j1Module-> class j1Gui

j1Gui:

public:

```
j1Gui()
virtual ~j1Gui()
bool Awake(pugi::xml_node&)
bool Start()
bool PreUpdate()
bool PostUpdate()
bool CleanUp()
bool Update(float)
UIImage* CreateImage(iPoint, SDL_Rect, const SDL_Texture*, j1Module*, bool)
UILabel* CreateLabel(iPoint, char*, SDL_Color, _TTF_Font*, j1Module*, bool, uint length= 100)
UIButton* CreateButton(iPoint, SDL_Rect, SDL_Rect, SDL_Rect, const SDL_Texture*, j1Module*, bool)
UISlider* CreateSlider(iPoint, SDL_Rect, SDL_Rect, const SDL_Texture*, j1Module*,float)
void SortByDrawOrder()
const SDL_Texture* GetAtlas() const
bool DeleteUIElements()
bool needOrderList
```

private:

```
SDL_Texture* atlas
p2SString atlas_file_name
p2List<UIElement*> UiElement
bool drawDebug
```

(para construir elementos de Gui, no tengo claro si hacer que herede de Gui o simplemente le pasamos el modulo Gui y ya, de momento lo he puesto así)

j1Element:

```
enum ElementType{  NoTypeElement, ButtonElement, LabelElement, ImageElement, SliderElement }

enum EventElement{ NoEventElement, MouseEnterEvent, MouseLeaveEvent, MouseRightClickEvent, MouseLeftClickEvent,
        FocusEventElement }
```

public:

```
UIElement(iPoint, j1Module*, bool)
virtual void Update(float)
virtual void Draw()
virtual void DebugDraw() const
virtual void SetSliderButtonPos(int)
void SetParent(UIElement*)
void SetLocalPosition(iPoint)
void ChangeTexture(const SDL_Texture*)
void ChangeTextureRect(SDL_Rect)
virtual ~UIElement()
ElementType type
EventElement eventElement
SDL_Rect rectUi
uint positionToDraw
iPoint positionUi
bool dragable
bool toDelete
iPoint localPosition
```

protected:

```
UIElement* parent
const SDL_Texture *texture
j1Module* listener
iPoint mouse
int mousePositionDragX, mousePositionDragY
```

(Herencia de j1Element)j1Button:

public:

UIButton(iPoint, SDL_Rect, SDL_Rect, SDL_Rect, const SDL_Texture*, ElementType, j1Module*, bool)
SDL_Rect default_texture_rect
SDL_Rect mouse_on_rect
SDL_Rect clicked_rect

(Herencia de j1Element)j1Image:

public:

UIImage(iPoint, SDL_Rect, const SDL_Texture*, ElementType, j1Module*, bool)

(Herencia de j1Element)j1Label:

public:

UILabel(iPoint, const SDL_Texture *, ElementType, j1Module*, bool)

(Herencia de j1Element)j1Slider:

(por si queremos algo que se pueda deslizar un boton como el del volumen)

public:

UISlider(iPoint,SDL_Rect, SDL_Rect, const SDL_Texture*, ElementType, j1Module*,float)
float getValue()
void Draw()
void SetSliderButtonPos(int)
iPoint button_position

private:

SDL_Rect rect_bar
SDL_Rect rect_button
float currentValue (coger el valor en que se deja el boton deslizable)