
Subject: Re: C++ templated class referencing each other

Posted by [Lance](#) on Sun, 11 Oct 2020 02:36:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Notice that I essentially put everything in the same header file (equivalently).

Also, if you can guarantee that ComponentManager Will be the first member variable (because why-not and you should put a static_assert to avoid accidentally add some other member variable in front it subsequently), you can probably save the Object& obj; member variable in the ComponentManager class. Simply define it like:

```
class Object;

class ComponentManager{
public:
    ComponentManager(){}

    template <class T> T& CreateComponent(bool active, int position);

    /*
    Many templated function that force me to write declaration in this .h file
    */
private:
    Object& object(){
        return *reinterpret_cast<Object*>(this);
    }
};
```

And change other part of your code accordingly.

PS: If your Object class is virtual, some compiler might put vtable at this. In this case, above hack will be problematic. I will consider to let Object to privately inherit from ComponentManager.
