Table of Contents

Namespace MonoUI

Classes

Button

Class for GUI button.

Checkbox

Class for GUI checkbox.

DropDownList

Class for GUI drop-down list.

GUI

Base class for GUI tree.

Image

Class for GUI image.

Input

Class for all keyboard and mouse input.

Label

Class for GUI label.

ProgressBar

Class for GUI progress bar.

RadioButton

Class for GUI radio button.

Slider

Class for GUI slider.

StatusBar

Base class for status bars. Only for inheritance.

TextBox

Class for GUI text box.

Tool

The utility class of the GUI system.

Tooltip

Class for GUI tooltip.

Widget

Base class for all control elements. Only for inheritance.

Structs

GUI.LayerDepth

Values between 0.9 and 1.0 for all the different layers of drawable elements. (Exclusive)

Enums

Widget.DockControl

Defines where the control element should dock onto the screen.

Delegates

GUI.DrawHandler

Delegate for managing draw calls.

Widget.EventHandler

Delegate for managing events.

Class Button

Class for GUI button.

Inheritance

System.Object
GUI
Widget
Button

Inherited Members

Widget.IsActive
Widget.IsVisible
Widget.Color
Widget.SelectionColor
Widget.IsSelected
Widget.IsPressed
Widget.OnClick
Widget.Position
Widget.Tooltip
Widget.Dock
Widget.Offset
Widget.Layer
Widget.Scale
Widget.Texture
Widget.ToggleIsActive()
Widget.ToggleIsVisible()
Widget.CalculatePosition(Widget.DockControl, Point, Texture2D, Single)
Widget.IsTextureSelected(Texture2D, Vector2, Single)
Widget.IsTexturePressed(Texture2D, Vector2, Single)
Widget.ResetPressState()
GUI.DrawAll
GUI.Tag
GUI.Display<T>()
GUI.RemoveElement()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: MonoUI
Assembly: MonoUI.dll

Syntax

```
public class Button : Widget, IDrawable
```

Constructors

Button(Game, Widget.DockControl, Point, Texture2D, Texture2D, Texture2D, Single, Single)

Creates a new button with a picture.

Declaration

```
public Button(Game game, Widget.DockControl dock, Point offset, Texture2D textureNormal, Texture2D
texturePressed, Texture2D picture, float insideScale, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the control element is to be created.
Widget.DockControl	dock	The location the control element should dock onto.
Microsoft.Xna.Framework.Point	offset	The offset in pixel between the GUI element and the dock location.
Microsoft.Xna.Framework.Graphics.Texture2D	textureNormal	The texture for the button in normal state.
Microsoft.Xna.Framework.Graphics.Texture2D	texturePressed	The texture for the button in pressed state.
Microsoft.Xna.Framework.Graphics.Texture2D	picture	The name of the picture that is displayed within the button.
System.Single	insideScale	A value between 0 and 1 for scaling the picture inside the Button. 1 is the original size.
System.Single	scale	The value for scaling the textures. 1.0 is no scaling.

Button(Game, Widget.DockControl, Point, Texture2D, Texture2D, Single)

Creates a new simple button.

Declaration

```
public Button(Game game, Widget.DockControl dock, Point offset, Texture2D textureNormal, Texture2D
texturePressed, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the control element is to be created.
Widget.DockControl	dock	The location the control element should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the control element and the dock location.

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.Texture2D	textureNormal	The texture for the button in normal state.
Microsoft.Xna.Framework.Graphics.Texture2D	texturePressed	The texture for the button in pressed state.
System.Single	scale	The value for scaling the textures. 1.0 is no scaling.

Button(Game, Widget.DockControl, Point, Texture2D, Texture2D, String, Color, SpriteFont, Single)

Creates a new button with text.

Declaration

```
public Button(Game game, Widget.DockControl dock, Point offset, Texture2D textureNormal, Texture2D
texturePressed, string text, Color textColor, SpriteFont font, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the control element is to be created.
Widget.DockControl	dock	The location the control element should dock onto.
Microsoft.Xna.Framework.Point	offset	The offset in pixel between the control element and the dock location.
Microsoft.Xna.Framework.Graphics.Texture2D	textureNormal	The texture for the button in normal state.
Microsoft.Xna.Framework.Graphics.Texture2D	texturePressed	The texture for the button in pressed state.
System.String	text	The text that is to be displayed within the button.
Microsoft.Xna.Framework.Color	textColor	The color of the text.
Microsoft.Xna.Framework.Graphics.SpriteFont	font	The font of the text.
System.Single	scale	The value for scaling the textures. 1.0 is no scaling.

Properties

Text

The text on the button.

Declaration

```
public string Text { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

TextColor

The color of the text.

Declaration

```
public Color TextColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Color	

Methods

Draw(SpriteBatch, GameTime)

Draws the the button.

Declaration

```
public override void Draw(in SpriteBatch spriteBatch, in GameTime gameTime)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.SpriteBatch	spriteBatch	The sprite batch that draws the texture.
Microsoft.Xna.Framework.GameTime	gameTime	The game time for the elapsed time since the last update call.

Overrides

Widget.Draw(SpriteBatch, GameTime)

SetPosition(Widget.DockControl, Point)

Sets the position of the control element.

Declaration

```
public override void SetPosition(Widget.DockControl dock, Point offset)
```

Parameters

TYPE	NAME	DESCRIPTION
Widget.DockControl	dock	The location the control element should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the control element and the dock location.

Overrides

Widget.SetPosition(Widget.DockControl, Point)

Class Checkbox

Class for GUI checkbox.

Inheritance

System.Object
GUI
Widget
Checkbox

Inherited Members

Widget.IsActive
Widget.IsVisible
Widget.Color
Widget.SelectionColor
Widget.IsSelected
Widget.IsPressed
Widget.OnClick
Widget.Position
Widget.Tooltip
Widget.Dock
Widget.Offset
Widget.Layer
Widget.Scale
Widget.Texture
Widget.ToggleIsActive()
Widget.ToggleIsVisible()
Widget.CalculatePosition(Widget.DockControl, Point, Texture2D, Single)
Widget.SetPosition(Widget.DockControl, Point)
Widget.IsTextureSelected(Texture2D, Vector2, Single)
Widget.IsTexturePressed(Texture2D, Vector2, Single)
Widget.ResetPressState()
GUI.DrawAll
GUI.Tag
GUI.Display<T>()
GUI.RemoveElement()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public class Checkbox : Widget, IDrawable
```

Constructors

Checkbox(Game, Widget.DockControl, Point, Texture2D, Texture2D, Boolean, Single)

Creates a new checkbox.

Declaration

```
public Checkbox(Game game, Widget.DockControl dock, Point offset, Texture2D textureOff, Texture2D textureOn,
bool state, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the control element is to be created.
Widget.DockControl	dock	The location the control element should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the control element and the dock location.
Microsoft.Xna.Framework.Graphics.Texture2D	textureOff	The the texture for the of/not checked state.
Microsoft.Xna.Framework.Graphics.Texture2D	textureOn	The the texture for the on/checked state.
System.Boolean	state	The start state of the checkbox. True for checked.
System.Single	scale	The value for scaling the texture. 1.0 is no scaling.

Properties

State

The current On/Off state of the checkbox.

Declaration

```
public bool State { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

Draw(SpriteBatch, GameTime)

Draws the checkbox.

Declaration

```
public override void Draw(in SpriteBatch spriteBatch, in GameTime gameTime)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.SpriteBatch	spriteBatch	The sprite batch that draws the texture.
Microsoft.Xna.Framework.GameTime	gameTime	The game time for the elapsed time since the last update call.

Overrides

Widget.Draw(SpriteBatch, GameTime)

Class DropDownList

Class for GUI drop-down list.

Inheritance

System.Object
GUI
Widget
DropDownList

Inherited Members

Widget.IsActive
Widget.IsVisible
Widget.Color
Widget.SelectionColor
Widget.IsSelected
Widget.IsPressed
Widget.OnClick
Widget.Position
Widget.Tooltip
Widget.Dock
Widget.Offset
Widget.Layer
Widget.Scale
Widget.Texture
Widget.ToggleIsActive()
Widget.ToggleIsVisible()
Widget.CalculatePosition(Widget.DockControl, Point, Texture2D, Single)
Widget.SetPosition(Widget.DockControl, Point)
Widget.IsTextureSelected(Texture2D, Vector2, Single)
Widget.IsTexturePressed(Texture2D, Vector2, Single)
Widget.ResetPressState()
GUI.DrawAll
GUI.Tag
GUI.Display<T>()
GUI.RemoveElement()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public class DropDownList : Widget, IDrawable
```

Constructors

DropDownList(Game, Widget.DockControl, Point, String[], Color, SpriteFont, Int32, Int32)

Creates a new drop-down list.

Declaration

```
public DropDownList(Game game, Widget.DockControl dock, Point offset, string[] items, Color textColor,
SpriteFont font, int width = 180, int height = 25)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the drop-down list is to be created.
Widget.DockControl	dock	The location the drop-down list should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the drop-down list and the dock location.
System.String[]	items	The selectable items for the drop-down list. The item at index 0 is the first active item.
Microsoft.Xna.Framework.Color	textColor	The color of the text.
Microsoft.Xna.Framework.Graphics.SpriteFont	font	The font the text is using. The font size should match with the measurements of the drop-down list.
System.Int32	width	The total width of the drop-down list in pixel. This have to be greater than zero.
System.Int32	height	The height of the collapsed drop-down list in pixel. This have to be greater than zero.

Properties

ActiveItem

The currently selected item.

Declaration

```
public string ActiveItem { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ItemBackgroundColor

The color of the texture of a non-selected item.

Declaration

```
public Color ItemBackgroundColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Color	

ItemSelectedColor

The color of the texture of a selected item.

Declaration

```
public Color ItemSelectedColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Color	

TextColor

The color of the text.

Declaration

```
public Color TextColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Color	

Methods

Draw(SpriteBatch, GameTime)

Draws the drop-down list.

Declaration

```
public override void Draw(in SpriteBatch spriteBatch, in GameTime gameTime)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.SpriteBatch	spriteBatch	The sprite batch that draws the texture.
Microsoft.Xna.Framework.GameTime	gameTime	The game time for the elapsed time since the last update call.

Overrides

Widget.Draw(SpriteBatch, GameTime)

SetPicture(Texture2D, Color, Single)

Places an optional picture on the right side of the drop-down list.

Declaration

```
public void SetPicture(Texture2D texture, Color color, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.Texture2D	texture	The texture that is to be placed.
Microsoft.Xna.Framework.Color	color	The color of the texture.
System.Single	scale	The scale of the texture. 1.0 is no scaling.

Class GUI

Base class for GUI tree.

Inheritance

System.Object
GUI
Tooltip
Widget

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public abstract class GUI
```

Constructors

GUI()

Base constructor for inheritance.

Declaration

```
protected GUI()
```

Properties

DrawAll

Draws all GUI elements.

Declaration

```
public static GUI.DrawHandler DrawAll { get; }
```

Property Value

TYPE	DESCRIPTION
GUI.DrawHandler	

Tag

For storing additional informations about the GUI element.

Declaration

```
public string Tag { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

Display<T>()

Returns items of a specific GUI type or type that derived from it.

Declaration

```
public static IEnumerable<T> Display<T>()

    where T : GUI
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<T>	

Type Parameters

NAME	DESCRIPTION
T	The type of the GUI items, which are searched for.

RemoveElement()

Removes the item from the GUI collection.

Declaration

```
public void RemoveElement()
```

Delegate GUI.DrawHandler

Delegate for managing draw calls.

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public delegate void DrawHandler(in SpriteBatch spriteBatch, in GameTime gameTime);
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.SpriteBatch	spriteBatch	The sprite batch that draws the textures.
Microsoft.Xna.Framework.GameTime	gameTime	The game time of the game instance.

Struct GUI.LayerDepth

Values between 0.9 and 1.0 for all the different layers of drawable elements. (Exclusive)

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
protected struct LayerDepth
```

Fields

lowerPicture

Lower layer for pictures.

Declaration

```
public const float lowerPicture = 0.92F
```

Field Value

TYPE	DESCRIPTION
System.Single	

lowerText

Lower layer for texts.

Declaration

```
public const float lowerText = 0.93F
```

Field Value

TYPE	DESCRIPTION
System.Single	

lowerTexture

Lower layer for textures.

Declaration

```
public const float lowerTexture = 0.91F
```

Field Value

TYPE	DESCRIPTION
System.Single	

middlePicture

Middle layer for pictures.

Declaration

```
public const float middlePicture = 0.95F
```

Field Value

TYPE	DESCRIPTION
System.Single	

middleText

Middle layer for texts.

Declaration

```
public const float middleText = 0.96F
```

Field Value

TYPE	DESCRIPTION
System.Single	

middleTexture

Middle layer for textures.

Declaration

```
public const float middleTexture = 0.94F
```

Field Value

TYPE	DESCRIPTION
System.Single	

upperPicture

Upper layer for pictures.

Declaration

```
public const float upperPicture = 0.98F
```

Field Value

TYPE	DESCRIPTION
System.Single	

upperText

Upper layer for texts.

Declaration

```
public const float upperText = 0.99F
```

Field Value

TYPE	DESCRIPTION
System.Single	

upperTexture

Upper layer for textures.

Declaration

```
public const float upperTexture = 0.97F
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class Image

Class for GUI image.

Inheritance

System.Object

GUI

Widget

Image

Inherited Members

Widget.IsActive

Widget.IsVisible

Widget.Color

Widget.SelectionColor

Widget.IsSelected

Widget.IsPressed

Widget.OnClick

Widget.Position

Widget.Tooltip

Widget.Dock

Widget.Offset

Widget.Layer

Widget.Scale

Widget.Texture

Widget.ToggleIsActive()

Widget.ToggleIsVisible()

Widget.CalculatePosition(Widget.DockControl, Point, Texture2D, Single)

Widget.SetPosition(Widget.DockControl, Point)

Widget.IsTextureSelected(Texture2D, Vector2, Single)

Widget.IsTexturePressed(Texture2D, Vector2, Single)

Widget.ResetPressState()

GUI.DrawAll

GUI.Tag

GUI.Display<T>()

GUI.RemoveElement()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public class Image : Widget, IDrawable
```

Constructors

Image(Game, Widget.DockControl, Point, Texture2D, Single)

Creates a new image within a box.

Declaration

```
public Image(Game game, Widget.DockControl dock, Point offset, Texture2D image, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the control element is to be created.
Widget.DockControl	dock	The location the control element should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the control element and the dock location.
Microsoft.Xna.Framework.Graphics.Texture2D	image	The texture for the image.
System.Single	scale	The value for scaling the texture. 1.0 is no scaling.

Methods

Draw(SpriteBatch, GameTime)

Draws the image.

Declaration

```
public override void Draw(in SpriteBatch spriteBatch, in GameTime gameTime)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.SpriteBatch	spriteBatch	The sprite batch that draws the texture.
Microsoft.Xna.Framework.GameTime	gameTime	The game time for the elapsed time since the last update call.

Overrides

Widget.Draw(SpriteBatch, GameTime)

SetImage(Texture2D)

Sets a new image.

Declaration

```
public void SetImage(Texture2D newImage)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.Texture2D	newImage	The texture for the new image.

Class Input

Class for all keyboard and mouse input.

Inheritance

System.Object
Input

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public static class Input
```

Properties

IsAnyKeyPressed

Checks if any key is pressed down.

Declaration

```
public static bool IsAnyKeyPressed { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

IsLeftMouseButtonPressed

Checks if the left mouse button is pressed down.

Declaration

```
public static bool IsLeftMouseButtonPressed { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

IsRightMouseButtonPressed

Checks if the right mouse button is pressed down.

Declaration

```
public static bool IsRightMouseButtonPressed { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

MousePosition

Returns the mouse position in screen space coordinates;

Declaration

```
public static Point MousePosition { get; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Point	

PressedKeys

Returns all pressed keys;

Declaration

```
public static Keys[] PressedKeys { get; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Input.Keys[]	

Methods

IsSpecificKeyPressed(Keys)

Checks if a specific key is pressed down.

Declaration

```
public static bool IsSpecificKeyPressed(Keys key)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Input.Keys	key	The name of the key that is pressed.

Returns

TYPE	DESCRIPTION
System.Boolean	

Class Label

Class for GUI label.

Inheritance

System.Object

GUI

Widget

Label

Inherited Members

Widget.IsActive

Widget.IsVisible

Widget.Color

Widget.SelectionColor

Widget.IsSelected

Widget.IsPressed

Widget.OnClick

Widget.Position

Widget.Tooltip

Widget.Dock

Widget.Offset

Widget.Layer

Widget.Scale

Widget.Texture

Widget.ToggleIsActive()

Widget.ToggleIsVisible()

Widget.CalculatePosition(Widget.DockControl, Point, Texture2D, Single)

Widget.SetPosition(Widget.DockControl, Point)

Widget.IsTextureSelected(Texture2D, Vector2, Single)

Widget.IsTexturePressed(Texture2D, Vector2, Single)

Widget.ResetPressState()

GUI.DrawAll

GUI.Tag

GUI.Display<T>()

GUI.RemoveElement()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public class Label : Widget, IDrawable
```

Constructors

Label(Game, Widget.DockControl, Point, String, Color, SpriteFont)

Creates a new simple label.

Declaration

```
public Label(Game game, Widget.DockControl dock, Point offset, string text, Color textColor, SpriteFont font)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the control element is to be created.
Widget.DockControl	dock	The location the control element should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the control element and the dock location.
System.String	text	The text that is to be displayed within the label.
Microsoft.Xna.Framework.Color	textColor	The color of the text.
Microsoft.Xna.Framework.Graphics.SpriteFont	font	The font of the text.

Label(Game, Widget.DockControl, Point, String, Color, SpriteFont, Texture2D, Single)

Creates a new label with a background texture.

Declaration

```
public Label(Game game, Widget.DockControl dock, Point offset, string text, Color textColor, SpriteFont font,
Texture2D texture, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the control element is to be created.
Widget.DockControl	dock	The location the control element should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the control element and the dock location.
System.String	text	The text that is to be displayed within the label.
Microsoft.Xna.Framework.Color	textColor	The color of the text.

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.SpriteFont	font	The font of the text.
Microsoft.Xna.Framework.Graphics.Texture2D	texture	The background texture for the label.
System.Single	scale	The value for scaling the texture. 1.0 is no scaling.

Properties

Text

The text of the label.

Declaration

```
public string Text { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

TextColor

The color of the text.

Declaration

```
public Color TextColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Color	

Methods

Draw(SpriteBatch, GameTime)

Draws the the Label.

Declaration

```
public override void Draw(in SpriteBatch spriteBatch, in GameTime gameTime)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.SpriteBatch	spriteBatch	The sprite batch that draws the texture.

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.GameTime	gameTime	The game time for the elapsed time since the last update call.

Overrides

Widget.Draw(SpriteBatch, GameTime)

Class ProgressBar

Class for GUI progress bar.

Inheritance

System.Object
GUI
Widget
StatusBar
ProgressBar

Inherited Members

StatusBar.Alignment
StatusBar.StatusTexture
StatusBar.StatusBarSize()
Widget.IsActive
Widget.IsVisible
Widget.Color
Widget.SelectionColor
Widget.IsSelected
Widget.IsPressed
Widget.OnClick
Widget.Position
Widget.Tooltip
Widget.Dock
Widget.Offset
Widget.Layer
Widget.Scale
Widget.Texture
Widget.ToggleIsActive()
Widget.ToggleIsVisible()
Widget.CalculatePosition(Widget.DockControl, Point, Texture2D, Single)
Widget.SetPosition(Widget.DockControl, Point)
Widget.IsTextureSelected(Texture2D, Vector2, Single)
Widget.IsTexturePressed(Texture2D, Vector2, Single)
Widget.ResetPressState()
GUI.DrawAll
GUI.Tag
GUI.Display<T>()
GUI.RemoveElement()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public class ProgressBar : StatusBar, IDrawable
```

Constructors

ProgressBar(Game, Widget.DockControl, Point, Single, Texture2D, Texture2D, Single)

Creates a new progress bar with textures.

Declaration

```
public ProgressBar(Game game, Widget.DockControl dock, Point offset, float status, Texture2D
textureBackground, Texture2D textureProgress, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the progress bar is to be created.
Widget.DockControl	dock	The location the progress bar should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the progress bar and the dock location.
System.Single	status	The status of the progress bar, 0 means 0% filled and 1 means 100% filled.
Microsoft.Xna.Framework.Graphics.Texture2D	textureBackground	The texture for the background bar.
Microsoft.Xna.Framework.Graphics.Texture2D	textureProgress	The texture for the foreground bar.
System.Single	scale	The value for scaling the textures. 1.0 is no scaling.

ProgressBar(Game, Widget.DockControl, Point, Single, Int32, Int32, Color, Color)

Creates a new simple progress bar.

Declaration

```
public ProgressBar(Game game, Widget.DockControl dock, Point offset, float status, int width, int height,
Color progressColor, Color backgroundColor)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the progress bar is to be created.
Widget.DockControl	dock	The location the progress bar should dock onto.

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Point	offset	The space in pixel between the progress bar and the dock location.
System.Single	status	The status of the progress bar, 0 means 0% filled and 1 means 100% filled.
System.Int32	width	The width of the progress bar in pixel. This have to be greater than zero.
System.Int32	height	The height of the progress bar in pixel. This have to be greater than zero.
Microsoft.Xna.Framework.Color	progressColor	The color of the progress bar. (Foreground)
Microsoft.Xna.Framework.Color	backgroundColor	The color behind the progress bar. (Background)

Properties

ProgressColor

The color of the progress bar. This does not affect the background color.

Declaration

```
public Color ProgressColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Color	

Status

The status of the progress bar, 0 means 0% filled and 1 means 100% filled.

Declaration

```
public override float Status { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Overrides

StatusBar.Status

Methods

Draw(SpriteBatch, GameTime)

Draws the progress bar.

Declaration

```
public override void Draw(in SpriteBatch spriteBatch, in GameTime gameTime)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.SpriteBatch	spriteBatch	The sprite batch that draws the texture.
Microsoft.Xna.Framework.GameTime	gameTime	The game time for the elapsed time since the last update call.

Overrides

Widget.Draw(SpriteBatch, GameTime)

Class RadioButton

Class for GUI radio button.

Inheritance

System.Object
GUI
Widget
RadioButton

Inherited Members

Widget.IsActive
Widget.IsVisible
Widget.Color
Widget.SelectionColor
Widget.IsSelected
Widget.IsPressed
Widget.OnClick
Widget.Position
Widget.Tooltip
Widget.Dock
Widget.Offset
Widget.Layer
Widget.Scale
Widget.Texture
Widget.ToggleIsActive()
Widget.ToggleIsVisible()
Widget.CalculatePosition(Widget.DockControl, Point, Texture2D, Single)
Widget.SetPosition(Widget.DockControl, Point)
Widget.IsTextureSelected(Texture2D, Vector2, Single)
Widget.IsTexturePressed(Texture2D, Vector2, Single)
Widget.ResetPressState()
GUI.DrawAll
GUI.Tag
GUI.Display<T>()
GUI.RemoveElement()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: MonoUI
Assembly: MonoUI.dll

Syntax

```
public class RadioButton : Widget, IDrawable
```

Constructors

RadioButton(Game, Widget.DockControl, Point, Texture2D, Texture2D, List<RadioButton>, Single)

Creates a new radio button.

Declaration

```
public RadioButton(Game game, Widget.DockControl dock, Point offset, Texture2D textureOff, Texture2D
textureOn, List<RadioButton> group, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the control element is to be created.
Widget.DockControl	dock	The location the control element should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the control element and the dock location.
Microsoft.Xna.Framework.Graphics.Texture2D	textureOff	The texture for the of/not checked state.
Microsoft.Xna.Framework.Graphics.Texture2D	textureOn	The texture for the on/checked state.
System.Collections.Generic.List<RadioButton>	group	A list of this type to group radio buttons together.
System.Single	scale	The value for scaling the texture. 1.0 is no scaling.

Properties

State

The current On/Off state of the radio button.

Declaration

```
public bool State { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

Draw(SpriteBatch, GameTime)

Draws the radio button.

Declaration

```
public override void Draw(in SpriteBatch spriteBatch, in GameTime gameTime)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.SpriteBatch	spriteBatch	The sprite batch that draws the texture.
Microsoft.Xna.Framework.GameTime	gameTime	The game time for the elapsed time since the last update call.

Overrides

Widget.Draw(SpriteBatch, GameTime)

Class Slider

Class for GUI slider.

Inheritance

System.Object
GUI
Widget
StatusBar
Slider

Inherited Members

StatusBar.StatusTexture
StatusBar.StatusBarSize()
Widget.IsActive
Widget.IsVisible
Widget.Color
Widget.SelectionColor
Widget.IsSelected
Widget.IsPressed
Widget.OnClick
Widget.Position
Widget.Tooltip
Widget.Dock
Widget.Offset
Widget.Layer
Widget.Scale
Widget.Texture
Widget.ToggleIsActive()
Widget.ToggleIsVisible()
Widget.CalculatePosition(Widget.DockControl, Point, Texture2D, Single)
Widget.IsTextureSelected(Texture2D, Vector2, Single)
Widget.IsTexturePressed(Texture2D, Vector2, Single)
Widget.ResetPressState()
GUI.DrawAll
GUI.Tag
GUI.Display<T>()
GUI.RemoveElement()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public class Slider : StatusBar, IDrawable
```

Constructors

Slider(Game, Widget.DockControl, Point, Single, Single, Single, Texture2D, Texture2D, Texture2D, Single)

Creates a new slider with textures.

Declaration

```
public Slider(Game game, Widget.DockControl dock, Point offset, float minValue, float maxValue, float value,
Texture2D textureBackground, Texture2D textureStaus, Texture2D textureHandle, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the slider is to be created.
Widget.DockControl	dock	The location the slider should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the slider and the dock location.
System.Single	minValue	The minimum value of the slider. (Inclusive)
System.Single	maxValue	The maximum value of the slider. (Inclusive)
System.Single	value	The current value of the slider. This value is clamped between minValue and maxValue.
Microsoft.Xna.Framework.Graphics.Texture2D	textureBackground	The texture for the background bar.
Microsoft.Xna.Framework.Graphics.Texture2D	textureStaus	The texture for the foreground bar.
Microsoft.Xna.Framework.Graphics.Texture2D	textureHandle	The texture for the slider handle.
System.Single	scale	The value for scaling the textures. 1.0 is no scaling.

Slider(Game, Widget.DockControl, Point, Single, Single, Single, Int32, Int32, Texture2D, Single)

Creates a new simple slider.

Declaration

```
public Slider(Game game, Widget.DockControl dock, Point offset, float minValue, float maxValue, float value,
int width, int height, Texture2D textureHandle, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the slider is to be created.
Widget.DockControl	dock	The location the slider should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the slider and the dock location.
System.Single	minValue	The minimum value of the slider. (Inclusive)
System.Single	maxValue	The maximum value of the slider. (Inclusive)
System.Single	value	The current value of the slider. This value is clamped between minValue and maxValue.
System.Int32	width	The width of the slider bar in pixel. This have to be greater than zero.
System.Int32	height	The height of the slider bar in pixel. This have to be greater than zero.
Microsoft.Xna.Framework.Graphics.Texture2D	textureHandle	The texture for the slider handle.
System.Single	scale	The scale of the handle texture. 1.0 is no scaling.

Properties

Alignment

True is horizontal alignment, false is vertical alignment.

Declaration

```
public override bool Alignment { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Overrides

StatusBar.Alignment

BackgroundColor

The color of the background bar.

Declaration

```
public Color BackgroundColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Color	

MaxValue

The maximum value of the slider. (Inclusive)

Declaration

```
public float MaxValue { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

MinValue

The minimum value of the slider. (Inclusive)

Declaration

```
public float MinValue { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Status

The status of the slider, 0 means 0% filled and 1 means 100% filled.

Declaration

```
public override float Status { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Overrides

StatusBar.Status

StatusColor

The color of the status bar.

Declaration

```
public Color StatusColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Color	

Step

The step between each value when using the handle. No value results in no steps.

Declaration

```
public float? Step { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Single>	

Value

The current value of the slider. This value is clamped between MinValue and MaxValue. (Inclusive)

Declaration

```
public float Value { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Draw(SpriteBatch, GameTime)

Draws the slider.

Declaration

```
public override void Draw(in SpriteBatch spriteBatch, in GameTime gameTime)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.SpriteBatch	spriteBatch	The sprite batch that draws the texture.
Microsoft.Xna.Framework.GameTime	gameTime	The game time for the elapsed time since the last update call.

Overrides

Widget.Draw(SpriteBatch, GameTime)

SetPosition(Widget.DockControl, Point)

Sets the position of the control element.

Declaration

```
public override void SetPosition(Widget.DockControl dock, Point offset)
```

Parameters

TYPE	NAME	DESCRIPTION
Widget.DockControl	dock	The location the control element should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the control element and the dock location.

Overrides

Widget.SetPosition(Widget.DockControl, Point)

Class StatusBar

Base class for status bars. Only for inheritance.

Inheritance

System.Object
GUI
Widget
StatusBar
ProgressBar
Slider

Inherited Members

Widget.IsActive
Widget.IsVisible
Widget.Color
Widget.SelectionColor
Widget.IsSelected
Widget.IsPressed
Widget.OnClick
Widget.Position
Widget.Tooltip
Widget.Dock
Widget.Offset
Widget.Layer
Widget.Scale
Widget.Texture
Widget.ToggleIsActive()
Widget.ToggleIsVisible()
Widget.CalculatePosition(Widget.DockControl, Point, Texture2D, Single)
Widget.SetPosition(Widget.DockControl, Point)
Widget.IsTextureSelected(Texture2D, Vector2, Single)
Widget.IsTexturePressed(Texture2D, Vector2, Single)
Widget.ResetPressState()
Widget.Draw(SpriteBatch, GameTime)
GUI.DrawAll
GUI.Tag
GUI.Display<T>()
GUI.RemoveElement()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: MonoUI
Assembly: MonoUI.dll

Syntax

```
public abstract class StatusBar : Widget, IDrawable
```

Constructors

StatusBar(Game, Widget.DockControl, Point, Single)

Base constructor for status bars. Only for inheritance.

Declaration

```
protected StatusBar(Game game, Widget.DockControl dock, Point offset, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the control element is to be created.
Widget.DockControl	dock	The location the control element should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the control element and the dock location.
System.Single	scale	The value for scaling the texture. 1.0 is no scaling.

Properties

Alignment

True is horizontal alignment, false is vertical alignment.

Declaration

```
public virtual bool Alignment { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Status

The status of the status bar, 0 means 0% filled and 1 means 100% filled.

Declaration

```
public virtual float Status { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

StatusTexture

The texture that represents the status.

Declaration

```
protected Texture2D StatusTexture { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Graphics.Texture2D	

Methods

StatusBarSize()

Returns the rectangle for the status bar.

Declaration

```
protected Rectangle StatusBarSize()
```

Returns

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Rectangle	

Class TextBox

Class for GUI text box.

Inheritance

System.Object

GUI

Widget

TextBox

Inherited Members

Widget.IsActive

Widget.IsVisible

Widget.Color

Widget.SelectionColor

Widget.IsSelected

Widget.IsPressed

Widget.OnClick

Widget.Position

Widget.Tooltip

Widget.Dock

Widget.Offset

Widget.Layer

Widget.Scale

Widget.Texture

Widget.ToggleIsActive()

Widget.ToggleIsVisible()

Widget.CalculatePosition(Widget.DockControl, Point, Texture2D, Single)

Widget.SetPosition(Widget.DockControl, Point)

Widget.IsTextureSelected(Texture2D, Vector2, Single)

Widget.IsTexturePressed(Texture2D, Vector2, Single)

Widget.ResetPressState()

GUI.DrawAll

GUI.Tag

GUI.Display<T>()

GUI.RemoveElement()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public class TextBox : Widget, IDrawable
```

Constructors

TextBox(Game, Widget.DockControl, Point, String, Color, SpriteFont, Texture2D, Single)

Creates a new single line text box with a texture.

Declaration

```
public TextBox(Game game, Widget.DockControl dock, Point offset, string text, Color textColor, SpriteFont
font, Texture2D texture, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the text box is to be created.
Widget.DockControl	dock	The location the text box should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the text box and the dock location.
System.String	text	The initial text that is to be displayed in the text box.
Microsoft.Xna.Framework.Color	textColor	The color of the text.
Microsoft.Xna.Framework.Graphics.SpriteFont	font	The font of the text. The font size should match with the measurements.
Microsoft.Xna.Framework.Graphics.Texture2D	texture	The texture of the text box.
System.Single	scale	The value for scaling the texture. 1.0 is no scaling.

TextBox(Game, Widget.DockControl, Point, String, Color, SpriteFont, Int32, Int32)

Creates a new simple single line text box.

Declaration

```
public TextBox(Game game, Widget.DockControl dock, Point offset, string text, Color textColor, SpriteFont
font, int width = 200, int height = 25)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the text box is to be created.
Widget.DockControl	dock	The location the text box should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the text box and the dock location.

TYPE	NAME	DESCRIPTION
System.String	text	The initial text that is to be displayed in the text box.
Microsoft.Xna.Framework.Color	textColor	The color of the text.
Microsoft.Xna.Framework.Graphics.SpriteFont	font	The font of the text. The font size should match with the measurements.
System.Int32	width	The width of the text box in pixel. This have to be greater than zero.
System.Int32	height	The height of the text box in pixel. This have to be greater than zero.

Properties

Indentation

The empty space in pixel on the left side between the texture and the text.

Declaration

```
public ushort Indentation { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.UInt16	

MaxCharacters

The maximum number of characters the text can contain.

Declaration

```
public ushort MaxCharacters { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.UInt16	

OverFlow

Determines if the text can reach over the text field.

Declaration

```
public bool OverFlow { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Text

The text inside the text box.

Declaration

```
public string Text { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

TextColor

The color of the text.

Declaration

```
public Color TextColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Color	

Methods

Draw(SpriteBatch, GameTime)

Draws the the text box.

Declaration

```
public override void Draw(in SpriteBatch spriteBatch, in GameTime gameTime)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.SpriteBatch	spriteBatch	The sprite batch that draws the texture.
Microsoft.Xna.Framework.GameTime	gameTime	The game time for the elapsed time since the last update call.

Overrides

Widget.Draw(SpriteBatch, GameTime)

Class Tool

The utility class of the GUI system.

Inheritance

System.Object
Tool

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public static class Tool
```

Methods

CreateTexture(GraphicsDevice, Int32, Int32, Color)

Creates a new single color texture.

Declaration

```
public static Texture2D CreateTexture(in GraphicsDevice graphicsDevice, int width, int height, Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.GraphicsDevice	graphicsDevice	The graphics device that is going to draw the texture.
System.Int32	width	The width of the new texture in pixel. This have to be greater than zero.
System.Int32	height	The height of the new texture in pixel. This have to be greater than zero.
Microsoft.Xna.Framework.Color	color	The color of the new texture.

Returns

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Graphics.Texture2D	

Class Tooltip

Class for GUI tooltip.

Inheritance

System.Object
GUI
Tooltip

Inherited Members

GUI.DrawAll
GUI.Tag
GUI.Display<T>()
GUI.RemoveElement()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public class Tooltip : GUI
```

Constructors

Tooltip(Game, Texture2D, Single)

Creates a picture tooltip that displays additional informations in combination with a control element.

Declaration

```
public Tooltip(Game game, Texture2D picture, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the tooltip is to be created.
Microsoft.Xna.Framework.Graphics.Texture2D	picture	The picture that is to be displayed within the tooltip.
System.Single	scale	The scale of the picture.

Tooltip(Game, String, Color, SpriteFont, Color)

Creates a text tooltip that displays additional informations in combination with a control element.

Declaration

```
public Tooltip(Game game, string text, Color textColor, SpriteFont font, Color backgroundColor)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the tooltip is to be created.
System.String	text	The text that is to be displayed within the tooltip.
Microsoft.Xna.Framework.Color	textColor	The color of the text.
Microsoft.Xna.Framework.Graphics.SpriteFont	font	The font of the text.
Microsoft.Xna.Framework.Color	backgroundColor	The color of the text background.

Properties

BackgroundColor

The color of the background texture.

Declaration

```
public Color BackgroundColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Color	

FadeInTime

The time in milliseconds the fade in effect lasts.

Declaration

```
public int FadeInTime { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

MouseCursorOffset

The space in pixel between the tooltip and the active mouse cursor.

Declaration

```
public static Point MouseCursorOffset { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Point	

ShowTime

The time in milliseconds that the control have to be selected before the tooltip is shown.

Declaration

```
public int ShowTime { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

TextColor

The color of the text.

Declaration

```
public Color TextColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Color	

Methods

Activate(SpriteBatch, GameTime, Boolean)

Determines the current status of the tooltip.

Declaration

```
public void Activate(SpriteBatch spriteBatch, GameTime gameTime, bool isSelected)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.SpriteBatch	spriteBatch	The sprite batch that draws the texture.
Microsoft.Xna.Framework.GameTime	gameTime	The game time for the elapsed time since the last update call.
System.Boolean	isSelected	The isSelected state of the connected control element.

SetPicture(Texture2D, Single)

Sets the picture of the tooltip. (Doesn't work in combination with text.)

Declaration

```
public void SetPicture(Texture2D picture, float scale)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.Texture2D	picture	The picture that is to be displayed within the tooltip.
System.Single	scale	The scale of the picture.

SetText(String)

Sets the text of the tooltip. (Doesn't work in combination with a picture.)

Declaration

```
public void SetText(string text)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	text	The text that is to be displayed within the tooltip.

Class Widget

Base class for all control elements. Only for inheritance.

Inheritance

System.Object
GUI
Widget
Button
Checkbox
DropDownList
Image
Label
RadioButton
StatusBar
TextBox

Inherited Members

GUI.DrawAll
GUI.Tag
GUI.Display<T>()
GUI.RemoveElement()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public abstract class Widget : GUI, IDrawable
```

Constructors

Widget(Game, Widget.DockControl, Point, Single)

Base constructor for all control elements. Only for inheritance.

Declaration

```
protected Widget(Game game, Widget.DockControl dock, Point offset, float scale = 1F)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Game	game	The game instance in that the control element is to be created.
Widget.DockControl	dock	The location the control element should dock onto.

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Point	offset	The space in pixel between the control element and the dock location.
System.Single	scale	The value for scaling the texture. 1.0 is no scaling.

Properties

Color

The color of the texture of the control element.

Declaration

```
public Color Color { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Color	

Dock

The location the control element should dock onto.

Declaration

```
protected Widget.DockControl Dock { get; set; }
```

Property Value

TYPE	DESCRIPTION
Widget.DockControl	

IsActive

Defines if the user can interact with the control element.

Declaration

```
public bool IsActive { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

IsPressed

True if the control element is pressed, otherwise false;

Declaration

```
public bool IsPressed { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

IsSelected

True if the control element is selected, otherwise false.

Declaration

```
public bool IsSelected { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

IsVisible

Defines if the control element is visible. An invisible control element is always inactive.

Declaration

```
public bool IsVisible { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Layer

The layer on that the texture is drawn.

Declaration

```
protected float Layer { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Offset

The space in pixel between the control element and the dock location.

Declaration

```
protected Point Offset { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Point	

Position

The absolut position of the control element.

Declaration

```
public Vector2 Position { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Vector2	

Scale

The scale of the texture.

Declaration

```
protected float Scale { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

SelectionColor

The color the control element receives when it gets selected. No value results in no color change.

Declaration

```
public Color? SelectionColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<Microsoft.Xna.Framework.Color>	

Texture

The texture of the control element.

Declaration

```
protected Texture2D Texture { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Graphics.Texture2D	

Tooltip

The tooltip that is to be displayed when the control element gets selected.

Declaration

```
public Tooltip Tooltip { get; set; }
```

Property Value

TYPE	DESCRIPTION
Tooltip	

Methods

CalculatePosition(Widget.DockControl, Point, Texture2D, Single)

Gets the relative position for the control element.

Declaration

```
protected Vector2 CalculatePosition(Widget.DockControl dock, Point offset, Texture2D texture, float scale)
```

Parameters

TYPE	NAME	DESCRIPTION
Widget.DockControl	dock	The location the control element should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the control element and the dock location.
Microsoft.Xna.Framework.Graphics.Texture2D	texture	The texture that determines the positioning.
System.Single	scale	The scale of the texture.

Returns

TYPE	DESCRIPTION
Microsoft.Xna.Framework.Vector2	

Draw(SpriteBatch, GameTime)

Draws the the control element.

Declaration

```
public virtual void Draw(in SpriteBatch spriteBatch, in GameTime gameTime)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.SpriteBatch	spriteBatch	The sprite batch that draws the texture.
Microsoft.Xna.Framework.GameTime	gameTime	The game time for the elapsed time since the last update call.

IsTexturePressed(Texture2D, Vector2, Single)

Checks if a texture at a specific position is pressed.

Declaration

```
protected bool IsTexturePressed(Texture2D texture, Vector2 position, float scale)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.Texture2D	texture	The texture that is to be checked.
Microsoft.Xna.Framework.Vector2	position	The absolute position of the texture.
System.Single	scale	The scale of the texture. 1 is no scaling.

Returns

TYPE	DESCRIPTION
System.Boolean	

IsTextureSelected(Texture2D, Vector2, Single)

Checks if a texture at a specific position is selected.

Declaration

```
protected bool IsTextureSelected(Texture2D texture, Vector2 position, float scale)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Xna.Framework.Graphics.Texture2D	texture	The texture that is to be checked.
Microsoft.Xna.Framework.Vector2	position	The absolute position of the texture.
System.Single	scale	The scale of the texture. 1 is no scaling.

Returns

TYPE	DESCRIPTION
System.Boolean	

ResetPressState()

Resets the press state. Use this if a control element has more than one pressable sections.

Declaration

```
protected void ResetPressState()
```

SetPosition(Widget.DockControl, Point)

Sets the position of the control element.

Declaration

```
public virtual void SetPosition(Widget.DockControl dock, Point offset)
```

Parameters

TYPE	NAME	DESCRIPTION
Widget.DockControl	dock	The location the control element should dock onto.
Microsoft.Xna.Framework.Point	offset	The space in pixel between the control element and the dock location.

ToggleIsActive()

Toggles the IsActive state of the control element.

Declaration

```
public void ToggleIsActive()
```

ToggleIsVisible()

Toggles the IsVisible state of the control element.

Declaration

```
public void ToggleIsVisible()
```

Events

OnClick

Raises an event during the frame the control element is clicked.

Declaration

```
public event Widget.EventHandler OnClick
```

Event Type

TYPE	DESCRIPTION
Widget.EventHandler	

Enum Widget.DockControl

Defines where the control element should dock onto the screen.

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public enum DockControl
```

Fields

NAME	DESCRIPTION
Center	The center of the screen.
CornerBottomLeft	The corner at the bottom left of the screen.
CornerBottomRight	The corner at the bottom right of the screen.
CornerTopLeft	The corner at the top left of the screen.
CornerTopRight	The corner at the top right of the screen.
MiddleBottom	The middle at the bottom side of the screen.
MiddleLeft	The middle at the left side of the screen.
MiddleRight	The middle at the right side of the screen.
MiddleTop	The middle at the top side of the screen.

Delegate Widget.EventHandler

Delegate for managing events.

Namespace: MonoUI

Assembly: MonoUI.dll

Syntax

```
public delegate void EventHandler();
```