSize [get]
```

Gets the size of the map.

The size of the map.

5.61.4.2 Texture

`Texture2D MB2D.Tiles.TileMap.Texture [get]`

Gets the texture atlases undivided texture.

The texture.

5.61.4.3 this[int x, int y]

`Tile MB2D.Tiles.TileMap.this[int x, int y] [get]`

Gets the T:MB2D.Tiles.Tile at the specified x y.

Parameters

<i>x</i>	The x coordinate.
<i>y</i>	The y coordinate.

5.61.4.4 TileSize

`Point MB2D.Tiles.TileMap.TileSize [get]`

Gets the size of each tile in the world.

The size of the tile.

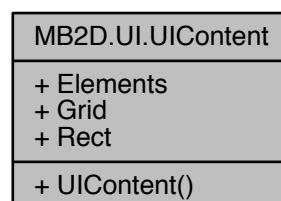
The documentation for this class was generated from the following file:

- MB2D/src/Tiles/TileMap.cs

5.62 MB2D.UI.UIContent Class Reference

Holds content in a grid structure for a UIContext or [Layout](#)

Collaboration diagram for MB2D.UI.UIContent:



Public Member Functions

- [UIContent](#) (int rows, int cols, Rectangle parent)
Initializes a new instance of the T:MB2D.UI.UIContent class.

Properties

- [UIElement](#) [,] [Elements](#) [get]
Gets the elements of the content.
- [Grid](#) [Grid](#) [get]
Gets a grid geometry representation of the content
- Rectangle [Rect](#) [get, set]
Gets or sets the rectangle encompassing the content.

5.62.1 Detailed Description

Holds content in a grid structure for a UIContext or [Layout](#)

5.62.2 Constructor & Destructor Documentation

5.62.2.1 UIContent()

```
MB2D.UI.UIContent.UIContent (
    int rows,
    int cols,
    Rectangle parent ) [inline]
```

Initializes a new instance of the T:MB2D.UI.UIContent class.

Parameters

<i>rows</i>	Rows.
<i>cols</i>	Cols.
<i>parent</i>	Parent.

5.62.3 Property Documentation

5.62.3.1 Elements

```
UIElement [,] MB2D.UI.UIContent.Elements [get]
```

Gets the elements of the content.

The [UI](#) elements.

5.62.3.2 Grid

`Grid` MB2D.UI.UIContent.Grid [get]

Gets a grid geometry representation of the content

The grid.

5.62.3.3 Rect

`Rectangle` MB2D.UI.UIContent.Rect [get], [set]

Gets or sets the rectangle encompassing the content.

The rectangle.

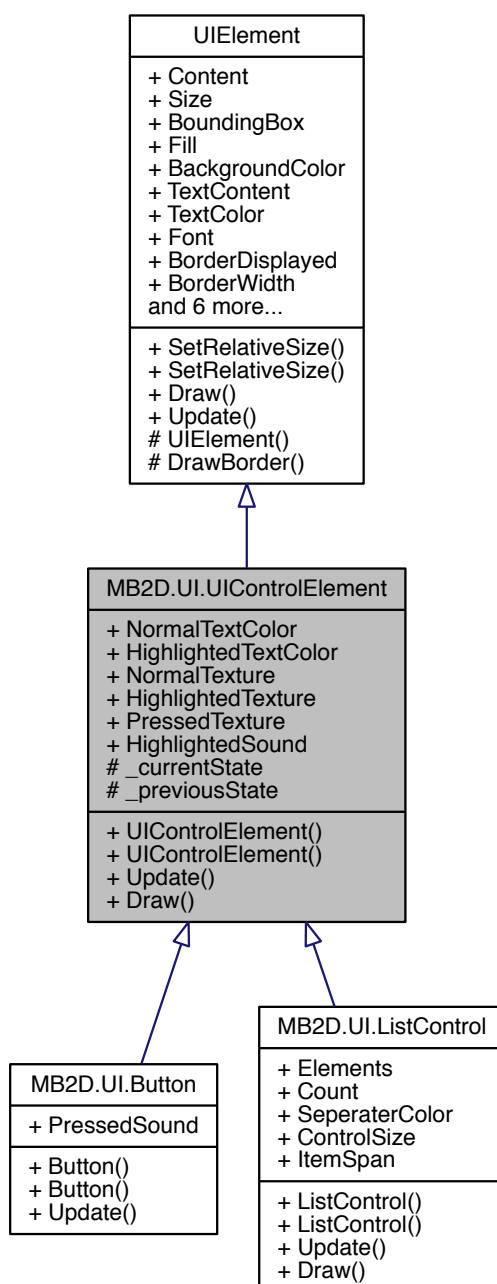
The documentation for this class was generated from the following file:

- MB2D/src/UI/UIData.cs

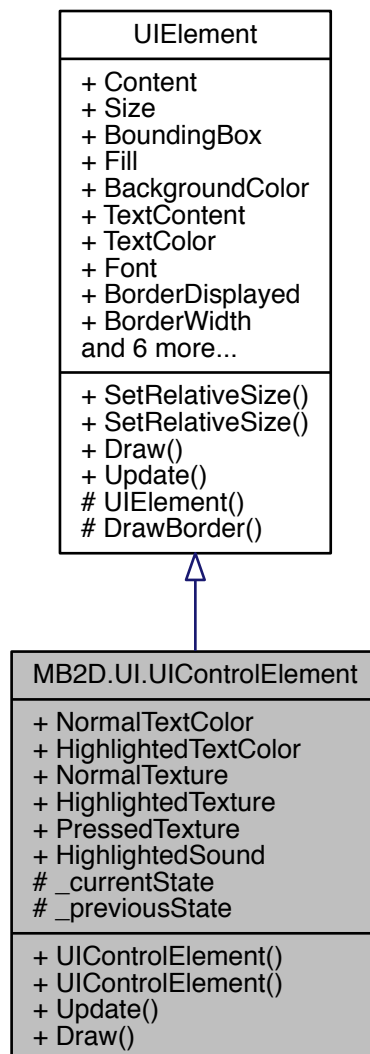
5.63 MB2D.UI.UIControlElement Class Reference

An interactive and controllable [UIElement](#)

Inheritance diagram for MB2D.UI.UIControlElement:



Collaboration diagram for MB2D.UI.UIControlElement:



Public Member Functions

- [UIControlElement](#) (Texture2D normal, Texture2D selected, Texture2D pressed)
Initializes a new instance of the T:MB2D.UI.UIControlElement class.
- [UIControlElement](#) ()
Initializes a new instance of the T:MB2D.UI.UIControlElement class with no textures associated
- override void [Update](#) ()
Update the UIState of the control element based on mouse position
- override void [Draw](#) (SpriteBatch spriteBatch)
Draws the texture associated with the elements current UIState and then its TextContent on top of the texture

Protected Attributes

- [UIState _currentState](#)
The current UIState of the element
- [UIState _previousState](#)
The last state of the element

Properties

- Color [NormalTextColor](#) [get, set]
Gets or sets the TextContent color associated with the Normal UIState of the element.
- Color [HighlightedTextColor](#) [get, set]
Gets or sets the TextContent color associated with the Selected UIState of the element.
- Texture2D [NormalTexture](#) [get, set]
Gets or sets the normal UIState texture.
- Texture2D [HighlightedTexture](#) [get, set]
Gets or sets the selected UIState texture.
- Texture2D [PressedTexture](#) [get, set]
Gets or sets the pressed UIState texture.
- SoundEffect [HighlightedSound](#) [get, set]
Gets or sets the sound played when an element switches to the selected state.

Additional Inherited Members

5.63.1 Detailed Description

An interactive and controllable [UIElement](#)

5.63.2 Constructor & Destructor Documentation

5.63.2.1 UIControlElement() [1/2]

```
MB2D.UI.UIControlElement.UIControlElement (
    Texture2D normal,
    Texture2D selected,
    Texture2D pressed ) [inline]
```

Initializes a new instance of the T:MB2D.UI.UIControlElement class.

Parameters

<i>normal</i>	Normal state texture
<i>selected</i>	Selected state texture
<i>pressed</i>	Pressed state texture

5.63.2.2 UIControlElement() [2/2]

```
MB2D.UI.UIControlElement.UIControlElement ( ) [inline]
```


Initializes a new instance of the T:MB2D.UI.UIControlElement class with no textures associated

5.63.3 Member Function Documentation

5.63.3.1 Draw()

```
override void MB2D.UI.UIControlElement.Draw (
    SpriteBatch spriteBatch ) [inline], [virtual]
```

Draws the texture associated with the elements current UIState and then its TextContent on top of the texture

Parameters

<i>spriteBatch</i>	Sprite batch to draw to
--------------------	-------------------------

Reimplemented from [MB2D.UI.UIElement](#).

5.63.3.2 Update()

```
override void MB2D.UI.UIControlElement.Update ( ) [inline], [virtual]
```

Update the UIState of the control element based on mouse position

Implements [MB2D.UI.UIElement](#).

5.63.4 Member Data Documentation

5.63.4.1 _currentState

```
UIState MB2D.UI.UIControlElement._currentState [protected]
```

The current UIState of the element

5.63.4.2 _previousState

```
UIState MB2D.UI.UIControlElement._previousState [protected]
```

The last state of the element

5.63.5 Property Documentation

5.63.5.1 HighlightedSound

```
SoundEffect MB2D.UI.UIControlElement.HighlightedSound [get], [set]
```

Gets or sets the sound played when an element switches to the selected state.

The highlighted state sound effect.

5.63.5.2 HighlightedTextColor

`Color MB2D.UI.UIControlElement.HighlightedTextColor [get], [set]`

Gets or sets the `TextContent` color associated with the `Selected` `UIState` of the element.

The `TextContents` selected color

5.63.5.3 HighlightedTexture

`Texture2D MB2D.UI.UIControlElement.HighlightedTexture [get], [set]`

Gets or sets the selected `UIState` texture.

The selected texture.

5.63.5.4 NormalTextColor

`Color MB2D.UI.UIControlElement.NormalTextColor [get], [set]`

Gets or sets the `TextContent` color associated with the `Normal` `UIState` of the element.

The `TextContents` normal color

5.63.5.5 NormalTexture

`Texture2D MB2D.UI.UIControlElement.NormalTexture [get], [set]`

Gets or sets the normal `UIState` texture.

The normal texture.

5.63.5.6 PressedTexture

`Texture2D MB2D.UI.UIControlElement.PressedTexture [get], [set]`

Gets or sets the pressed `UIState` texture.

The pressed texture.

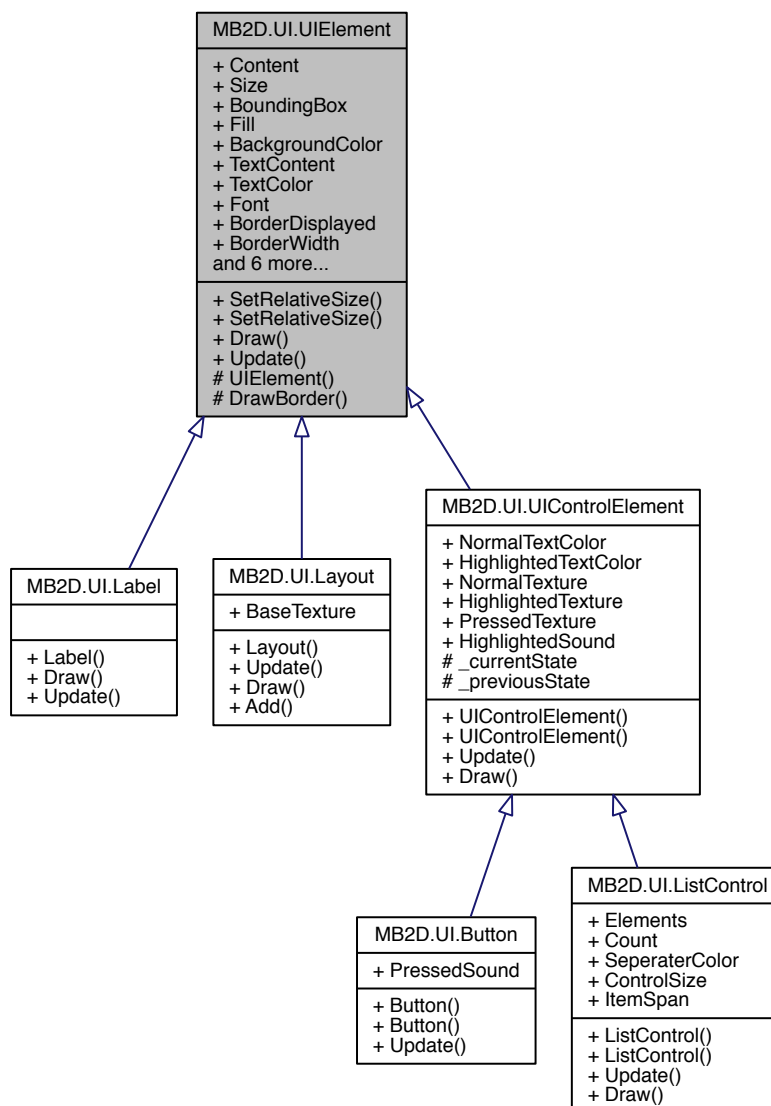
The documentation for this class was generated from the following file:

- MB2D/src/UI/UIControlElement.cs

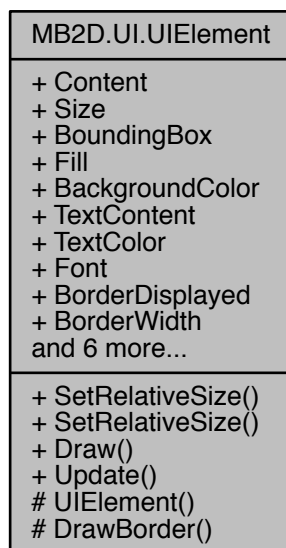
5.64 MB2D.UI.UIElement Class Reference

Defines a [UI](#) object that can be contained within Views and Layouts, drawn, updated, and moved about

Inheritance diagram for MB2D.UI.UIElement:



Collaboration diagram for MB2D.UI.UIElement:



Public Member Functions

- void [SetRelativeSize](#) ([UIContent](#) parent, Point at, Point span)
Sets the size of the element relative to its parent
- void [SetRelativeSize](#) ([UIContent](#) parent, int atRow, int atCol, int rowSpan, int colSpan)
Sets the size of the element relative to its parent
- virtual void [Draw](#) (SpriteBatch spriteBatch)
Draws the element to the window. Overriden in derived classes
- abstract void [Update](#) ()
Update the elements state and handles input. Overriden in derived classes

Protected Member Functions

- [UIElement](#) (int rows, int cols)
Initializes a new instance of the T:MB2D.UI.UIElement class. Sets default property values
- void [DrawBorder](#) (SpriteBatch spriteBatch)
Draws the elements border to the window. Skips sides that have color set to Color.Transparent

Properties

- [UIContent Content](#) [get]
Gets the [UIContent](#) of the element, only available in container elements
- Vector2 [Size](#) [get]
Gets the column and row count of the element
- Rectangle [BoundingBox](#) [get]

- Gets the bounding box of this element*
- bool **Fill** [get, set]
 - Gets or sets a value that indicates the element should be stretched or shrunk to fill its parents bounds exactly*
- Color **BackgroundColor** [get, set]
 - Gets or sets the color of the elements background.*
- string **TextContent** [get, set]
 - Gets or sets the string rendered by the element.*
- Color **TextColor** [get, set]
 - Gets or sets the current color of the text.*
- SpriteFont **Font** [get, set]
 - Gets or sets the font used in rendering the elements TextContent*
- bool **BorderDisplayed** [get, set]
 - Gets or sets a value indicating whether this T:MB2D.UI.UIElement has itsborder displayed.*
- int **BorderWidth** [get, set]
 - Gets or sets the width of the border.*
- Color **BorderColor** [get, set]
 - Gets or sets the color of the border.*
- Color **BorderTopColor** [get, set]
 - Gets or sets the color of the border top.*
- Color **BorderRightColor** [get, set]
 - Gets or sets the color of the border right.*
- Color **BorderBottomColor** [get, set]
 - Gets or sets the color of the border bottom.*
- Color **BorderLeftColor** [get, set]
 - Gets or sets the color of the border left.*
- string **Tag** [get, set]
 - Gets or sets the tag used to quickly access this element and uniquely identify it.*

5.64.1 Detailed Description

Defines a **UI** object that can be contained within Views and Layouts, drawn, updated, and moved about

5.64.2 Constructor & Destructor Documentation

5.64.2.1 UIElement()

```
MB2D.UI.UIElement.UIElement (
    int rows,
    int cols ) [inline], [protected]
```

Initializes a new instance of the T:MB2D.UI.UIElement class. Sets default property values

Parameters

<i>rows</i>	Number of rows this element should span - used only be used for container elements
<i>cols</i>	Number of columns this element should span - used only be used for container elements.

5.64.3 Member Function Documentation

5.64.3.1 Draw()

```
virtual void MB2D.UI.UIElement.Draw (
    SpriteBatch spriteBatch ) [inline], [virtual]
```

Draws the element to the window. Overriden in derived classes

Parameters

<i>spriteBatch</i>	Sprite batch to draw to.
--------------------	--------------------------

Reimplemented in [MB2D.UI.ListControl](#), [MB2D.UI.UIControlElement](#), [MB2D.UI.Layout](#), and [MB2D.UI.Label](#).

Here is the caller graph for this function:



5.64.3.2 DrawBorder()

```
void MB2D.UI.UIElement.DrawBorder (
    SpriteBatch spriteBatch ) [inline], [protected]
```

Draws the elements border to the window. Skips sides that have color set to `Color.Transparent`

Parameters

<i>spriteBatch</i>	Sprite batch to draw the border to.
--------------------	-------------------------------------

5.64.3.3 SetRelativeSize() [1/2]

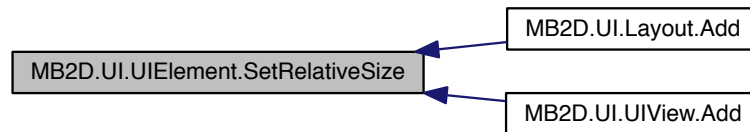
```
void MB2D.UI.UIElement.SetRelativeSize (
    UIContent parent,
    Point at,
    Point span ) [inline]
```

Sets the size of the element relative to its parent

Parameters

<i>parent</i>	Parent content to align to
<i>at</i>	Position element should be set to
<i>span</i>	Number of columns/rows the element should span.

Here is the caller graph for this function:



5.64.3.4 SetRelativeSize() [2/2]

```
void MB2D.UI.UIElement.SetRelativeSize (
    UIContent parent,
    int atRow,
    int atCol,
    int rowSpan,
    int colSpan ) [inline]
```

Sets the size of the element relative to its parent

Parameters

<i>parent</i>	Parent content to align to
<i>atRow</i>	Row position the element should align to.
<i>atCol</i>	Column position the element should align to.
<i>rowSpan</i>	Number of rows in the parent the element should span.
<i>colSpan</i>	Number of columns in the parent the element should span.

5.64.3.5 Update()

```
abstract void MB2D.UI.UIElement.Update ( ) [pure virtual]
```

Update the elements state and handles input. Overriden in derived classes

Implemented in [MB2D.UI.ListControl](#), [MB2D.UI.UIControlElement](#), [MB2D.UI.Label](#), [MB2D.UI.Button](#), and [MB2D.UI.UILayout](#).

Here is the caller graph for this function:



5.64.4 Property Documentation

5.64.4.1 BackgroundColor

`Color MB2D.UI.UIElement.BackgroundColor [get], [set]`

Gets or sets the color of the elements background.

The color of the background.

5.64.4.2 BorderBottomColor

`Color MB2D.UI.UIElement.BorderBottomColor [get], [set]`

Gets or sets the color of the border bottom.

The color of the border bottom.

5.64.4.3 BorderColor

`Color MB2D.UI.UIElement.BorderColor [get], [set]`

Gets or sets the color of the border.

The color of the border. Resets all border sides' colors to this color.

5.64.4.4 BorderDisplayed

`bool MB2D.UI.UIElement.BorderDisplayed [get], [set]`

Gets or sets a value indicating whether this T:MB2D.UI.UIElement has itsborder displayed.

`true` if border should be displayed; otherwise, `false`.

5.64.4.5 BorderLeftColor

`Color MB2D.UI.UIElement.BorderLeftColor [get], [set]`

Gets or sets the color of the border left.

The color of the border left.

5.64.4.6 BorderRightColor

`Color MB2D.UI.UIElement.BorderRightColor [get], [set]`

Gets or sets the color of the border right.

The color of the border right.

5.64.4.7 BorderTopColor

`Color MB2D.UI.UIElement.BorderTopColor [get], [set]`

Gets or sets the color of the border top.

The color of the border top.

5.64.4.8 BorderWidth

`int MB2D.UI.UIElement.BorderWidth [get], [set]`

Gets or sets the width of the border.

The width of the border.

5.64.4.9 BoundingBox

`Rectangle MB2D.UI.UIElement.BoundingBox [get]`

Gets the bounding box of this element

The bounding box.

5.64.4.10 Content

`UIContent MB2D.UI.UIElement.Content [get]`

Gets the [UIContent](#) of the element, only available in container elements

The content.

5.64.4.11 Fill

`bool MB2D.UI.UIElement.Fill [get], [set]`

Gets or sets a value that indicates the element should be stretched or shrunk to fill its parents bounds exactly

true if set to fill parent; otherwise, false.

5.64.4.12 Font

`SpriteFont MB2D.UI.UIElement.Font [get], [set]`

Gets or sets the font used in rendering the elements `TextContent`

The font.

5.64.4.13 Size

`Vector2 MB2D.UI.UIElement.Size [get]`

Gets the column and row count of the element

The elements grid size.

5.64.4.14 Tag

```
string MB2D.UI.UIElement.Tag [get], [set]
```

Gets or sets the tag used to quickly access this element and uniquely identify it.

The tag.

5.64.4.15 TextColor

```
Color MB2D.UI.UIElement.TextColor [get], [set]
```

Gets or sets the current color of the text.

The text contents current color value.

5.64.4.16 TextContent

```
string MB2D.UI.UIElement.TextContent [get], [set]
```

Gets or sets the string rendered by the element.

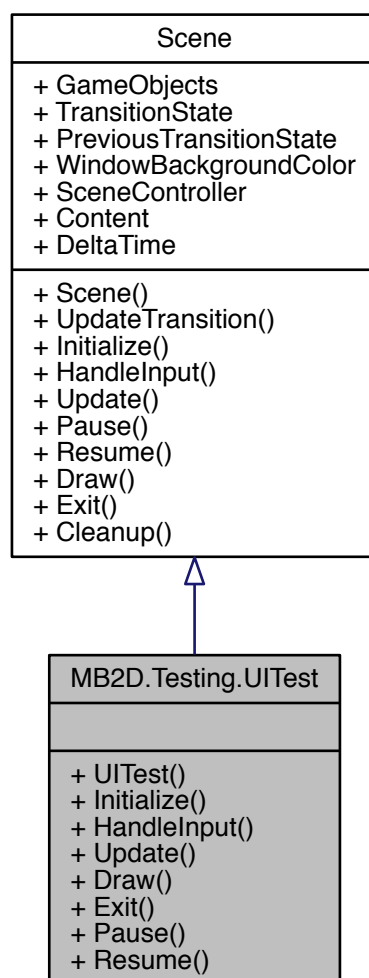
The content of the text.

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

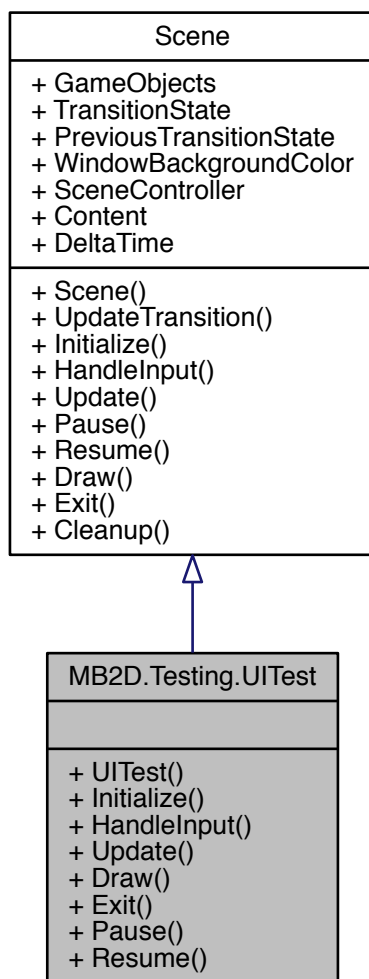
- MB2D/src/UI/UIElement.cs

5.65 MB2D.Testing.UITest Class Reference

Inheritance diagram for MB2D.Testing.UITest:



Collaboration diagram for MB2D.Testing.UITest:



Public Member Functions

- **UITest** ([EntityMap](#) map, [ContentManager](#) content)
- override void [Initialize](#) ()
Initialize this scene and loads all resources. Runs logic to execute during the Initializing state. Set state to None to end.
- override void [HandleInput](#) ()
Handles all input for the scene
- override void [Update](#) ()
Updates game logic and changes scene state.
- override void [Draw](#) ([SpriteBatch](#) spriteBatch, [SpriteBatch](#) uiSpriteBatch)
Draws entities and [UI](#) elements to the specified SpriteBatches
- override void [Exit](#) ()
Runs logic to execute while the scene is in the Exiting state. Set state to Null to end.

- override void [Pause](#) ()
Runs logic to execute while the scene is in the Pausing state. Set state to None to end.
- override void [Resume](#) ()
Runs logic to execute while the scene is in the Resuming state. Set state to None to end.

Additional Inherited Members

5.65.1 Member Function Documentation

5.65.1.1 Draw()

```
override void MB2D.Testing.UITest.Draw (  
    SpriteBatch spriteBatch,  
    SpriteBatch uiSpriteBatch ) [inline], [virtual]
```

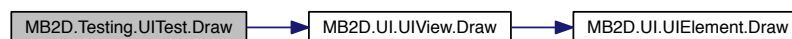
Draws entities and [UI](#) elements to the specified SpriteBatches

Parameters

<i>spriteBatch</i>	World-coordinate based sprite batch.
<i>uiSpriteBatch</i>	Camera-based User Interface sprite batch.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



5.65.1.2 Exit()

```
override void MB2D.Testing.UITest.Exit ( ) [inline], [virtual]
```

Runs logic to execute while the scene is in the Exiting state. Set state to Null to end.

Implements [MB2D.Scenes.Scene](#).

5.65.1.3 HandleInput()

```
override void MB2D.Testing.UITest.HandleInput ( ) [inline], [virtual]
```

Handles all input for the scene

Implements [MB2D.Scenes.Scene](#).

5.65.1.4 Initialize()

```
override void MB2D.Testing.UITest.Initialize ( ) [inline], [virtual]
```

Initialize this scene and loads all resources. Runs logic to execute during the Initializing state. Set state to None to end.

Implements [MB2D.Scenes.Scene](#).

5.65.1.5 Pause()

```
override void MB2D.Testing.UITest.Pause ( ) [inline], [virtual]
```

Runs logic to execute while the scene is in the Pausing state. Set state to None to end.

Implements [MB2D.Scenes.Scene](#).

5.65.1.6 Resume()

```
override void MB2D.Testing.UITest.Resume ( ) [inline], [virtual]
```

Runs logic to execute while the scene is in the Resuming state. Set state to None to end.

Implements [MB2D.Scenes.Scene](#).

5.65.1.7 Update()

```
override void MB2D.Testing.UITest.Update ( ) [inline], [virtual]
```

Updates game logic and changes scene state.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



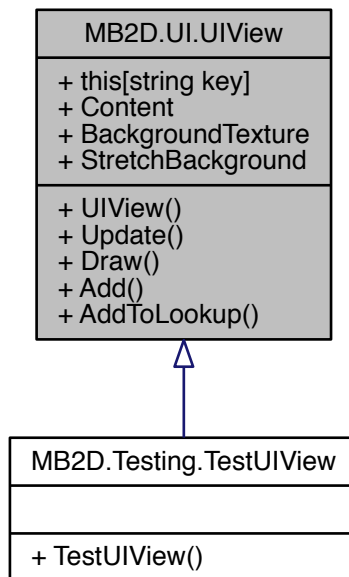
The documentation for this class was generated from the following file:

- `MB2D/src/Test/UITest.cs`

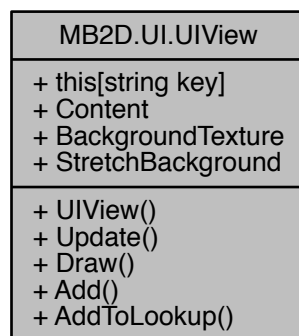
5.66 MB2D.UI.UIView Class Reference

A single context for all [UI](#) elements and layouts.

Inheritance diagram for MB2D.UI.UIView:



Collaboration diagram for MB2D.UI.UIView:



Public Member Functions

- [UIView](#) (int rows, int cols)

Initializes a new instance of the T:MB2D.UI.UIView class. Divides itself into the number of rows and columns evenly based on the size of the current viewport.

- void [Update](#) ()
Updates and handles input for all elements in the View.
- void [Draw](#) (SpriteBatch spriteBatch, SpriteBatch uiSpriteBatch)
Draws the View and its elements to the window
- void [Add](#) (UIElement element, int atRow, int atCol, int rowSpan, int colSpan)
Adds a new UIElement to the View
- void [AddToLookup](#) (UIElement element)
Adds an element to the lookup table used by the view

Properties

- [UIElement this\[string key\]](#) [get]
Gets the T:MB2D.UI.UIElement with the specified key.
- [UIElement \[,\] Content](#) [get]
Gets the elements this View contains in a 2D array
- Texture2D [BackgroundTexture](#) [get, set]
Gets or sets the background texture of the view.
- bool [StretchBackground](#) [get, set]
Gets or sets a value indicating whether the background image should stretch to fit the window.

5.66.1 Detailed Description

A single context for all [UI](#) elements and layouts.

5.66.2 Constructor & Destructor Documentation

5.66.2.1 UIView()

```
MB2D.UI.UIView.UIView (
    int rows,
    int cols ) [inline]
```

Initializes a new instance of the T:MB2D.UI.UIView class. Divides itself into the number of rows and columns evenly based on the size of the current viewport.

Parameters

<i>rows</i>	Number of rows in the view
<i>cols</i>	Number of columns in the view

5.66.3 Member Function Documentation

5.66.3.1 Add()

```
void MB2D.UI.UIView.Add (
    UIElement element,
```



```

    int atRow,
    int atCol,
    int rowSpan,
    int colSpan ) [inline]

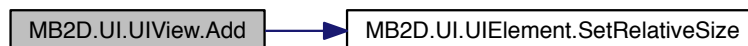
```

Adds a new [UIElement](#) to the View

Parameters

<i>element</i>	Element to add.
<i>atRow</i>	Row position in the View.
<i>atCol</i>	Column position in the View.
<i>rowSpan</i>	Number of rows the element takes up.
<i>colSpan</i>	Number of columns the element takes up.

Here is the call graph for this function:



5.66.3.2 AddToLookup()

```

void MB2D.UI.UIView.AddToLookup (
    UIElement element ) [inline]

```

Adds an element to the lookup table used by the view

Parameters

<i>element</i>	Element to add.
----------------	-----------------

Here is the caller graph for this function:



5.66.3.3 Draw()

```

void MB2D.UI.UIView.Draw (
    SpriteBatch spriteBatch,
    SpriteBatch uiSpriteBatch ) [inline]

```

Draws the View and its elements to the window

Parameters

<i>uiSpriteBatch</i>	Sprite batch to draw to.
----------------------	--------------------------

Here is the call graph for this function:



Here is the caller graph for this function:



5.66.3.4 Update()

```
void MB2D.UI.UIView.Update ( ) [inline]
```

Updates and handles input for all elements in the View.

Here is the call graph for this function:



Here is the caller graph for this function:



5.66.4 Property Documentation

5.66.4.1 BackgroundTexture

Texture2D MB2D.UI.UIView.BackgroundTexture [get], [set]

Gets or sets the background texture of the view.

The background texture.

5.66.4.2 Content

UIElement [,] MB2D.UI.UIView.Content [get]

Gets the elements this View contains in a 2D array

All UIElements.

5.66.4.3 StretchBackground

bool MB2D.UI.UIView.StretchBackground [get], [set]

Gets or sets a value indicating whether the background image should stretch to fit the window.

true if background should be stretched; otherwise, false.

5.66.4.4 this[string key]

UIElement MB2D.UI.UIView.this[string key] [get]

Gets the T:MB2D.UI.UIElement with the specified key.

Parameters

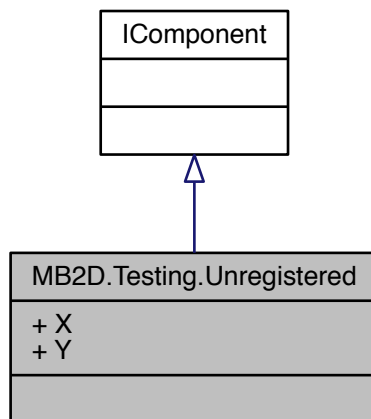
key	Tag of the element.
-----	---------------------

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

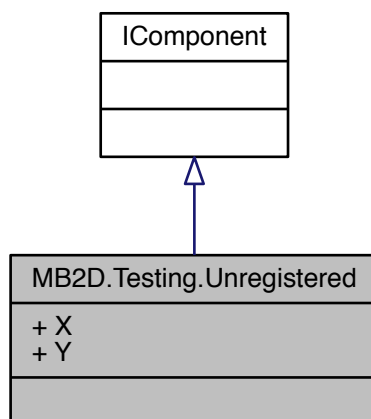
- MB2D/src/UI/UIView.cs

5.67 MB2D.Testing.Unregistered Class Reference

Inheritance diagram for MB2D.Testing.Unregistered:



Collaboration diagram for MB2D.Testing.Unregistered:



Properties

- `int X` [get, set]
- `int Y` [get, set]

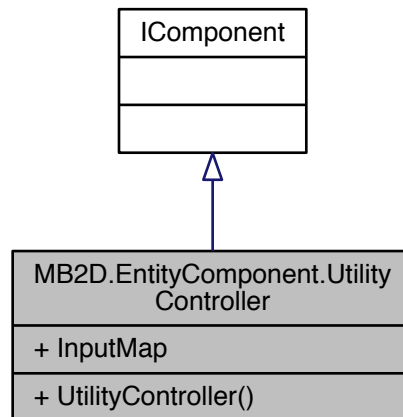
The documentation for this class was generated from the following file:

- `MB2D/src/Test/TestSystem.cs`

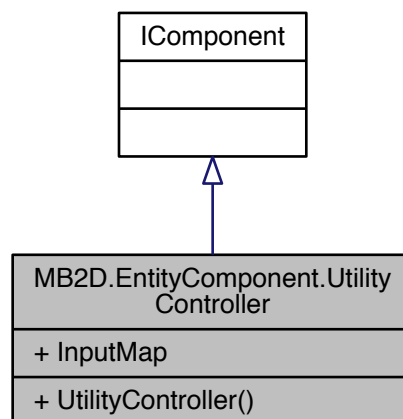
5.68 MB2D.EntityComponent.UtilityController Class Reference

Declares the attached entity as able to control utility commands such as opening the debug console

Inheritance diagram for MB2D.EntityComponent.UtilityController:



Collaboration diagram for MB2D.EntityComponent.UtilityController:



Public Member Functions

- [UtilityController\(\)](#)

Initializes a new instance of the T:MidnightBlue.UtilityController component with default input assignment

Properties

- [InputMap InputMap](#) [get]
Gets the input map.

5.68.1 Detailed Description

Declares the attached entity as able to control utility commands such as opening the debug console

5.68.2 Constructor & Destructor Documentation

5.68.2.1 UtilityController()

`MB2D.EntityComponent.UtilityController.UtilityController () [inline]`

Initializes a new instance of the T:MidnightBlue.UtilityController component with default input assignment

5.68.3 Property Documentation

5.68.3.1 InputMap

`InputMap MB2D.EntityComponent.UtilityController.InputMap [get]`

Gets the input map.

The input map.

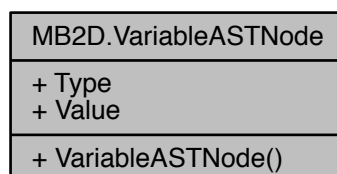
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Components/UtilityController.cs

5.69 MB2D.VariableASTNode Class Reference

Represents a variable with a type and a value

Collaboration diagram for MB2D.VariableASTNode:



Public Member Functions

- [VariableASTNode](#) ([Type](#) type, object value)
Initializes a new instance of the T:MB2D.VariableASTNode class.

Properties

- [Type](#) [Type](#) [get]
Gets the variables type info
- [Value](#) [get]
Gets the value to assign to the variable

5.69.1 Detailed Description

Represents a variable with a type and a value

5.69.2 Constructor & Destructor Documentation

5.69.2.1 VariableASTNode()

```
MB2D.VariableASTNode.VariableASTNode (
    Type type,
    object value ) [inline]
```

Initializes a new instance of the T:MB2D.VariableASTNode class.

Parameters

<i>type</i>	Type of the variable.
<i>value</i>	Value to assign.

5.69.3 Property Documentation

5.69.3.1 Type

```
Type MB2D.VariableASTNode.Type [get]
```

Gets the variables type info

The type.

5.69.3.2 Value

```
object MB2D.VariableASTNode.Value [get]
```

Gets the value to assign to the variable

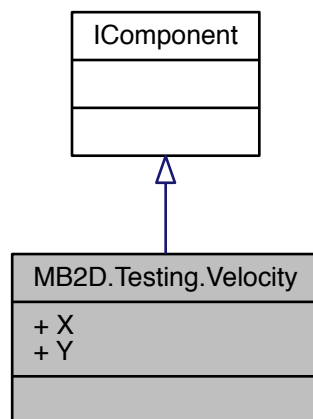
The value.

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

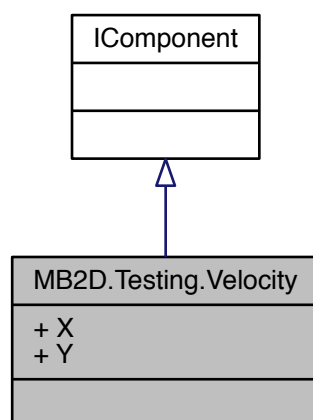
- MB2D/src/MBConsole/MBConsoleAST.cs

5.70 MB2D.Testing.Velocity Class Reference

Inheritance diagram for MB2D.Testing.Velocity:



Collaboration diagram for MB2D.Testing.Velocity:



Properties

- `int X` [get, set]
- `int Y` [get, set]

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

- `MB2D/src/Test/TestSystem.cs`