Code Yarns 👨‍💻

Notes from the world of software

≡ **Menu**

How to register class method as C callback

Ashwin Uncategorized 2015-09-012015-09-01 1 Minute

Problem

A typical problem when using a C library with your own C++ code: the library requires a **C callback** function pointer, but you want to pass your C++ class method (that is non-static) to it.

I face this problem when using C libraries like **GLFW** or **GLUT**, which provide an interface to **OpenGL**, which is also a C library. For example, say I want to register a C++ class method with GLFW as callback for mouse button event. GLFW expects me to pass it a C function pointer with this signature:

```
void ButtonCallback(GLFWwindow*, int, int, int);

// Register above function as callback
glfwSetMouseButtonCallback(window, ButtonCallback);
```

I want to register this C++ class method as the callback:

```
class Foo
{
public:
    Foo()
    {
        // Error!
        glfwSetMouseButtonCallback(window, FooButtonCallback);
    }

    void FooButtonCallback(GLFWwindow*, int, int, int)
    { /* something */ }
};
```

No pointer trickery can make it work because the signature of a C++ class non-static method is different from a C callback.

Solution

One solution is to only use C++ class **static methods** as callback. These can be passed as C callback because these are nothing but C functions with a glorified name. However, this causes serious problems later when you want update some class variable with the data received from the callback.

The solution I use in such a scenario is an ugly hack called **trampoline**. The idea is to create a global C function which can be passed as callback and inside it call the C++ method by using its object pointer:

```
Foo* g_foo_ptr = nullptr;
void TrampButtonCallback(GLFWwindow* a, int b, int c, int d)
{
    if (g_foo_ptr) // Check before calling
        g_foo_ptr->FooButtonCallback(a, b, c, d);
}

class Foo
{
public:
    Foo()
    {
        g_foo_ptr = this; // Store global
        glfwSetMouseButtonCallback(window, TrampButtonCallback);
    }

    void FooButtonCallback(GLFWwindow*, int, int, int)
    { /* something */ }
};
```

Advertisements

AdChoices

2019 HIGHLANDER

Get $2500 Cash Back

Options shown. View disclaimers.

SEE OFFERS

FIND A DEALER

REPORT THIS AD

4 Signs You're About To Die Of a Heart Attack

REPORT THIS AD

ADVERTISEMENT

Tagged:
callback,
glfw,
glut,
opengl

Published by Ashwin

View all posts by Ashwin

One thought on “How to register class method as C callback”

barben360 says:
2019-01-23 at 21:29
Wouldn’t ::std::bind work in your case?
Like:
glfwSetMouseButtonCallback(window, ::std::bind(&Foo::FooButtonCallback, this));

↪ Reply

This site uses Akismet to reduce spam. Learn how your comment data is processed.

Create a free website or blog at WordPress.com.