```
/*
 * Stewart Bracken Copyright 2014
 *
 */
#pragma once

#include "ofMain.h"

#include "ofxLua.h"
#include "ofxLuaBindings.h" // the OF api -> lua binding

#include "ofxUI.h"

#include <map>

class ofApp : public ofBaseApp, ofxLuaListener {

public:

    // main
    void setup();
    void update();
    void draw();
    void exit();


    // input
    void keyPressed(int key);
    void mouseMoved(int x, int y );
    void mouseDragged(int x, int y, int button);
    void mousePressed(int x, int y, int button);
    void mouseReleased(int x, int y, int button);

    // ofxLua error callback
    void errorReceived(string& msg);

    // script control
    void reloadScript();

    ofxLua lua;
    vector<string> scripts;
    int currentScript;
    bool hasError;
    std::string error;
    ofxUICanvas *gui;
    void guiEvent(ofxUIEventArgs &e);
    ofxUICanvas *guiConsole;
    void guiConsoleEvent(ofxUIEventArgs &e);

    void addConsoleMessage(const string&);

private:
    void build_directory_gui();
    void add_to_gui(string path);
    map<string,string> directory_map;
    void reset_directory_gui();
};


class ofGUILoggerChannel: public ofBaseLoggerChannel{
public:
    ofGUILoggerChannel(ofApp* _app):app(_app){};
    //virtual ~ofGUILoggerChannel(){};
    void log(ofLogLevel level, const string & module, const string & message);
    void log(ofLogLevel level, const string & module, const char* format, ...);
    void log(ofLogLevel level, const string & module, const char* format, va_list
args);
private:
    ofApp* app;
};
```