EDITOR

En este capítulo se mostrará como incluir un editor en nuestro motor. Para crear el editor se utilizará un recurso externo: imgui.

AÑADIENDO IMGUI

El primer paso es añadir al proyecto los ficheros de imgui necesarios, dichos ficheros pueden descargarse de la carpeta Librerías. Una vez descargados de dicha carpeta, deben añadirse al proyecto los siguientes ficheros: *imgui.h*, *imgui\_impl\_sdl.h* e *imgui\_impl\_opengl3.h*. Se recomienda añadir todos los archivos que se encuentran en la carpeta *imgui*.

Para poder utilizar este editor, no hace falta una librería ni un archivo *.dll*, lo único que hacen falta son los archivos de cabecera.

Ras el añadido de *imgui*, deberemos crear una nueva clase.

CREANDO EL MODULOEDITOR

Nuestro siguiente paso será la creación de la nueva clase *editor()*. Tras crear la nueva clase, lo primero que haremos será declararla en *Application*. Así pues abriremos el fichero *Application.cpp* y haremos un *include* para *ModuleEditor*.

*#include "ModuleEditor.h"*

Tras el include, como siempre, añadiremos la siguiente línea:

*modules.push\_back(editor = new ModuleEditor());*

Que será añadida al final del bloque. Con esto habremos finalizado con el fichero *Application.cpp* y pasaremos a su archivo de cabecera. En dicho archivo definiremos la clase *editor()*:

*class ModuleEditor;*

Tras la declaración de la clase sólo nos quedará realizar la declaración del puntero mediante:

*ModuleEditor\* editor = nullptr;*

Una vez asociado el nuevo módulo a la aplicación, se pasará a introducir el código del módulo. Para ello, iremos al fichero *ModuleEditor.cpp*.

Lo primeros que introduciremos serán los *include*. En este caso serán los siguientes:

*#include "ModuleEditor.h"*

*#include "Globals.h"*

*#include "Application.h"*

*#include "ModuleEditor.h"*

*#include "ModuleWindow.h"*

*#include "ModuleRender.h"*

*#include "ModuleCamera.h"*

*#include "ModuleInput.h"*

*#include "glew-2.1.0\include\GL\glew.h"*

*#include "SDL.h"*

*#include "IMGUI\imgui.h"*

*#include "IMGUI\imgui\_impl\_sdl.h"*

*#include "IMGUI\imgui\_impl\_opengl3.h"*

Para esta clase utilizaremos el espacio de nombres estándar; por lo que después de los *include* añadiremos la siguiente línea:

*using namespace std;*

Dejaremos el constructor y el destructor vacíos, no necesitamos incluir nada por el momento. Pero creamos el método Init() y añadimos a su interior el siguiente código:

*bool ModuleEditor::Init()*

*{*

*IMGUI\_CHECKVERSION();*

*ImGui::CreateContext();*

*io = ImGui::GetIO(); (void)io;*

*io.ConfigFlags |= ImGuiConfigFlags\_NavEnableKeyboard; // Enable Keyboard Controls*

*io.ConfigFlags |= ImGuiConfigFlags\_DockingEnable; // Enable Docking*

*io.ConfigFlags |= ImGuiConfigFlags\_ViewportsEnable; // Enable Multi-Viewport / Platform Windows*

*ImGui\_ImplSDL2\_InitForOpenGL(App->window->window, App->renderer->context);*

*ImGui\_ImplOpenGL3\_Init(glsl\_version);*

*// Setup style*

*ImGui::StyleColorsDark();*

*return true;*

*}*

También crearemos un método *PreUpdate()* y le añadiremos el siguiente código:

*update\_status ModuleEditor::PreUpdate()*

*{*

*fps\_log.erase(fps\_log.begin());*

*fps\_log.push\_back(App->fps);*

*return UPDATE\_CONTINUE;*

*}*

También añadiremos el método *Update()*, donde incluiremos el siguiente código:

*update\_status ModuleEditor::Update()*

*{*

*if (showConfiguration)*

*{*

*DrawConfiguration();*

*}*

*if (showModel)*

*{*

*DrawModel();*

*}*

*if (showAbout)*

*{*

*DrawAbout();*

*}*

*if (showHardware)*

*{*

*DrawHardware();*

*}*

*if (showConsole)*

*{*

*DrawConsole();*

*}*

*//Menu*

*if (DrawMenu()) return UPDATE\_STOP;*

*return UPDATE\_CONTINUE;*

*}*

Este método va llamando a todos los módulos que comprenderán el editor. Tras este método llamaremos al méodo *PostUpdate()*, que contendrá el siguiente código:

*update\_status ModuleEditor::PostUpdate()*

*{*

*return UPDATE\_CONTINUE;*

*}*

Con esto temenos los métodos principals definidos, a excepción del *CleanUp()* que se definirá al final. Lo que haremos a continuación será definr todos y cada uno de los métodos a los que llama el método *Update()*. Empezaremos definiendo el método *AddLog()*.

Este método contendrá el código para añadir un *log* donde ir mostrando lo que va sucediendo en nuestro motor:

*void ModuleEditor::AddLog(const char\* fmt, ...)*

*{*

*va\_list args;*

*va\_start(args, fmt);*

*Buf.appendfv(fmt, args);*

*va\_end(args);*

*ScrollToBottom = true;*

*}*

Tras este método definiremos el método *Clean()* que limpiará el *buffer*.

*void ModuleEditor::Clear()*

*{*

*Buf.clear();*

*}*

Tras este método añadiremos el método *AddMemory()* con el siguiente código:

*ModuleEditor::AddMemory(float memory)*

*{*

*for (unsigned int i = 0; i < MEMORY\_LOG\_SIZE - 1; ++i)*

*{*

*memory\_log[i] = memory\_log[i + 1];*

*}*

*memory\_log[MEMORY\_LOG\_SIZE - 1] = memory;*

*}*

Tras este método incluiremos el método *DrawMenu()* para dibujar el menú del editor.

*bool ModuleEditor::DrawMenu()*

*{*

*ImGui::BeginMainMenuBar();*

*if (ImGui::BeginMenu("File"))*

*{*

*if (ImGui::MenuItem("Exit"))*

*{*

*return true;*

*}*

*ImGui::EndMenu();*

*}*

*if (ImGui::BeginMenu("Engine"))*

*{*

*if (ImGui::MenuItem("Model"))*

*{*

*showModel = !showModel;*

*}*

*if (ImGui::MenuItem("Configuration"))*

*{*

*showConfiguration = !showConfiguration;*

*}*

*if (ImGui::MenuItem("Console"))*

*{*

*showConsole = !showConsole;*

*}*

*ImGui::EndMenu();*

*}*

*if (ImGui::BeginMenu("Help"))*

*{*

*if (ImGui::MenuItem("Repository"))*

*{*

*ShellExecute(NULL, "open", "https://github.com/Gabroide/DevelopingAnEngine", NULL, NULL, SW\_SHOWNORMAL);*

*}*

*if (ImGui::MenuItem("Hardware"))*

*{*

*showHardware = !showHardware;*

*}*

*if (ImGui::MenuItem("About"))*

*{*

*showAbout = !showAbout;*

*}*

*ImGui::EndMenu();*

*}*

*ImGui::EndMainMenuBar();*

*return false;*

*}*

fefw