

Immediate Mode GUI – Theory and Example

Adam Sawicki

<http://asawicki.info>

What is GUI?

Graphical User Interface (GUI)


aka Head-Up Display (HUD)

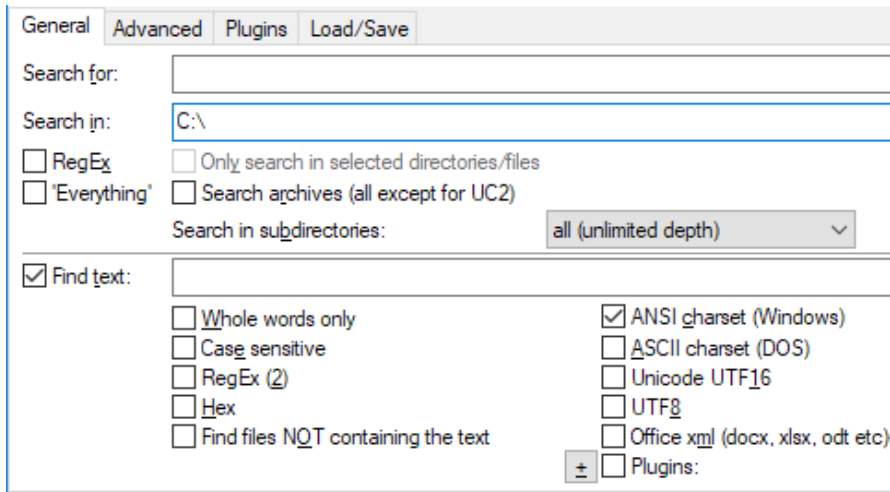
- Displays information
- Handles interaction
 - Mouse, touch, keyboard, other controllers...



Implementation

1. Default system controls

- (Used in desktop apps)
- Pure system API: WinAPI 
- C++ library: Qt, wxWidgets, ...
- Another programming language: C#, Java, ...

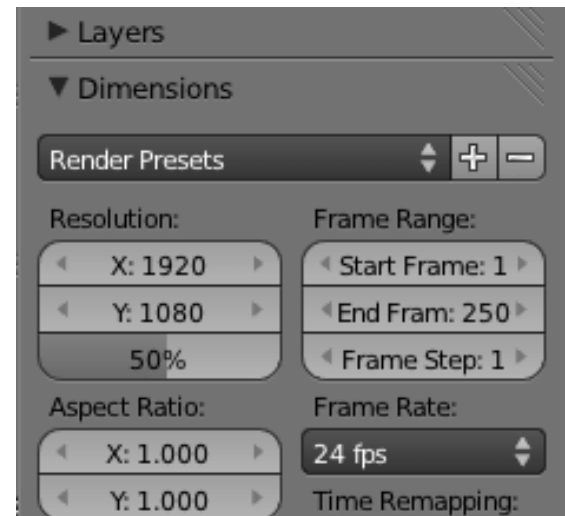


The screenshot shows a search dialog box with the following settings:

- Search for: (empty text box)
- Search in: C:\
- RegEx
- 'Everything'
- Only search in selected directories/files
- Search archives (all except for UC2)
- Search in subdirectories: all (unlimited depth)
- Find text:
- Whole words only
- Case sensitive
- RegEx (2)
- Hex
- Find files NQT containing the text
- ANSI charset (Windows)
- ASCII charset (DOS)
- Unicode UTF16
- UTF8
- Office xml (docx, xlsx, odt etc)
- Plugins: (with a plus-minus icon)

Implementation

1. Default system controls
2. Custom rendering
 - (Used in games, graphics apps)
 - Library: Scaleform, Dear ImGui, ...
 - Your own implementation



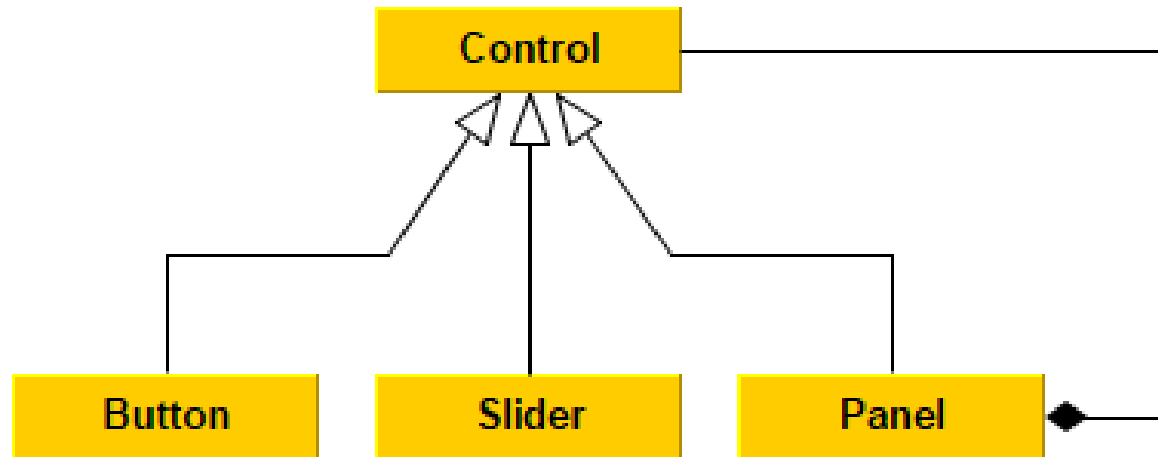
Architecture

Object-oriented – seems to be the most natural architecture for GUI

- Class hierarchy for types of controls
- Design patterns, e.g. composite
- Fields: Position, Size, ...
- Methods: Show, Hide, Enable, Disable, ...

Architecture

Object-oriented – seems to be the most natural architecture for GUI



Example

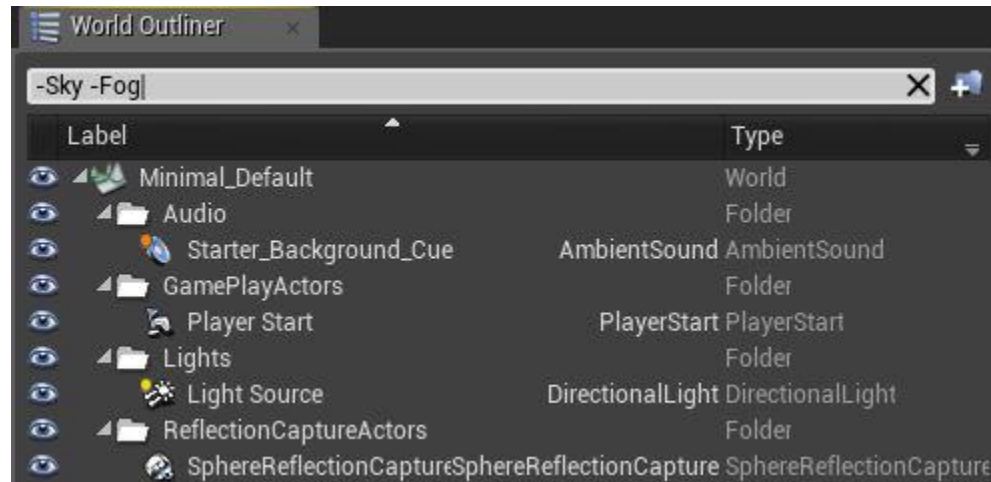
The screenshot displays a game engine interface with several windows and controls:

- Zabawa czcionką (Font Play):** A text editor window with the text "Hello World!". It includes settings for font (Arial Bold), color, and alignment (Left, Middle, Right; Top, Middle, Bottom). It also has options for wrapping (One line, Normal, Words, Characters) and underlining (Single, Double, Overline, Strikethrough).
- Statystyki (Statistics):** A window showing performance metrics: Frame=7828, FPS=60.0138, Draws=173, Primitives=3490, Resources=9, Loaded=1, Locked=7, PhysMem=43%, TexFree=485MB.
- Przeglądarka plików (File Explorer):** A window showing a directory listing with columns for Name, Size, Date modified, and Attributes. The "Documentation" folder is selected.
- Ustawienia (Settings):** A sidebar menu with options like Adapter, Device Type, Back Buffer, Multi Sample, Multi Sample Quality (0), Vertex Processing (AUTO), Back Buffer Count (2), Swap Effect (DISCARD), and Presentation Interval (DEFAULT).
- Przebieg (Timeline):** A grid at the bottom right with columns labeled A, B, C, D and rows labeled 1, 2, 3, 4 and 5, 6, 7, 8.

Overlaid on the text editor is the text: **Oto wpisany tekst**
Hello World!

Some thoughts

- Actual objects also form a hierarchy
 - Every control is positioned relative to its parent
- Similar to how games represent scene
 - High-level approach found in game engines – hierarchy of objects



Immediate mode GUI – idea

- On low level (DirectX, OpenGL, Vulkan), rendering is stateless – „immediate mode”
 - Sequence of draw calls repeated every frame

- SetShader
- SetTexture
- DrawTriangles
- SetTexture
- DrawTriangles
- ...

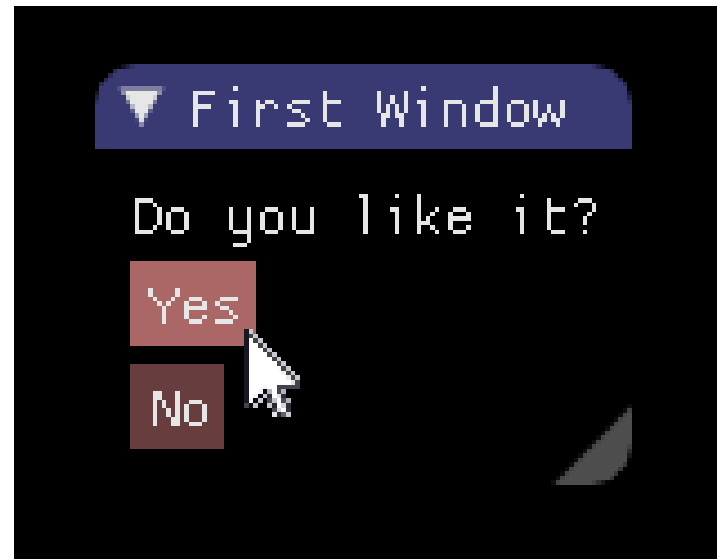
- What if... we could render GUI this way?

Dear ImGui

- <https://github.com/ocornut/imgui>
- C++ library
 - Bindings to many languages available
- License: MIT
- Author: Omar Cornut (game developer)
- Suited for real-time rendering
 - Efficient
 - Graphics API agnostic

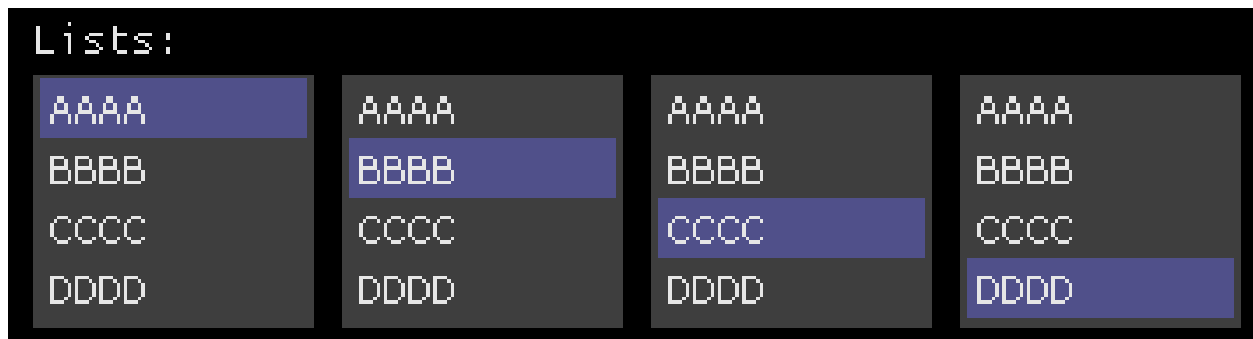
Example code

```
if(ImGui::Begin("First Window"))
{
    ImGui::Text("Do you like it?");
    ImGui::Button("Yes");
    ImGui::Button("No");
    ImGui::End();
}
```



Q: How are controls positioned?

- Automatic – each control in new row
- You can change this:
 - PushItemWidth(item_width), SameLine()
 - Columns: Columns(count)
 - Grouping: BeginGroup(), EndGroup()
 - Full control: GetCursorPos(), SetCursorPos(local_pos)



Q: How do I handle feedback?

You receive it the same moment you define a control

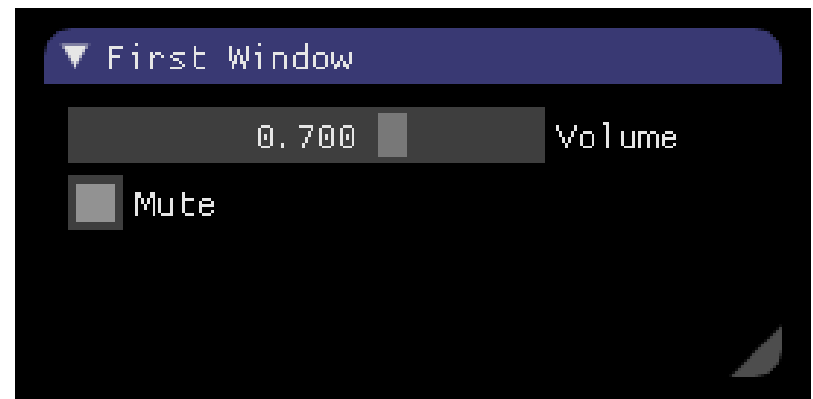
```
ImGui::Text("Do you really want to quit?");  
if(ImGui::Button("Yes"))  
    ExitProgram();  
if(ImGui::Button("No"))  
    CloseThisWindow();
```

Q: Where is value of controls?

- In your own variables
- You pass pointers to them as you define controls

```
float volume = 0.7f;  
bool mute = true;
```

```
ImGui::SliderFloat("Volume", &volume, 0.0f, 1.0f);  
ImGui::Checkbox("Mute", &mute);
```



Q: How about other state?

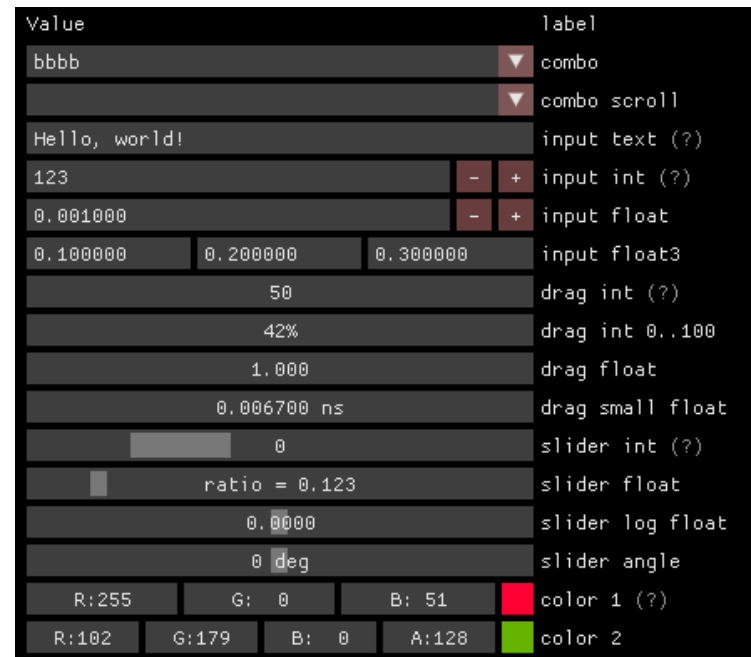
- There is other state
 - Window position & size, focus, text selection, ...
- Kept inside ImGui library
 - So it's not truly stateless...
- Controls are identified by hash from their labels
 - If not unique, you can use "Label##UNIQUE_ID"
 - You can also scope your labels: PushID(), PopID()

Q: How to render?

- If your frame has Update() and Render():
 - Do all ImGui inside Update()
 - Inside Render() you must render it yourself
- For rendering you receive a sequence of:
 - Vertex array
 - Index array
 - Texture ID
 - Scissor rectangle
- Examples available for: DirectX 9, 10, 11, OpenGL, Vulkan, Allegro, ...

Features: Property grid

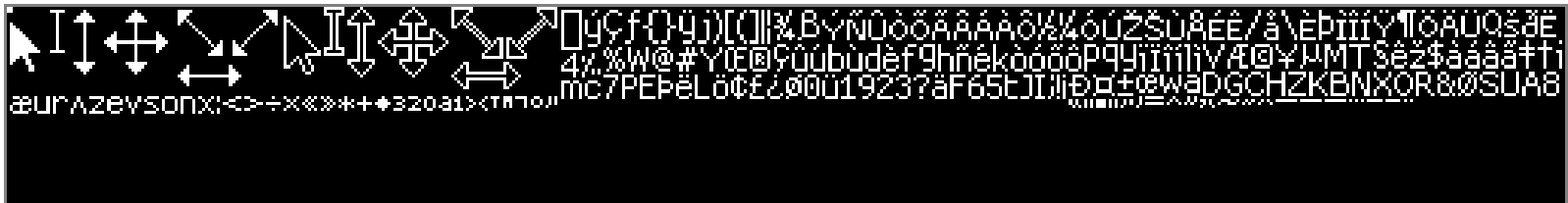
- Suited to provide editing of properties of various types
 - bool, int, float, string, enum
 - vec2, vec3, vec4, color



Value	label
bbbb	combo
	combo scroll
Hello, world!	input text (?)
123	input int (?)
0.001000	input float
0.100000 0.200000 0.300000	input float3
50	drag int (?)
42%	drag int 0..100
1.000	drag float
0.006700 ns	drag small float
0	slider int (?)
ratio = 0.123	slider float
0.0000	slider log float
0 deg	slider angle
R:255 G: 0 B: 51	color 1 (?)
R:102 G:179 B: 0 A:128	color 2

Features: Fonts

- Supports TTF fonts, loaded into atlas texture
- You receive data for this texture, need to upload it to GPU by yourself



Live demo

Conclusion

- Dear ImGui is good for:
 - Internal in-game GUI for debugging purposes
 - Easy to use
 - Efficient to render
- Dear ImGui is not good for:
 - Fully-featured game editor
 - Not well suited for complicated GUI, better use system controls
 - Final game HUD
 - No possibility of skinning or making it look pretty
- You can always make your own implementation
 - Based on idea of Immediate Mode GUI!

Thank you

Questions?