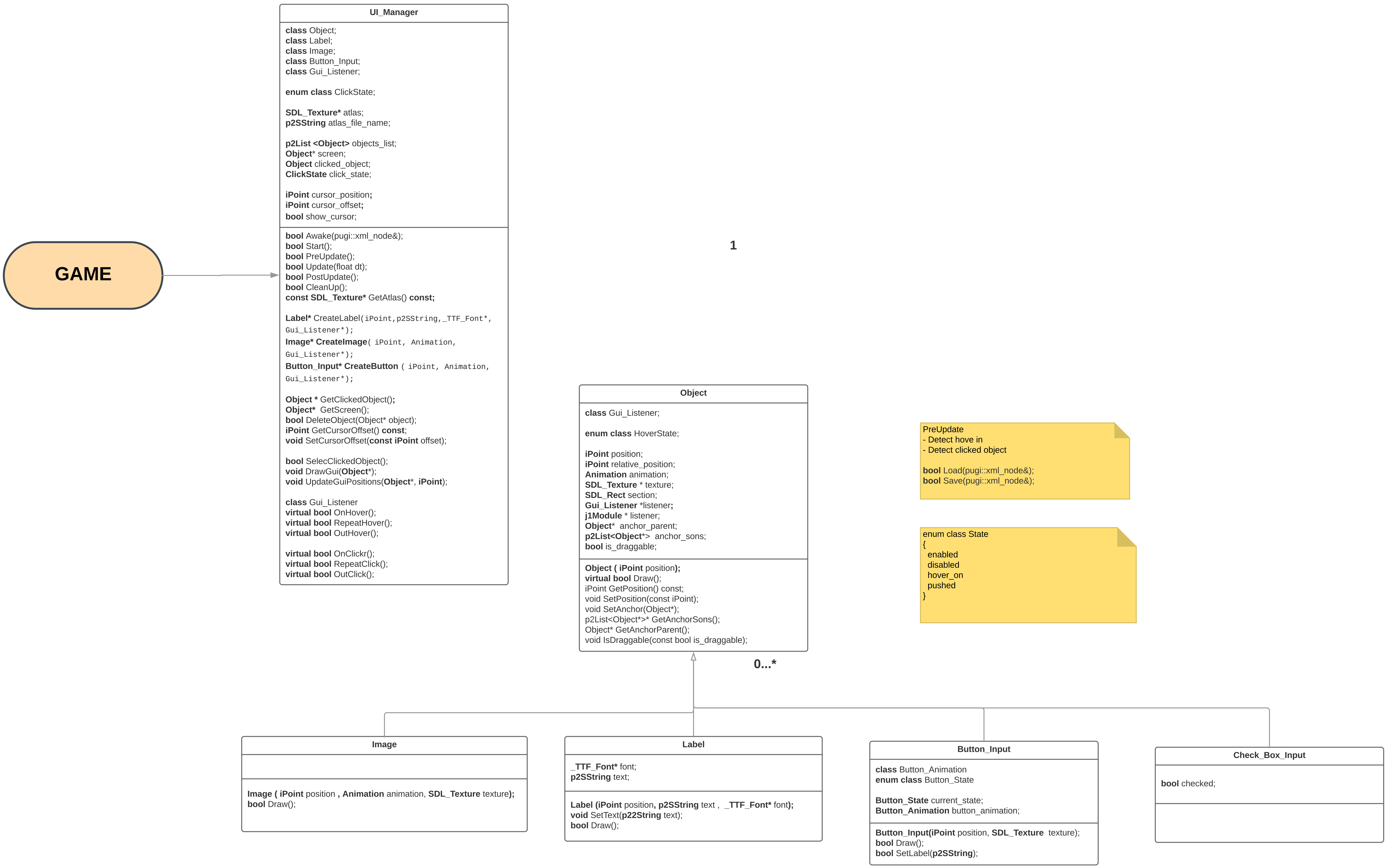

UI_Manager

```
class Object;
class Label;
class Image;
class Button_Input;
class Gui_Listener;

enum class ClickState;

SDL_Texture* atlas;
p2SString atlas_file_name;

p2List <Object> objects_list;
Object* screen;
Object clicked_object;
ClickState click_state;

iPoint cursor_position;
iPoint cursor_offset;
bool show_cursor;
```

```
bool Awake(pugi::xml_node&);
bool Start();
bool PreUpdate();
bool Update(float dt);
bool PostUpdate();
bool CleanUp();
const SDL_Texture* GetAtlas() const;

Label* CreateLabel(iPoint,p2SString,_TTF_Font*,
Gui_Listener*);
Image* CreateImage( iPoint, Animation,
Gui_Listener*);
Button_Input* CreateButton ( iPoint, Animation,
Gui_Listener*);

Object * GetClickedObject();
Object*  GetScreen();
bool DeleteObject(Object* object);
iPoint GetCursorOffset() const;
void SetCursorOffset(const iPoint offset);

bool SelecClickedObject();
void DrawGui(Object*);
void UpdateGuiPositions(Object*, iPoint);

class Gui_Listener
virtual bool OnHover();
virtual bool RepeatHover();
virtual bool OutHover();

virtual bool OnClickr();
virtual bool RepeatClick();
virtual bool OutClick();
```

Object

```
class Gui_Listener;

enum class HoverState;

iPoint position;
iPoint relative_position;
Animation animation;
SDL_Texture * texture;
SDL_Rect section;
Gui_Listener *listener;
j1Module * listener;
Object*  anchor_parent;
p2List<Object*>  anchor_sons;
bool is_draggable;
```

```
Object ( iPoint position);
virtual bool Draw();
iPoint GetPosition() const;
void SetPosition(const iPoint);
void SetAnchor(Object*);
p2List<Object*>* GetAnchorSons();
Object* GetAnchorParent();
void IsDraggable(const bool is_draggable);
```

PreUpdate
- Detect hove in
- Detect clicked object

```
bool Load(pugi::xml_node&);
bool Save(pugi::xml_node&);
```

```
enum class State
{
  enabled
  disabled
  hover_on
  pushed
}
```

Image

```
Image ( iPoint position , Animation animation, SDL_Texture texture);
bool Draw();
```

Label

```
_TTF_Font* font;
p2SString text;
```

```
Label (iPoint position, p2SString text ,  _TTF_Font* font);
void SetText(p22String text);
bool Draw();
```

Button_Input

```
class Button_Animation
enum class Button_State

Button_State current_state;
Button_Animation button_animation;
```

```
Button_Input(iPoint position, SDL_Texture  texture);
bool Draw();
bool SetLabel(p2SString);
```

Check_Box_Input

```
bool checked;
```