London Metropolitan University

UI Elements library Documentation

Aiste Simonaityte

Contents

[Button 3](#_Toc136030171)

[Checkbox 3](#_Toc136030172)

[Modal Window 4](#_Toc136030173)

[Progress Bar 4](#_Toc136030174)

[Slider 5](#_Toc136030175)

[Toggle Button 5](#_Toc136030176)

# Button

* void SetButtonRect(float rotationAngle = 0.0f) – creates/updates the collision bounding box for the button object
* SDL\_Point GetMousePosition()– returns the x and y of the mouse’s position
* void SetNormalAttributes(const std::string& filename, const SDL\_Point& spriteDimensions) – sets the attributes for the normal button state. Filename is the pathway to the image, spriteDimensions sets the desired size of the image on screen.
* void SetHoveredAttributes(const std::string& filename, const SDL\_Point& spriteDimensions) - sets the attributes for the hovered button state. Filename is the pathway to the image, spriteDimensions sets the desired size of the image on screen.
* virtual bool Update() override – inherited from its GameObject base class. Constantly updates and checks the current state of the button.
* virtual bool Render() override – inherited from its GameObject base class. Renders the appropriate image based on the current state of the button.

## How to create a button:

* Include the “Button.h” header file.
* Instantiate the button by using Button nameOfButton.
* Call Render() and Update() functions.

# Checkbox

* void SetCheckedAttributes(const std::string& filename, const SDL\_Point& spriteDimensions) – sets the attributes for the checked checkbox state. Filename is the pathway to the image, spriteDimensions sets the desired size of the image on screen.
* void SetUncheckedAttributes(const std::string& filename, const SDL\_Point& spriteDimensions) - sets the attributes for the unchecked checkbox state. Filename is the pathway to the image, spriteDimensions sets the desired size of the image on screen.
* bool OnChecked() – returns true when the checkbox is ticked.
* bool OnUnchecked() – returns true when the checkbox is unticked.
* virtual bool Update() override – inherited from its GameObject base class. Constantly updates and checks the current state of the checkbox.
* virtual bool Render() override – inherited from its GameObject base class. Renders the appropriate image based on the current state of the checkbox.

## How to create a checkbox:

* Include the “Checkbox.h” header file.
* Instantiate the checkbox by using Checkbox nameOfCheckbox.
* Call Render() and Update() functions.

# Modal Window

* void SetBackgroundAttributes(const std::string& filename, const SDL\_Point& spriteDimensions) – sets the attributes for the window image. Filename is the pathway to the image, spriteDimensions sets the desired size of the image on screen.
* void SetButtonAttributes(const std::string& filename, const SDL\_Point& spriteDimensions) – sets the attributes for the button image. Filename is the pathway to the image, spriteDimensions sets the desired size of the image on screen.
* void SetOpen() – sets m\_isOpen to true.
* void SetClose() – sets m\_isOpen to false.
* void HandleEvent(const SDL\_Event& event) – constantly checks the collision between the button and mouse. Closes the window if the button is pressed.
* virtual bool Render() override - inherited from its GameObject base class. Renders the modal window based on the state of m\_isOpen variable.

## How to create a modal window:

* Include the “ModalWindow.h” header file.
* Instantiate the modal window by using ModalWindow nameOfModalWindow.
* Call HandleEvent() and Update() functions.

# Progress Bar

* void SetBackgroundAttributes(const std::string& filename, const SDL\_Point& spriteDimension) – sets the attributes to the background image of the progress bar. Filename is the pathway to the image, spriteDimensions sets the desired size of the image on screen.
* void SetFillAttributes(const std::string& filename, const SDL\_Point& spriteDimension) – sets the attributes to the fill image of the progress bar. Filename is the pathway to the image, spriteDimensions sets the desired size of the image on screen.
* void SetValue(int value) – sets a new value to the m\_currentValue variable, which controls the fill image.
* virtual bool Render() override – inherited from its GameObject base class. Renders the background image and the fill image, depending on the m\_currentValue variable.

## How to create a progress bar:

* Include the “ProgressBar.h” header file.
* Instantiate the progress bar by using ProgressBar nameOfProgressBar.
* Call Render() and SetValue() functions.

# Slider

* void SetBackgroundAttributes(const std::string& filename, const SDL\_Point& dimensions) – sets the attributes to the background image of the slider. Filename is the pathway to the image, spriteDimensions sets the desired size of the image on screen.
* void SetSliderAttributes(const std::string& filename, const SDL\_Point& dimensions) – sets the attributes to the handle image of the slider. Filename is the pathway to the image, spriteDimensions sets the desired size of the image on screen.
* void SetValue(int newValue) – sets the m\_currentValue variable to a new one.
* int GetValue() const – returns the m\_currentValue.
* void HandleEvent(const SDL\_Event& event) – checks the collision between the bounding box of the handle, background and mouse.
* virtual bool Render() override – renders the background image and the handle image.

## How to create a slider:

* Include the “Slider.h” header file.
* Instantiate the checkbox by using Slider nameOfSlider.
* Call Render() and HandleEvent() functions.

# Toggle Button

* void SetOnAttributes(const std::string& filename, const SDL\_Point& spriteDimensions) – sets the attributes to the ON image of the toggle button. Filename is the pathway to the image, spriteDimensions sets the desired size of the image on screen.
* void SetOffAttributes(const std::string& filename, const SDL\_Point& spriteDimensions) – sets the attributes to the OFF image of the toggle button. Filename is the pathway to the image, spriteDimensions sets the desired size of the image on screen.
* void SetState(bool state) – sets the m\_state variable which controls whether the toggle is on or off.
* bool GetToggleState() const – returns the m\_state variable.
* virtual bool Render() override – depending on the m\_state variable, renders the appropriate image.

## How to create a toggle button:

* Include the “ToggleButton.h” header file.
* Instantiate the checkbox by using ToggleButton nameOfToggleButton.
* Call Render() and HandleEvent() functions.