

j1Module-> class j1Gui

j1Gui:

public:

```
j1Gui()
virtual ~j1Gui()
bool Awake(pugi::xml_node&)
bool Start()
bool PreUpdate()
bool CleanUp()
bool Update(float)
void DrawAll()
UIImage* CreateImage(iPoint, SDL_Rect, bool)
UILabel* CreateLabel(iPoint, p2SString, SDL_Color, _TTF_Font*, bool)
UIButton* CreateButton(iPoint, SDL_Rect[], p2SString, SDL_Color, _TTF_Font*, bool)
UIBoxText* CreateBoxText(iPoint, SDL_Rect, p2SString, Color, TTF_Font*, bool)
UISlider* CreateSlider(iPoint, SDL_Rect, SDL_Rect, bool)
SDL_Texture* GetAtlas() const bool
```

private:

```
SDL_Texture* atlas
p2SString atlas_file_name
p2List<UIElement*>GuiEntities
```

j1Element:

enum ElementType { NoTypeElement, ButtonElement, LabelElement, ImageElement, SliderElement, BoxTextElement };

```
enum EventElement {
    NoEventElement, MouseEnterEvent, MouseInside, MouseLeaveEvent, MouseLeftClickEvent, MouseLeftClickPressed,
    MouseLeftClickLeave
}
```

class UIElement

{

public:

```
UIElement(ElementType type, iPoint position, UIElement* parent, bool isEnabled, bool Draggable, SDL_Rect rectToDraw)
virtual ~UIElement()
virtual void PreUpdate()
virtual void Update(float dt)
void Draw(SDL_Texture* atlas)
virtual SDL_Texture* GetUITexture()
UIElement* GetParent() const
SDL_Rect GetRectToDraw()
iPoint GetPosition() const
bool IsMouseInsideElement(int MarginX, int marginY)
EventElement GetEvent() const
void SetParentAndChildren(UIElement* children)
void DragUIElement()
void MoveInParentLimits(int movementX, int)
void ChangeEnabled()
```

protected:

```
ElementType type
EventElement Event = NoEventElement
int posX
int posY 1
bool horizontalSlider

SDL_Rect rectToDraw
UIElement* parent

bool draggable
iPoint mousePositionFirst
iPoint mousePositionFinal
```

public:

```
bool isEnabled
p2List<UIElement*> listChildren
iPoint position
```

(Herencia de j1Element) **j1Button:**

```
public:
    UIButton(iPoint, SDL_Rect[], p2SString text, SDL_Color color, _TTF_Font* font, bool draggable, UIElement* parent =
nullptr, bool isEnabled = true)
    virtual ~UIButton()
    void Update(float dt)

private:
    UIImage* buttonImage
    UILabel* buttonLabel
    SDL_Rect rects[3]
    EventElement Event
```

(Herencia de j1Element) **j1Image:**

```
public:
    UIImage(iPoint, SDL_Rect, bool draggable, UIElement* parent = nullptr, bool isEnabled = true)
    virtual ~UIImage()
    void setRectToDraw(SDL_Rect newRectToDraw)
```

(Herencia de j1Element) **j1Label:**

```
public:
    UILabel(iPoint position, p2SString text, SDL_Color color, _TTF_Font* font, bool draggable, UIElement* parent = nullptr,
bool isEnabled = true)
    virtual ~UILabel()
    void CentralizeLabel(UIImage* image)
    void InitPosToWrite(UIImage* image)
    p2SString text
    SDL_Texture* GetUITexture()
    SDL_Texture* textureFont
    void SetNewSizeAndTextureFont(p2SString text)

private:
    SDL_Color color
    _TTF_Font* font
```

(Herencia de j1Element) **j1Slider:**

```
public:
    UISlider(iPoint position, SDL_Rect barImage, SDL_Rect zamImage, bool draggable, UIElement* parent = nullptr, bool
isEnabled = true)
    virtual ~UISlider()
    void PreUpdate()
    void SetValueSlider(UIElement* element, int& valueToChange, int maxValue)

private:
    UIImage* bar
    UIImage* zam
    iPoint previousRelativePosition
    int minValue = -1
```