Immediate Mode
Graphical User Interfaces

Presented by
Rickard Gustafsson
Johannes Algelind
Traditional method

• Retained Mode Graphical User Interfaces
  – Uses Model View Controller
  – All your data lives in the GUI library
    • You must copy your data back and forth between application and library (position, size, text, ...)
  – Communicate with your widgets using ID’s or objects
  – Big library with lot’s of classes and code.
Why do we want a new way of doing GUls?

• Retained mode is a pain
  – Message Queue / Callbacks
  – ID’s / Objects
  – Decentralized linkage
  – Synchronization between GUI library and application (inherently because of MVC)
  – Data-driven
  – Adding custom widgets is tricky
Typical RMGUI usage example

```cpp
WidgetSet* ws;
const unsigned int ID_BUTTON = 1;
const unsigned int ID_SLIDER = 2;
float my_float = 0.5f;

void init()
{
    ws = new WidgetSet();
    ws->add(create_button(ID_BUTTON, "Click me", ...));
    ws->add(create_slider_float(ID_SLIDER, "Slide me", 0.f, 1.f, ...));
    ws->set_float_value(ID_SLIDER, my_float);
    ws->set_callback(&callback(ws);
}

void callback_ws(int widget_id)
{
    switch(widget_id)
    {
    case ID_BUTTON: do_action(); break;
    case ID_SLIDER: my_float = ws->get_float_value(ID_SLIDER); break;
    }
}

void main()
{
    init();
    while(running)
        ws->draw();
}
```
Immediate Mode GUI

• Casey Muratori stumbled upon IMGUI around 2002 while working at RAD Game Tools.

• A style of doing GUIs which are well suited for real-time rendered applications.

• Zero Memory Widgets – IMGUI library for regular applications, appeared around 2003.
IMGUI usage example

```c
void draw_gui(struct GUIState_t* state, float* my_float)
{
    if (do_button(state, "Click me", ...))
        do_action();
    do_slider_float(state, "Slide me", my_float, 0.f, 1.f, ...);
}

void main()
{
    GUIState_t* state;
    float my_float = 0.5f;
    while(running)
    {
        update(state, ...);
        draw_gui(state, &my_float);
    }
}
```
IMGUI traits

• No data synchronization needed, all your data is stored in the application.
• Stores very little state in the library
• Uses procedural function-calls as ”widgets”.
• Code-driven
• Centralized flow control
• Easy to dive in to, easy to extend.
• Must be redrawn every frame
Simple IMGUI implementation

• Basic concepts
  – A widget is considered "hot" if there is a possibility that it will be interacted with.
  – An "active" widget is currently engaged by the user.
  – Widget ID’s are used for the library to keep track of hot and active widgets.
Simple IMGUI implementation contd.

```c
struct GUIState_t {
    void* hot_item;
    void* active_item;
    int mouse_x, mouse_y;
    unsigned mouse_buttons;
};

struct Rect_t {
    float x, y, w, h;
};

bool is_button_down(GUIState_t* state, unsigned button) {
    return (state->mouse_buttons >> button - 1) & 1;
}

bool mouse_inside_rect(GUIState_t* state, Rect_t* rect) {
    /* point-in-rect-test */
}

void update(GUIState_t* state, unsigned mb, int mx, int my) {
    state->mouse_x = mx;
    state->mouse_y = my;
    state->mouse_buttons = mb;
}

bool do_button(GUIState_t* state, void* id, Rect_t* rect, const char* text, ...) {
    bool result = false;
    bool inside = mouse_inside_rect(state, rect);
    if (inside)
        state->hot_item = id;
    if (state->active_item == id && !is_button_down(1)) {
        if (inside) {
            result = true;
            state->hot_item = id;
        } else
            state->hot_item = 0;
    } else if (state->hot_item == id) {
        if (is_button_down(1)) {
            state->focused_item = id;
            state->active_item = id;
        }
    }
    draw_button(rect, text, ...);
    return result;
}
```
Demo-time
# IMGUI in Teeworlds

## Server List

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Map</th>
<th>Players</th>
<th>Ping</th>
</tr>
</thead>
<tbody>
<tr>
<td>(SWE) CTF</td>
<td>CTF</td>
<td>ctf2</td>
<td>3/16</td>
<td>21</td>
</tr>
<tr>
<td><em>gV</em> - ctf ctf2</td>
<td>CTF</td>
<td>ctf2</td>
<td>16/16</td>
<td>41</td>
</tr>
<tr>
<td>TinyBox.net</td>
<td>CTF</td>
<td>ctf2</td>
<td>4/16</td>
<td>44</td>
</tr>
<tr>
<td>[BiZzaro]ctf</td>
<td>DM</td>
<td>ctf2</td>
<td>0/16</td>
<td>49</td>
</tr>
<tr>
<td>CTF Server by GameMods.org</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/10</td>
<td>50</td>
</tr>
<tr>
<td>DT:Elite</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/16</td>
<td>52</td>
</tr>
<tr>
<td>TieBones TeeWorld</td>
<td>DM</td>
<td>ctf2</td>
<td>2/16</td>
<td>52</td>
</tr>
<tr>
<td>KirovNet.CTF Teeworlds Server</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/12</td>
<td>56</td>
</tr>
<tr>
<td>-CyberPain Teeworld CTF Server-</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/16</td>
<td>65</td>
</tr>
<tr>
<td>e66_tw_server</td>
<td>DM</td>
<td>ctf2</td>
<td>2/16</td>
<td>69</td>
</tr>
<tr>
<td>GameZoo.it CTF</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/12</td>
<td>77</td>
</tr>
<tr>
<td>set sail for fail CTF</td>
<td>DM</td>
<td>ctf2</td>
<td>0/16</td>
<td>82</td>
</tr>
<tr>
<td>MegaStyle TW CTF</td>
<td>CTF</td>
<td>ctf2</td>
<td>1/12</td>
<td>84</td>
</tr>
<tr>
<td>Gagarino.NET CTF</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/16</td>
<td>86</td>
</tr>
<tr>
<td>Phönix CTF2 // by ZockerStube.net</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/16</td>
<td>88</td>
</tr>
<tr>
<td>bloodarmy.ru TW CTF</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/16</td>
<td>96</td>
</tr>
<tr>
<td>Games-by.net Teeworlds sample ctf</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/8</td>
<td>113</td>
</tr>
<tr>
<td>teeworlds.raspi.fi</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/16</td>
<td>124</td>
</tr>
<tr>
<td>GG.ST HS ctf server</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/16</td>
<td>130</td>
</tr>
<tr>
<td><em>gV</em> - cKz Clanserver</td>
<td>CTF</td>
<td>ctf2</td>
<td>6/16</td>
<td>130</td>
</tr>
<tr>
<td>CoCoS [RUS] Teeworlds CTF</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/8</td>
<td>134</td>
</tr>
<tr>
<td>ROSTOV dm</td>
<td>DM</td>
<td>ctf2</td>
<td>0/12</td>
<td>135</td>
</tr>
<tr>
<td>srgames.net</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/16</td>
<td>142</td>
</tr>
<tr>
<td>6on6 BattleTees #1 (Qi Clanserver)</td>
<td>CTF</td>
<td>ctf2</td>
<td>2/16</td>
<td>145</td>
</tr>
<tr>
<td>[Qi] High level only (pass= qiRox)</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/16</td>
<td>148</td>
</tr>
<tr>
<td>EFS-Teeeworlds-Server</td>
<td>CTF</td>
<td>ctf2</td>
<td>0/12</td>
<td>162</td>
</tr>
<tr>
<td>AB</td>
<td>Public Server</td>
<td>CTF</td>
<td>24/7</td>
<td>ctf2</td>
</tr>
</tbody>
</table>

Quick search: ctf2

28 of 679 servers, 36 players

Current version: 0.5.1

Host address: 217.172.180.79:8303
Unity
What’s next?

- Partitioned IMGUI ("PIMGUI")
Sources & more info

• “Immediate Mode GUIs” written by Sean Barret for GDM Sept. 2005.
• Introduction video by Casey Muratori
• The Molly Rocket forums
• Book in progress by Johannes Norneby
• Tutorial by Jari Komppa
• IMGUI implementation by nVidia
• Zero Memory Widgets
Questions?