# UML - UI SYSTEM

Víctor Masó Garcia
Guillermo García Subirana

## j1App

+ ...
---
+ Update
+ ...
+ CleanUp

## j1Module

+ ...
---
+ Update
+ ...
+ CleanUp

1

## Assumptions

We assume **j1Entities** has a parent class called **j1Module** that has all *Awake, Update* and *CleanUp* methods that are called from a **j1App** class that controls all modules.

We assume some basic classes are implemented to help us manage data in a better way. We mean classes such as **iPoint**, **p2SString**, **p2Animation**, **j1Fonts** and others.

We assume both **SDL2** and **pugixml** libraries are included in our code.

## j1UI

- **p2ArrayDyn<UIElement*>** ui_elements;
- **SDL_Texture*** atlas;
- **p2SString** atlas_file_name;
---
+ j1UI();
+ ~j1UI();

+ **bool** Awake(**pugi::xml_node&** conf);
+ **bool** Start();
+ **bool** PreUpdate();
+ **bool** Update(**float** dt);
+ **bool** PostUpdate();
+ **bool** CleanUp();

+ **const SDL_Texture*** GetAtlas() **const**;

+ **bool** DeleteUIElement(**UIElement** &element);
+ **bool** DeleteAllUIElements();

+ **UIElement*** AddUIImage(**int** position_x, **int** position_y, **SDL_Rect** rect);
+ **UIElement*** AddUIButton(**int** position_x, **int** position_y, **SDL_Rect** normal_rect, **SDL_Rect** focused_rect, **SDL_Rect** pressed_rect, **j1Module*** callback = **nullptr**, **UIElement*** parent = **nullptr**);
+ **UIElement*** AddUILabel(**int** position_x, **int** position_y, **p2SString** text, **Color** color = *COLOR_NOT_DEF*);
+ **UIElement*** AddUISlider(**int** position_x, **int** position_y, **SDL_Rect** slider_rect, **j1Module*** callback = **nullptr**, **UIElement*** parent = **nullptr**);
+ **UIElement*** GetElementUnderMouse(**int** x, **int** y);

*

## UIElement

+ **bool** draggable;
+ **bool** interactable;
+ **bool** to_destroy;
+ **UIElement*** parent;
+ **UI_Type** type;
+ **UI_State** current_state;
+ **UI_State** last_state;

# **j1Module*** callback;
# **SDL_Rect** current_rect;
# **iPoint** screen_position;
# **iPoint** local_position;
# **uint** mouse_in_fx;
# **int** mouse_motion_x, mouse_motion_y;
# **int** tmp_mouse_x, tmp_mouse_y;
# **uint** pressed_fx;
# **SDL_Texture*** texture;

- **bool** debug_draw;
- **iPoint** camera_parent_postion;
---
+ **UIElement**(**int** x, **int** y, **UI_Type** type);
+ **virtual** ~UIElement();

+ **virtual void** Update() ();
+ **virtual void** Draw(**SDL_Texture***sprites);

+ **iPoint** GetLocalPosition() **const**;
+ **iPoint** GetScreenPosition() **const**;
+ **SDL_Rect** GetRect() **const**;

+ **void** SetLocalPosition(**int** x, **int** y);
+ **void** SetScreenPosition(**int** x, **int** y);

+ **virtual void** SetParent(**UIElement*** parent);
+ **virtual UIElement*** GetParent() **const**;

## enum UI_State

*STATE_NORMAL,*
*STATE_MOUSE_ENTER,*
*STATE_MOUSE_LEAVE,*
*STATE_LEFT_MOUSE_PRESSED,*
*STATE_LEFT_MOUSE_RELEASED,*
*STATE_RIGHT_MOUSE_PRESSED,*
*STATE_FOCUSED,*

*STATE_NO_DEF*

## enum Color

*RED,*
*GREEN,*
*BLUE,*
*YELLOW,*
*GREY,*
*BLACK,*

*COLOR_NOT_DEF*

## enum UI_Type

*LABEL,*
*BUTTON,*
*IMAGE,*
*SLIDER,*
*CHECK_BOX,*

*UI_NOT_DEF*

## UIButton

+ **SDL_Rect** btn_normal;
+ **SDL_Rect** btn_focused;
+ **SDL_Rect** btn_pressed;
+ **UILabel*** button_lbl;

- **uint** pressed_fx, released_fx , hover_fx ;
---
+ **UIButton**(**int** x, **int** y, **UI_Type** type, **SDL_Rect** normal_rect, **SDL_Rect** focused_rect, **SDL_Rect** pressed_rect, **j1Module*** callback, **UIElement*** parent = **nullptr**);

+ ~ UIButton();
+ **void** UpdateButtonWithSelfRect(**SDL_Rect** self_rect);

## UILabel

- **UIFont** normal_font;
- **p2SString** text;
- **int** size;
- **Color** text_color;
---
+UILabel(**int** x, **int** y, **UI_Type** type, **p2SString** text, **Color** color);
+ **void** SetTextFromNum(**int** value);

## UIImage

- **SDL_Rect** img_normal;
- **SDL_Rect** img_pressed;
- **SDL_Rect** img_mouse_in;
---
+ UIImage (**int** x, **int** y, **UI_Type** type, **SDL_Rect** rect, **j1Module*** callback);
+ **void** UpdateImageWithCoords(**SDL_Rect** rect);

## UISlider

+ **SDL_Rect** slider_rect;

- **uint** slider_value;
- **UIButton*** slider_btn;
---
+ UISlider(**int** x, **int** y, **UI_Type** type, **SDL_Rect** slider_rect, **j1Module*** callback, **UIElement*** parent = nullptr);
+ **void** SetSliderButtons(**UIButton*** slider_btn);
+ **void** SetSliderValue(uint slider_btn_pos);

+ **uint** GetSliderValue() **const**;
+ **UIButton*** GetSliderButton() **const**;

## UICheckBox

+ **SDL_Rect** cb_tick_normal;
+ **SDL_Rect** cb_no_tick_normal;
+ **SDL_Rect** cb_tick_pressed;
+ **SDL_Rect** cb_no_tick_pressed;
+ **SDL_Rect** cb_tick_focus;
+ **SDL_Rect** cb_no_tick_focus;
+ **bool** check_box_state;

- **uint** pressed_fx, released_fx , hover_fx ;
---
+ **UICheckBox**(**int** x, **int** y, **UI_Type** type, **SDL_Rect** cb_tick_normal, **SDL_Rect** cb_no_tick_normal, **SDL_Rect** cb_tick_pressed, **SDL_Rect** cb_no_tick_pressed, **SDL_Rect** cb_tick_focus, **SDL_Rect** cb_no_tick_focus, **j1Module*** callback, **UIElement*** parent = **nullptr**);
+ ~ UICheckBox();
+ **void** UpdateCheckBoxWithSelfRect(**SDL_Rect** self_rect);