Документация к библиотеке SKBGUI

Версия библиотеки: 0.0.10.1
Дата обновления: 23 мая 2018 г., 05:37
С версии 0.0.10.0 поддерживается русский язык (юникод, UTF-8)

Оглавление

Общие принципы для всех компонентов

Каждый компонент состоит из модели (Model), вьювера (Viewer) и контроллера (Controller).
Смысл каждого из элементов:

- Model - хранит в себе все данные компонета. Их изменение также происходит через model.
- Viewer - рисует компонент.
- Controller - обрабатывает события.

Модель

Свойства модели:

- имя

- абсолютные координаты
- локальные координаты
- размеры
- предок
- список потомков
- состояние
- фокус
- видимость границ
- доступность
- видимость
- указатель на сам компонент

Методы модели

Объявление функции	Пояснение
string Name()	Вовзращает имя компонента
void SetName(**string** StName)	Устанавливает имя компонента
Vector2f LocalCoord()	Вовзращает локальные координаты
void SetLocalCoord(**double** x,**double** y)	Устанавливает локальные координаты
void SetLocalCoord(**Vector2f** ComCoord)	Устанавливает локальные координаты
Vector2f Size()	Вовзращает размеры
void SetSize(**double** x,**double** y)	Установить размеры
void SetSize(**Vector2f** ComSize)	Установить размеры
void SetState(**int** state)	Установить состояние
int State()	Возвращает состояние
void SetFocus(**bool** focus)	Установить фокус
bool Focus()	Возвращает фокус
void SetParent(**pComponentModel** parrent)	Указать предка для компонента
pComponentModel Parent()	Вовзращает предка или nullptr если его нет
void Add(**pComponent** child)	Добавить потомка
std::string Info()	Возвращает информацию о компоненте

pComponent Find(**string** StName)	Возвращает адрес компонента если находит иначе nullptr
pComponent FindAll(**std::string** name)	Аналогично предыдущему, но поиск рекурсивный среди всех потомков
void Delete(**pComponent** children)	Удаляет потомка с указаным адресом
void Delete(**string** ChName)	Удаляет потомка с указаным именем
pComponent Children(**int** index)	Возвращает потомка с номером index
void SetChildren(**int** index, **pComponent** component)	Указывает нового потомка с номером index
int Count()	Возвращает количество потомков
bool VisibleBorders()	Возвращает значение - “отображать границы”
void SetVisibleBorders(**bool** visibleBorders)	Установить - “отображать границы”
Vector2f AbsoluteCoord()	Возвращает абсолютные координаты
void SetAbsoluteCoord(**Vector2f** Abs)	Устанавливает абсолютные координаты
void SetEnabled(**bool** enabled)	Устанавливает доступность
bool Enabled()	Возвращает доступность
bool Visible()	возвращает видисомть компонента
void SetVisible(**bool** visible)	устанавливает видимость компонента
void Show()	показывает объект
void Hide()	скрывает объект
pComponent Owner();	возвращает указатель на свой компонент
void SetOwner(**pComponent** owner);	устанавливает указатель на свой компонент

Вьювер

Как пользователи, с ним вы не будете сталкиваться

Контроллер

Список событий контроллера

Событие	Описание
MouseDown	клавиша мыши была нажата
MouseUp	клавиша мыши была отпущена
MouseMove	курсор мыши был перемещен
MouseWheel	колесико мыши было прокручено
KeyDown	нажата клавиша клавиатуры
KeyUp	отпущена клавиша клавиатуры
Click	нажатие на компонент курсором мыши
Idle	срабатывает на каждый кадр

Описание типов обработчиков:

```
typedef void (*simpleCallback)(pComponentModel model)
```

Используется для событий:

- Click
- Idle
- MouseMove

```
typedef void (*MouseButtonCallback)(pComponentModel model, int x, int y, int button)
```

Используется для событий:

- MouseDown
- MouseUp

```
typedef void (*MouseWheelCallback)(pComponentModel model, int x, int y, int delta)
```

Используется для событий:

- MouseWheel

```
typedef void (*keyboardCallback)(pComponentModel model, int key)
```

Используется для событий:

- KeyDown
- KeyUp

Методы контроллера

Метод	Описание
bool Handle(**sf::Event** event);	занимается обработкой событий
void Update(**sf::Time** time);	занимается обработкой времени

Метод	Описание
void SetMouseDown(MouseButtonCallback func)	Установить одноименный обработчик
void SetMouseUp(MouseButtonCallback func)	Установить одноименный обработчик
void SetMouseMove(simpleCallback func)	Установить одноименный обработчик
void SetMouseWheel(MouseWheelCallback func)	Установить одноименный обработчик
void SetKeyDown(keyboardCallback func)	Установить одноименный обработчик
void SetKeyUp(keyboardCallback func)	Установить одноименный обработчик
void SetClick(simpleCallback func)	Установить одноименный обработчик
void SetCreated(simpleCallback func)	Установить одноименный обработчик
void SetDestroy(simpleCallback func)	Установить одноименный обработчик
void SetPaint(simpleCallback func)	Установить одноименный обработчик
void SetIdle(simpleCallback func)	Установить одноименный обработчик

Метод	Описание
MouseButtonCallback MouseDown()	Возвращает указатель на одноименный обработчик
MouseButtonCallback MouseUp()	Возвращает указатель на одноименный обработчик
simpleCallback MouseMove()	Возвращает указатель на одноименный обработчик
MouseWheelCallback MouseWheel()	Возвращает указатель на одноименный обработчик

keyboardCallback KeyDown()	Возвращает указатель на одноименный обработчик
keyboardCallback KeyUp()	Возвращает указатель на одноименный обработчик
simpleCallback Click()	Возвращает указатель на одноименный обработчик
simpleCallback Created()	Возвращает указатель на одноименный обработчик
simpleCallback Destroy()	Возвращает указатель на одноименный обработчик
simpleCallback Paint()	Возвращает указатель на одноименный обработчик
simpleCallback Idle()	Возвращает указатель на одноименный обработчик

Компоненты

Label

Модель

Label - текстовая строка.

Свойство	Описание
std::string caption	выводимый текст
sf::Color color	цвет текста

Метод	Описание
std::string Caption()	Возвращает значание одноименного свойства
void SetCaption(**std::string** caption)	Установить значение одноименного свойства
sf::Color Color()	Возвращает значание одноименного свойства
void SetColor(**sf::Color** color)	Установить значение одноименного свойства

Edit

Edit - однострочное поле ввода.

Модель

Свойство	Описание
std::string text	текст в поле ввода
int caretPos	положение каретки
sf::Color fillColor	цвет фона поля
sf::Color textColor	цвет текста

Метод	Описание
std::string Text()	Возвращает значание одноименного свойства
int CaretPos()	Возвращает значание одноименного свойства
sf::Color FillColor()	Возвращает значание одноименного свойства
sf::Color TextColor()	Возвращает значание одноименного свойства
void SetText(**std::string** text)	Установить значение одноименного свойства
void SetCaretPos(**int** pos)	Установить значение одноименного свойства
void SetFillColor(**sf::Color** color)	Установить значение одноименного свойства
void SetTextColor(**sf::Color** color)	Установить значение одноименного свойства

Button

Button - кнопка.

Модель

Свойство	Описание
std::string caption	Надпись на кнопке

Метод	Описание
std::string Caption()	Возвращает значание одноименного свойства

void SetCaption(std::string caption)	Установить значение одноименного свойства

Panel

Panel - это тупо прямоугольная область, предназначеная для группировки компонентов.

Модель

Свойство	Описание
sf::Color color	цвет фона

Метод	Описание
sf::Color Color()	Возвращает значание одноименного свойства
void SetColor(**sf::Color** color)	Установить значение одноименного свойства

Picture

Picture - прямоугольное изображение.
Поддерживает загрузку PNG, TGA, JPEG, BMP.

Модель

Свойство	Описание
sf::Texture* texture	текстура (загруженно изображение)

Метод	Описание
sf::Texture* Texture()	Возвращает значание одноименного свойства
void SetTexture(**sf::Texture*** texture)	Установить значение одноименного свойства
void LoadFromFile(**std::string** filename)	загружает картинку из файла filename

Table

Table - таблица, состоит из множества Edit.

Модель

Свойство	Описание
int rowCount	количество строк
int colCount	количество столбцов
std::vector rowHeight	список высот строк
std::vector colWidth	список ширин стобцов
std::vector < std::vector < pEdit >> cell	матрица полей ввода

Метод	Описание
void SetRowCount(**int** rowCount)	Установить значение одноименного свойства
void SetColCount(**int** colCount)	Установить значение одноименного свойства
int RowCount()	Возвращает значание одноименного свойства
int ColCount()	Возвращает значание одноименного свойства
int RowHeight(**int** index)	Возвращает значание одноименного свойства
int ColWidth(**int** index)	Возвращает значание одноименного свойства
void SetRowHeight(**int** row, **int** height)	Установить значение одноименного свойства
void SetColWidth(**int** col, **int** width)	Установить значение одноименного свойства
pEdit Cell(**int** row, **int** col)	Возвращает значание одноименного свойства

Timer

Timer - таймер.

По умолчанию в модели установлено поле *Enabled* равное *false*, поэтому для запуска таймера его надо включить с помощью *SetEnabled(true)*.

Модель

Свойство	Описание
sf::Time interval	Интервал, через который срабатывает таймер

void (*onTimer)()	обработчик на срабатывание таймера

Метод	Описание
sf::Time Interval()	возвращает интервал
void SetInterval(**sf::Time** interval)	установить интервал
void SetOnTimer(**void (*func)()**)	установить обработчик

CheckBox

Флажок, флаговая кнопка, чекбокс (от англ. check box), галочка — элемент графического пользовательского интерфейса, позволяющий пользователю управлять параметром с двумя состояниями — [x] включено и [ ] выключено

Модель

Метод	Описание
bool Checked();	возвращает состояние флажка
void SetChecked(**bool** checked);	установить состояние флажка
std::string Caption();	возвращает строку надписи флажка
void SetCaption(**std::string** caption);	установить надпись для флажка

ProgressBar

Элемент (виджет) графического интерфейса пользователя, который представляет собой прямоугольную область, которая «заполняется» областью другого цвета/фактуры по мере выполнения какой-либо задачи, например, загрузки файла. Стандартный индикатор процесса заполняется слева направо.

Модель

Метод	Описание
bool Vertical();	Возвращает расположение ProgressBar'a, true - вертикально, false - горизонтально
void SetVertical(**bool** vertical);	установить расположение

double Max();	максимальное значение
double Min();	минимальное значение
double Current();	текущее знаение
void SetMax(**double** max);	установить максимальное значение
void SetMin(**double** min);	установить минимальное значение
void SetCurrent(**double** current);	установить текущее знаение