

欢迎使用 Lua Perfect

一、欢迎

欢迎您使用 LuaPerfect! LuaPerfect 是 TX 开发的专业级的 Lua 调试器，致力于为广大 Lua 开发人员提供免费的专业的 Lua 调试工具。

二、调试指南

1、LuaPerfect 调试原理

如下图所示，LuaPerfect 的调试原理是通过在游戏中通过 `require("LuaDebuggee")` 来加载 `LuaDebuggee.dll`，并通过 `LuaDebuggee` 模块设置调试钩子，从而完成调试的功能。

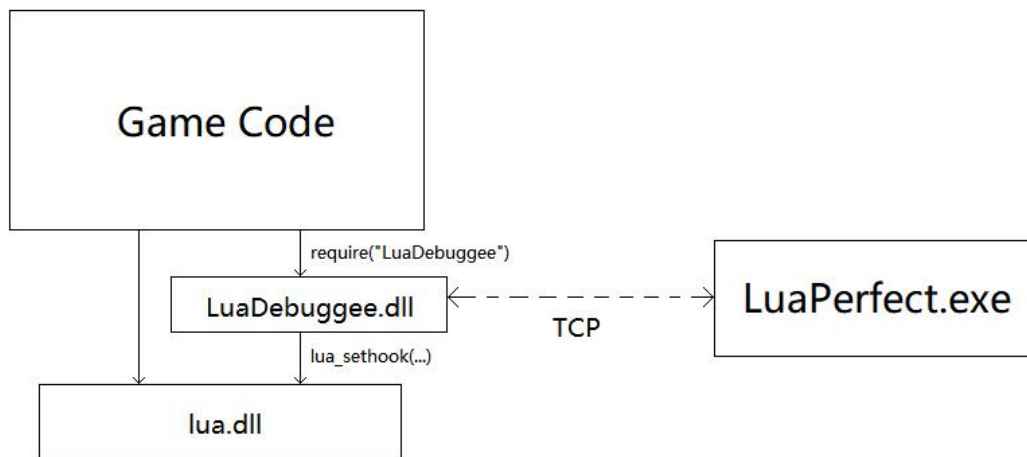


图 1、LuaPerfect 调试原理

2、LuaPerfect 调试具体步骤

步骤 1：在 LuaPerfect 工程列表里选择需要打开的项目，打开项目的同时，LuaPerfect 会自动地将 `LuaDebuggee` 复制到工程的目录，以供游戏中调用（无需手动复制）。

步骤 2：在工程入口点的 Lua 文件的开头，粘贴如下代码：

```
require('LuaDebuggee').StartDebug("127.0.0.1", 9826)
```

步骤 3：在刚刚添加的语句后面执行的语句处放置一个断点（快捷键 F9）。在 Unity 里启动游戏，即会断点到该断点处。

特别说明 1：由于 xLua 官方示例工程中很多地方使用 `DoString()`，建议使用自带的 `00_LuaPerfectTest_xLua` 这个例子进行测试，自带的例子在官方群 `00_LuaPerfectTest_xLua.zip` 中。

特别说明 2：如果加入 `require(“LuaDebuggee”).StartDebug(“127.0.0.1”, 9826)` 后，自定义的 `CustomLuaLoader` 里发生错误，提示找不到 `LuaDebuggee.lua` 或者类似的提示。请修改 `CustomLuaLoader`，对于” `LuaDebuggee`”的情况，直接返回 0。

```
public static int MyCustomLoader(IntPtr L)
{
    string libraryName = XLua.LuaDLL.Lua.lua_tostring(L, 1);
    if (libraryName == "LuaDebuggee") //加上这个判断"LuaDebuggee"返回 0 的即可调试
    {
        return 0;
    }
    ...其他代码
}
```

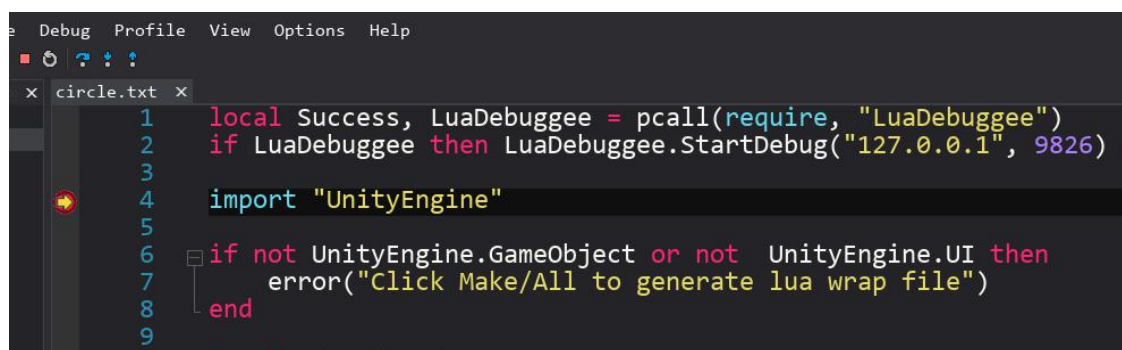


图 2、成功启动调试

三、直接查看 Unity 对象成员

LuaPerfect 除了提供查看常规的 Lua 值/对象的能力，也提供了查看 Unity 里的 C#对象成员的能力，并且更进一步，如果查看的对象是 `GameObject` 对象，会列出其各个子 `GameObject` 对象及 `GameComponent` 对象。因此 `XLua` 和 `SLua`，需要 `Make` 一下，确保 `ThirdParty/LuaPerfect/ObjectFormater.cs` 被正常导出到 Lua。在 `ObjectFormater` 类被正常导出到 Lua 之后，`LuaPerfect` 中即可正常查看 Unity 里的 C#对象。

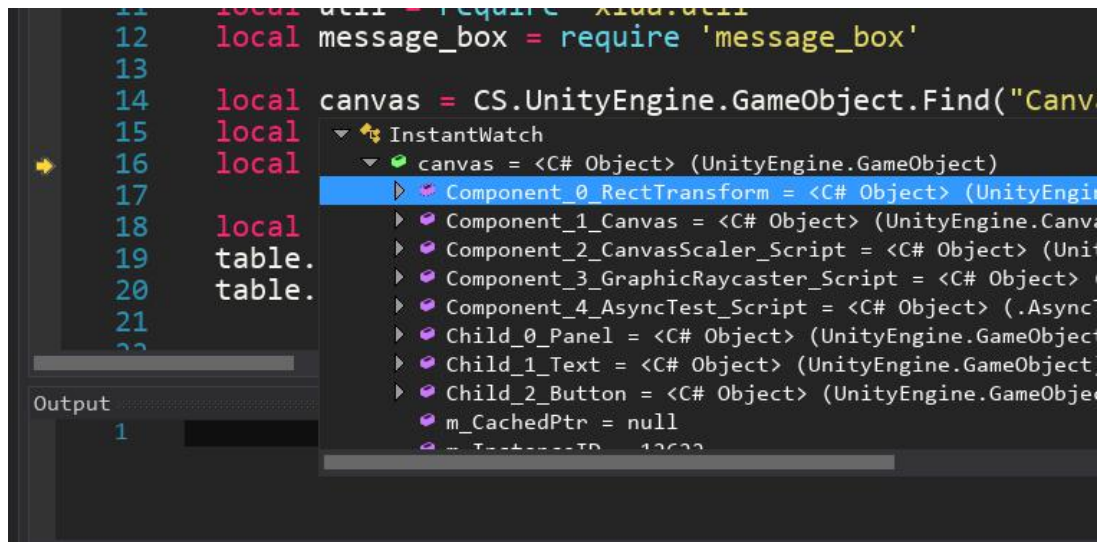


图 3、查看 Unity 里 C#对象内部的值

四、Unity 的 Lua 版本 API 生成

LuaPerfect 拥有强大的类型推导引擎，但是也仍然需要类型信息库才能很好地工作。可以在 Unity 主菜单->LuaPerfect->Generate XXX Apis 按需生成自己版本的 Unity Lua API 文件。生成的 API 文件通常在 Assets/lpproj/Apis/Unity/中。生成完 API 之后，LuaPerfect 即可根据 API 文件对类型进行提示，从而提高编码效率。

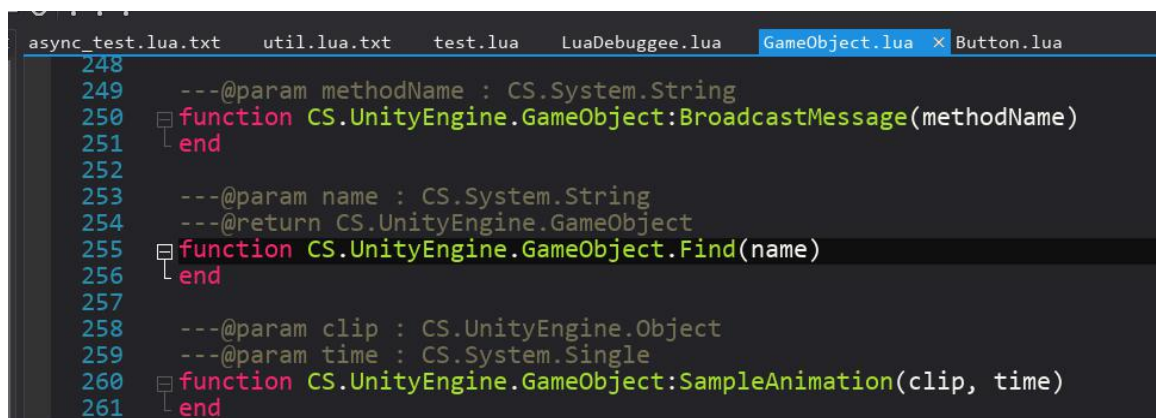


图 4、LuaPerfect 为 Unity 生成的 API 文件

五、大型工程 Lua 代码路径设置

对于大型工程，游戏里的资源文件以十万计，LuaPerfect 默认会打开 Assets 文件夹。但是在 Project 视图里直接指定 Lua 目录是一种更为高效的做法。可以在 Project 界面的 LuaFolders 项点击右键添加 Lua 目录，添加完自己的 Lua 目录之后就可以从工程中移除 Assets 目录了。也可以直接在资源管理器中将需要添加的目录拖动至 Project 界面。

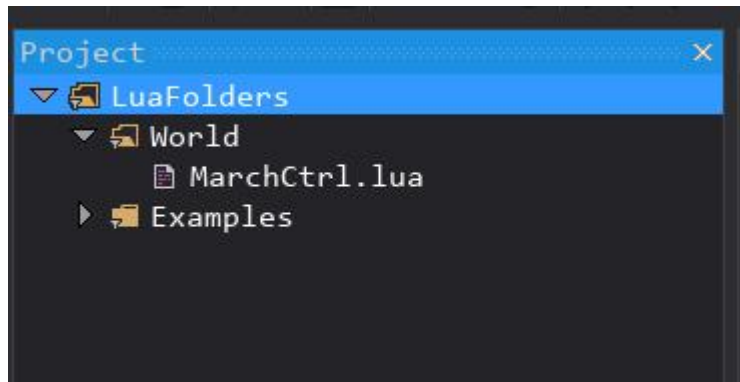


图 5、自定义 Lua 代码文件夹

重新启动时，系统会自动添加 Apis 目录，以便读取 API 信息。

六、性能测试指南

1、启动性能测试

将 StartDebug 替换成 StartProfile，即可在游戏中开始性能测试。亦可以在调试状态下，执行 Profile->Start Profiling 菜单项开始性能测试。

2、结束性能测试

调用 `require("LuaDebuggee").Stop()` 即可结束性能测试，并显示性能测试结果。亦可在编辑器里执行 Profile->Stop Profiling 菜单项结束性能测试，并显示性能测试结果。

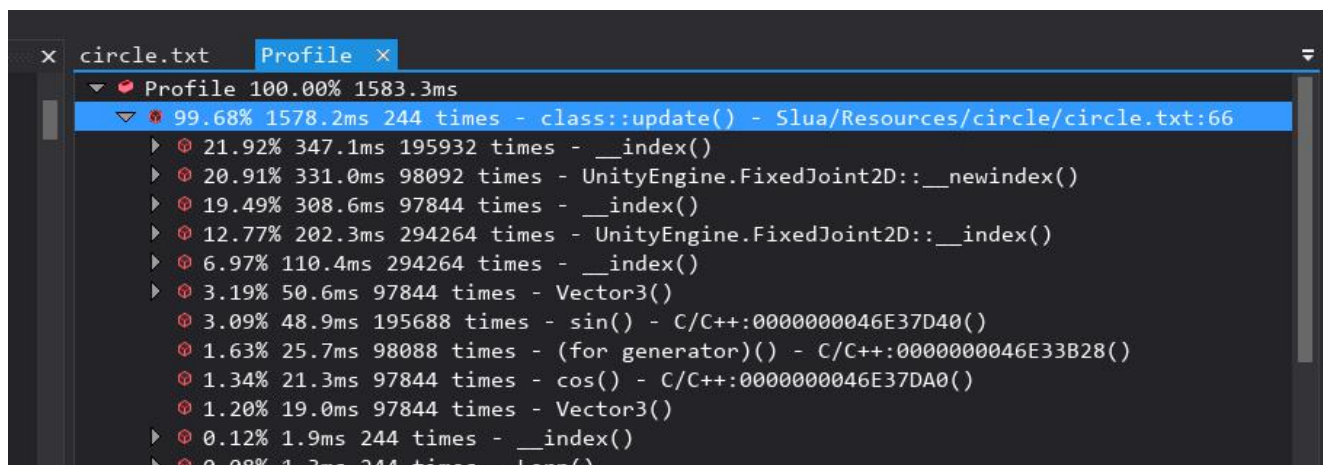


图 6、性能测试结果

七、将工具上传到 SVN/Git

对于希望将工具上传到 SVN/Git 的团队，由于工具自带自动更新功能，所以建议是上传原始的 zip 包，使用时解压即可。

八、实用小技巧

LuaPerfect 里有许多隐藏的小技巧，下面列出一些实用的小技巧：

- 1、双击输出的 log，可以直接跳转输出该 log 的 Lua 代码。
- 2、可以通过右键菜单->Paste StartDebug()快速粘贴此段代码。
- 3、在各变量窗口里，双击 Lua 函数节点，可以查看该 Lua 函数的反汇编。
- 4、Intelligence->Check Syntax(F7)可以快速检查工程中的语法错误。
- 5、Intelligence->Go To File(Ctrl+P)可以快速定位到某个特定文件。
- 6、Intelligence->Go To Symbol(Ctrl+R)可以在文件里快速定位到某个特定函数/变量。
- 7、Intelligence->Go To Symbol In Project(Ctrl+Shift+R)可以在整个工程里快速定位到某个特定函数/变量。
- 8、按住 Ctrl 滚动鼠标滚轮可以调整代码字体大小。
- 9、各种窗口及选项卡均可以拖放。
- 10、主菜单 Options->Track Alive 可以开启/关闭工程界面追踪点击的文件的功能。

更多惊喜等你来发现&创造。

九、视频资源

下面收录的是 Lua Perfect 相关的演示视频：

视频内容：Lua Perfect 演示视频_新 视频时长：6 分 02 秒 获取方法：请到 LuaPerfect 官方群群文件中下载。

十、隐私陈述

为确认版本质量，在联机环境下，Lua Perfect 会对启动/关闭/转储等基础事件进行统计，但不会尝试获取其他任何数据。有需要关闭统计的，可以在 Data/Config/EditorConfig.lpxml 中 SendStatistics 项进行设置。

十一、反馈

感谢您使用 LuaPerfect，在使用中如果遇到任何问题，或者有任何意见建议，请您立刻马上

通过以下方式联系我们：

QQ: 672250695

邮箱: 672250695@qq.com

QQ 群: 932801740（推荐）

推荐加入 LuaPerfect 官方群（932801740）了解最新的版本信息及问题解答，与我们一起打造您专属的 Lua 开发工具。



群名称:LuaPerfect官方群
群 号:932801740