

UI Test 使用

UI Test 是自动化测试工具，依赖于 FairyGUI 和 XLua。

一 导入与初始化

1.1 导入

直接导入 `UITest.unityPackage`。

或者，将 `UITestManager.cs` 放入到 `Assets/` 下，能 `using FairyGUI` 和 `using XLua` 的任意位置就可以了。

1.2 初始化

如果需要运行。UITestManager 的功能，需要在开始时调用

```
UITestManager.Initialize(m_gameObject, new UITestManager.UITestInitializeParameters
{
    m_DoFile = (sFile) => { LuaManager.Instance.DoFile(sFile); },
    m_CallFunction = (sFunction) => { LuaManager.Instance.CallFunction(sFunction); },
});
```

其中，UITestInitializeParameters 的参数，可以参考 UITestManager.cs。

1.3 运行

如果正确调用了 Initialize，PC 和 Editor 中，应该能看到



左上角能看到 UITest 的状态，叫做 Idel 状态。

1.4 自动运行（尚未实现，热重启好了后）

1.5 快捷键

目前，UITest 占用 F7,F8,F9,F10，4 个快捷键。

二 录制

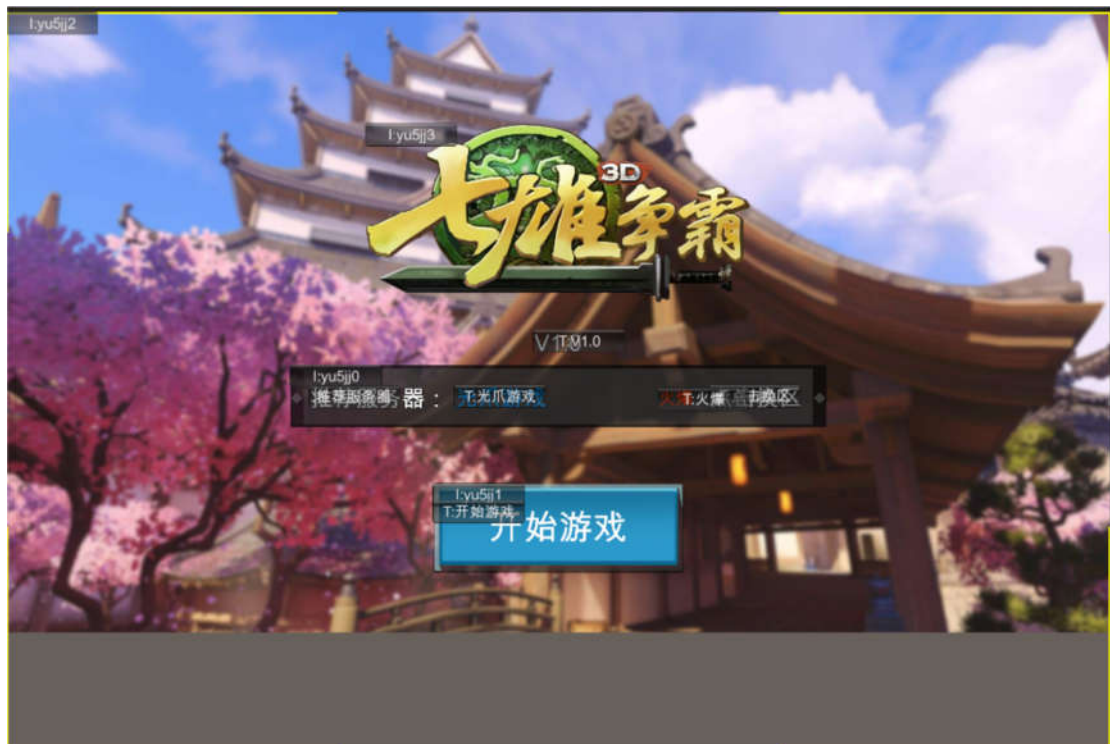
在 Idel 状态下，按 F9 可以开始录制脚本。

按了 F9 后，会看到红色外框，说明此时进入了录制状态



录制状态中，点击按钮会被记录成脚本。

在录制中，按 F10，进入暂停录制界面，此时，会变成黄色外框

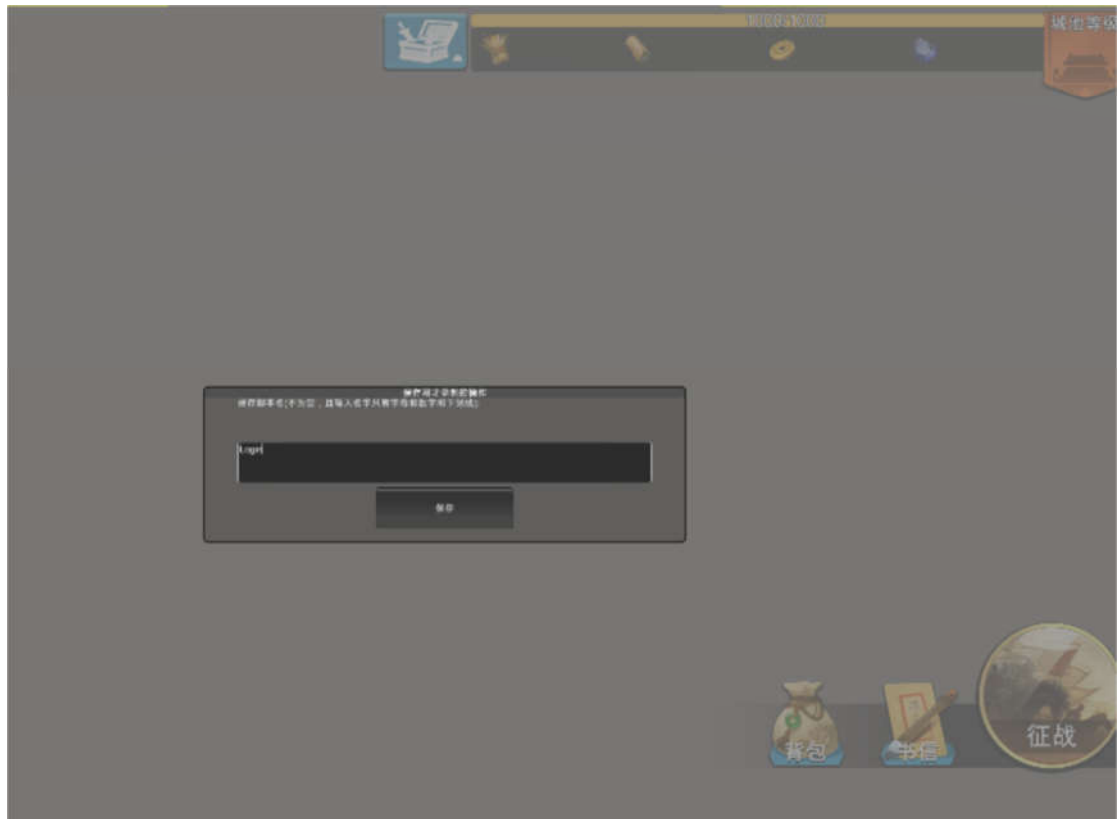


在暂停界面，会显示当前界面检测到的 UI 元素。可以点击它们，这些元素将在脚本运行到这里时，做检查。



被选中做检查的元素会变成蓝色。比如这张图里的状态是，到了这个界面，将检查一个以“yu5jj0”结尾的图片框，和“光爪游戏”，“v1.0”两个文本框，是否存在，以及它们的内容是否为“yu5jj0”结尾的图片，“光爪游戏”以及。“v1.0”

在暂停时，按 F10 将继续录制，录制过程中，随时可以按 F10 暂停。
在录制时，按 F9，将结束录制，进入到保存脚本界面。



比如，前面录制的脚本，就是点击了一下登陆按钮。进入到主城。点击保存后。可以看到在 `Resources/Lua/Test/UITest/Generate/`（此文件夹，可以在 `UITestInitializeParameters` 中设置）下，有一个 `Login.lua` 文件。

点击保存后，随时可以开始新的录制。

注意：同名文件将直接覆盖。

详细介绍：

录制脚本的内容为

--由 `UITest` 自动生成的脚本

```
function UITestFunction_Login()
    --开始测试
    CS.LPCFramework.UITesterManager.StartTestCommandQueue()
    CS.LPCFramework.UITesterManager.EnterSection("Login")

    --等待 1424 毫秒
    CS.LPCFramework.UITesterManager.AddInterval(1424)

    --检查文本框_n0.n6 的文字是否是"光爪游戏"
```

```
CS.LPCFramework.UITesterManager.CheckText("_n0.n6","光爪游戏")
```

```
--检查文本框_n0.n4 的文字是否是"V1.0"
```

```
CS.LPCFramework.UITesterManager.CheckText("_n0.n4","V1.0")
```

```
--检查图片框_n0.n2 的图片是否是"ui://cey8mwyuu5jj0"
```

```
CS.LPCFramework.UITesterManager.CheckImage("_n0.n2","ui://cey8mwyuu5jj0")
```

```
--等待 1509 毫秒后点击_n0.n3
```

```
CS.LPCFramework.UITesterManager.AddInterval(1509)
```

```
CS.LPCFramework.UITesterManager.AddClick("_n0.n3")
```

```
CS.LPCFramework.UITesterManager.FinishTestCommandQueue()
```

end

基本在每条命令前都有注释。其中“_n0.n3”是 FairyGUI 中的层级结构。如果需要修改测试，又不想录脚本，可以直接问 UI 开发程序这些控件的名称。

三 播放

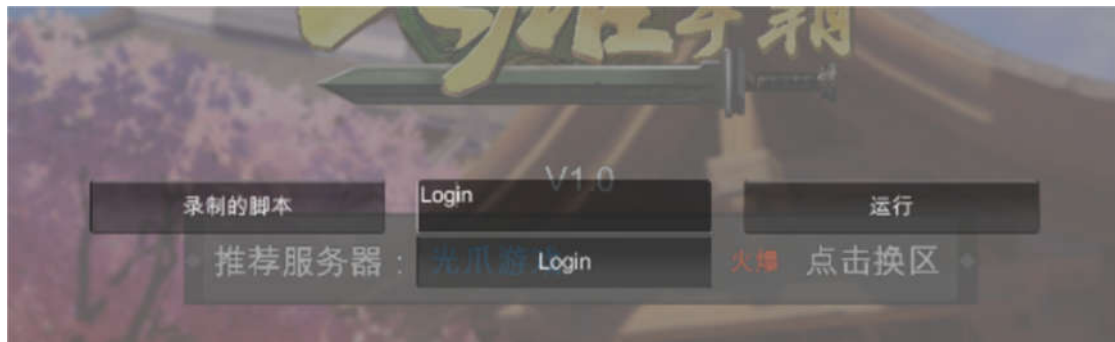
3.1 播放

在 Idle 状态下，按 F7 可以开始播放脚本。

按 F7 后，可以看到



第一个框选择是一个录制的脚本，还是一个合并的脚本。合并脚本见后面第四部分



输入“Log”后，会根据现有的脚本提示一个名字最符合的，点击它以后，“运行”按钮变成白色，点击可以运行。

运行后，会看到蓝色边框的



运行结束后，将会回到这个界面



这个界面点击 F7 能返回 Idel。

3.2 运行时 Log

目前暂时与客户端的日志是一样的。

四 合并脚本

4.1 为什么要合并脚本。

如果仅仅能够重播录制好的脚本，当其中一个流程更换掉，就会需要重新录制大量的脚本。比方说，登陆流程更换了，那么几乎所有的脚本都需要重新录制。这是很难维护的。所以有一个合并脚本的概念。

合并脚本，就是把操作尽量分解为不同界面的切换。比如录制的脚本有这些：

Login:登陆

City1: 从 City 界面出发点击左下角的按钮，后检查一些 UI 元素，后点击返回。回到 City 界面。

City2: 从 City 界面出发点击右下角的按钮，后检查一些 UI 元素，点击返回。回到 City 界面。

...

可以合并为：

Login->City1

Login->City2

Login->City1->City2

这些脚本。

当其中一个流程改变了，仅需要单独录制这个流程，然后替换就可以了。

4.2 合并脚本

在 Idel 界面，按 **F8** 进入到合并脚本界面。如图



合并后的脚本，保存在：
Resources/Test/UITest/ Combined/下。

比如

之前合并了 Login->City2 的脚本，保存为 Comb1 后，有两个文件，一个是 Comb1.lua。

一个是 comb1.txt，内容为

Login, City2

在合并界面，按 F8 可以回到 Idel 界面。

注意：如果要对刚合并的脚本操作（比如加载，比如重新生成所有合并脚本，需要按两次 F8 刷新一下）

4.2.1 详细介绍

新生成的 comb1.lua 的内容为

--由 UITest 自动生成的脚本

```
function UITestFunctionComb_comb1()
    --开始测试
    CS.LPCFramework.UITesterManager.StartTestCommandQueue()

    CS.LPCFramework.UITesterManager.EnterSection("Login")

    --等待 1424 毫秒
    CS.LPCFramework.UITesterManager.AddInterval(1424)

    --检查文本框_n0.n6 的文字是否是"光爪游戏"
    CS.LPCFramework.UITesterManager.CheckText("_n0.n6", "光爪游戏")

    --检查文本框_n0.n4 的文字是否是"V1.0"
    CS.LPCFramework.UITesterManager.CheckText("_n0.n4", "V1.0")

    --检查图片框_n0.n2 的图片是否是"ui://cey8mwyuu5jj0"
    CS.LPCFramework.UITesterManager.CheckImage("_n0.n2", "ui://cey8mwyuu5jj0")

    --等待 1509 毫秒后点击_n0.n3
    CS.LPCFramework.UITesterManager.AddInterval(1509)
    CS.LPCFramework.UITesterManager.AddClick("_n0.n3")

    CS.LPCFramework.UITesterManager.EnterSection("City2")
```

```

--等待 1127 毫秒后点击_n13.n37
CS.LPCFramework.UITesterManager.AddInterval(1127)
CS.LPCFramework.UITesterManager.AddClick("_n13.n37")

--等待 2935 毫秒后点击_n96.n8._n113
CS.LPCFramework.UITesterManager.AddInterval(2935)
CS.LPCFramework.UITesterManager.AddClick("_n96.n8._n113")

--等待 16 毫秒后点击_n96.n8._n113.n16
CS.LPCFramework.UITesterManager.AddInterval(16)
CS.LPCFramework.UITesterManager.AddClick("_n96.n8._n113.n16")

--等待 3627 毫秒后点击_n295.n67
CS.LPCFramework.UITesterManager.AddInterval(3627)
CS.LPCFramework.UITesterManager.AddClick("_n295.n67")

--等待 3081 毫秒后点击_n96.n4
CS.LPCFramework.UITesterManager.AddInterval(3081)
CS.LPCFramework.UITesterManager.AddClick("_n96.n4")

CS.LPCFramework.UITesterManager.FinishTestCommandQueue()

end

```

可以看到，就是将 Login.lua 与 City2.lua 的内容连接在一起。

4.3 加载合并的脚本

加载一个合并的脚本，可以修改它，或者另存为一个新的合并脚本。
在合并脚本界面，填入需要加载的脚本名



注意：加载后，当前正在编辑的脚本序列将被加载的覆盖掉。所以如果需要保存当前编辑的序列，一定要保存后再加载。

4.4 重新生成所有合并脚本

合并脚本最重要的目的是批量重新生成。比如之前有两个合并脚本，分别是

Comb1:Login->City2

Comb2:Login->City2->City3

如果登陆流程发生变化，可以重新录制 Login，覆盖 Login。然后在合并脚本界面点击



后面的数字表示将要发生变化的脚本数量。

之后，Comb1，和 Comb2 的登陆操作将被新录制的 Login 取代。

4.4.1 详细介绍：

他们的内容是

--由 UITest 自动生成的脚本

```
function UITestFunctionComb_comb1()
--开始测试
CS.LPCFramework.UITesterManager.StartTestCommandQueue()

CS.LPCFramework.UITesterManager.EnterSection("Login")

--等待 1424 毫秒
CS.LPCFramework.UITesterManager.AddInterval(1424)

--检查文本框_n0.n6 的文字是否是"光爪游戏"
CS.LPCFramework.UITesterManager.CheckText("_n0.n6","光爪游戏")

--检查文本框_n0.n4 的文字是否是"V1.0"
CS.LPCFramework.UITesterManager.CheckText("_n0.n4","V1.0")

--检查图片框_n0.n2 的图片是否是"ui://cey8mwyuu5jj0"
CS.LPCFramework.UITesterManager.CheckImage("_n0.n2","ui://cey8mwyuu5jj0")

--等待 1509 毫秒后点击_n0.n3
CS.LPCFramework.UITesterManager.AddInterval(1509)
CS.LPCFramework.UITesterManager.AddClick("_n0.n3")

CS.LPCFramework.UITesterManager.EnterSection("City2")

--等待 1127 毫秒后点击_n13.n37
CS.LPCFramework.UITesterManager.AddInterval(1127)
CS.LPCFramework.UITesterManager.AddClick("_n13.n37")
```

```

--等待 2935 毫秒后点击_n96.n8._n113
CS.LPCFramework.UITesterManager.AddInterval(2935)
CS.LPCFramework.UITesterManager.AddClick("_n96.n8._n113")

--等待 16 毫秒后点击_n96.n8._n113.n16
CS.LPCFramework.UITesterManager.AddInterval(16)
CS.LPCFramework.UITesterManager.AddClick("_n96.n8._n113.n16")

--等待 3627 毫秒后点击_n295.n67
CS.LPCFramework.UITesterManager.AddInterval(3627)
CS.LPCFramework.UITesterManager.AddClick("_n295.n67")

--等待 3081 毫秒后点击_n96.n4
CS.LPCFramework.UITesterManager.AddInterval(3081)
CS.LPCFramework.UITesterManager.AddClick("_n96.n4")

CS.LPCFramework.UITesterManager.FinishTestCommandQueue()

end

```

和:

--由 UITest 自动生成的脚本

```

function UITestFunctionComb_comb2()
    --开始测试
    CS.LPCFramework.UITesterManager.StartTestCommandQueue()

    CS.LPCFramework.UITesterManager.EnterSection("Login")

    --等待 1424 毫秒
    CS.LPCFramework.UITesterManager.AddInterval(1424)

    --检查文本框_n0.n6 的文字是否是"光爪游戏"
    CS.LPCFramework.UITesterManager.CheckText("_n0.n6","光爪游戏")

    --检查文本框_n0.n4 的文字是否是"V1.0"
    CS.LPCFramework.UITesterManager.CheckText("_n0.n4","V1.0")

    --检查图片框_n0.n2 的图片是否是"ui://cey8mwyuu5jj0"
    CS.LPCFramework.UITesterManager.CheckImage("_n0.n2","ui://cey8mwyuu5jj0")

```

```
--等待 1509 毫秒后点击_n0.n3
CS.LPCFramework.UITesterManager.AddInterval(1509)
CS.LPCFramework.UITesterManager.AddClick("_n0.n3")
```

```
CS.LPCFramework.UITesterManager.EnterSection("City2")
```

```
--等待 1127 毫秒后点击_n13.n37
CS.LPCFramework.UITesterManager.AddInterval(1127)
CS.LPCFramework.UITesterManager.AddClick("_n13.n37")
```

```
--等待 2935 毫秒后点击_n96.n8._n113
CS.LPCFramework.UITesterManager.AddInterval(2935)
CS.LPCFramework.UITesterManager.AddClick("_n96.n8._n113")
```

```
--等待 16 毫秒后点击_n96.n8._n113.n16
CS.LPCFramework.UITesterManager.AddInterval(16)
CS.LPCFramework.UITesterManager.AddClick("_n96.n8._n113.n16")
```

```
--等待 3627 毫秒后点击_n295.n67
CS.LPCFramework.UITesterManager.AddInterval(3627)
CS.LPCFramework.UITesterManager.AddClick("_n295.n67")
```

```
--等待 3081 毫秒后点击_n96.n4
CS.LPCFramework.UITesterManager.AddInterval(3081)
CS.LPCFramework.UITesterManager.AddClick("_n96.n4")
```

```
CS.LPCFramework.UITesterManager.EnterSection("City3")
```

```
--等待 1738 毫秒后点击_n13.n8
CS.LPCFramework.UITesterManager.AddInterval(1738)
CS.LPCFramework.UITesterManager.AddClick("_n13.n8")
```

```
--等待 4581 毫秒后点击_n13.n4
CS.LPCFramework.UITesterManager.AddInterval(4581)
CS.LPCFramework.UITesterManager.AddClick("_n13.n4")
```

```
--等待 8360 毫秒
CS.LPCFramework.UITesterManager.AddInterval(8360)
```

```
--检查文本框_n295.n33.n21 的文字是否是"[盟友]玩家名字 7 个字"
CS.LPCFramework.UITesterManager.CheckText("_n295.n33.n21","[盟友]玩家名字 7 个字")
```



```
--检查文本框_n295.n33.n19.n2 的文字是否是"选择目标"  
CS.LPCFramework.UITesterManager.CheckText("_n295.n33.n19.n2","选择目标")
```

```
--等待 2341 毫秒后点击_n887.n7  
CS.LPCFramework.UITesterManager.AddInterval(2341)  
CS.LPCFramework.UITesterManager.AddClick("_n887.n7")
```

```
CS.LPCFramework.UITesterManager.FinishTestCommandQueue()
```

End

其中，没有缩小的部分，都是来自于
Login.lua

--由 UITest 自动生成的脚本

```
function UITestFunction_Login()  
    --开始测试  
    CS.LPCFramework.UITesterManager.StartTestCommandQueue()  
    CS.LPCFramework.UITesterManager.EnterSection("Login")  
  
    --等待 1424 毫秒  
    CS.LPCFramework.UITesterManager.AddInterval(1424)  
  
    --检查文本框_n0.n6 的文字是否是"光爪游戏"  
    CS.LPCFramework.UITesterManager.CheckText("_n0.n6","光爪游戏")  
  
    --检查文本框_n0.n4 的文字是否是"V1.0"  
    CS.LPCFramework.UITesterManager.CheckText("_n0.n4","V1.0")  
  
    --检查图片框_n0.n2 的图片是否是"ui://cey8mwyuu5jj0"  
    CS.LPCFramework.UITesterManager.CheckImage("_n0.n2","ui://cey8mwyuu5jj0")  
  
    --等待 1509 毫秒后点击_n0.n3  
    CS.LPCFramework.UITesterManager.AddInterval(1509)  
    CS.LPCFramework.UITesterManager.AddClick("_n0.n3")  
  
    CS.LPCFramework.UITesterManager.FinishTestCommandQueue()  
  
end
```

比方说，现在登录流程换了。于是重新录制了 Login.lua

这是重新录的

--由 UITest 自动生成的脚本

```
function UITestFunction_Login()
    --开始测试
    CS.LPCFramework.UITesterManager.StartTestCommandQueue()
    CS.LPCFramework.UITesterManager.EnterSection("Login")

    --等待 1359 毫秒后点击_n0.n3
    CS.LPCFramework.UITesterManager.AddInterval(1359)
    CS.LPCFramework.UITesterManager.AddClick("_n0.n3")

    CS.LPCFramework.UITesterManager.FinishTestCommandQueue()

end
```

进入到合并脚本页面，点重新生成

重新生成后的 lua 脚本为：

--由 UITest 自动生成的脚本

```
function UITestFunctionComb_comb1()
    --开始测试
    CS.LPCFramework.UITesterManager.StartTestCommandQueue()

    CS.LPCFramework.UITesterManager.EnterSection("Login")

    --等待 1359 毫秒后点击_n0.n3
    CS.LPCFramework.UITesterManager.AddInterval(1359)
    CS.LPCFramework.UITesterManager.AddClick("_n0.n3")

    CS.LPCFramework.UITesterManager.EnterSection("City2")

    --等待 1127 毫秒后点击_n13.n37
    CS.LPCFramework.UITesterManager.AddInterval(1127)
    CS.LPCFramework.UITesterManager.AddClick("_n13.n37")

    --等待 2935 毫秒后点击_n96.n8._n113
    CS.LPCFramework.UITesterManager.AddInterval(2935)
    CS.LPCFramework.UITesterManager.AddClick("_n96.n8._n113")

end
```

```

--等待 16 毫秒后点击_n96.n8._n113.n16
CS.LPCFramework.UITesterManager.AddInterval(16)
CS.LPCFramework.UITesterManager.AddClick("_n96.n8._n113.n16")

--等待 3627 毫秒后点击_n295.n67
CS.LPCFramework.UITesterManager.AddInterval(3627)
CS.LPCFramework.UITesterManager.AddClick("_n295.n67")

--等待 3081 毫秒后点击_n96.n4
CS.LPCFramework.UITesterManager.AddInterval(3081)
CS.LPCFramework.UITesterManager.AddClick("_n96.n4")

CS.LPCFramework.UITesterManager.FinishTestCommandQueue()

end

和

--由 UITest 自动生成的脚本

function UITestFunctionComb_comb2()
    --开始测试
    CS.LPCFramework.UITesterManager.StartTestCommandQueue()

    CS.LPCFramework.UITesterManager.EnterSection("Login")

    --等待 1359 毫秒后点击_n0.n3
    CS.LPCFramework.UITesterManager.AddInterval(1359)
    CS.LPCFramework.UITesterManager.AddClick("_n0.n3")

    CS.LPCFramework.UITesterManager.EnterSection("City2")

    --等待 1127 毫秒后点击_n13.n37
    CS.LPCFramework.UITesterManager.AddInterval(1127)
    CS.LPCFramework.UITesterManager.AddClick("_n13.n37")

    --等待 2935 毫秒后点击_n96.n8._n113
    CS.LPCFramework.UITesterManager.AddInterval(2935)
    CS.LPCFramework.UITesterManager.AddClick("_n96.n8._n113")

    --等待 16 毫秒后点击_n96.n8._n113.n16
    CS.LPCFramework.UITesterManager.AddInterval(16)

```

```
CS.LPCFramework.UITesterManager.AddClick("_n96.n8._n113.n16")
```

```
--等待 3627 毫秒后点击_n295.n67
```

```
CS.LPCFramework.UITesterManager.AddInterval(3627)
```

```
CS.LPCFramework.UITesterManager.AddClick("_n295.n67")
```

```
--等待 3081 毫秒后点击_n96.n4
```

```
CS.LPCFramework.UITesterManager.AddInterval(3081)
```

```
CS.LPCFramework.UITesterManager.AddClick("_n96.n4")
```

```
CS.LPCFramework.UITesterManager.EnterSection("City3")
```

```
--等待 1738 毫秒后点击_n13.n8
```

```
CS.LPCFramework.UITesterManager.AddInterval(1738)
```

```
CS.LPCFramework.UITesterManager.AddClick("_n13.n8")
```

```
--等待 4581 毫秒后点击_n13.n4
```

```
CS.LPCFramework.UITesterManager.AddInterval(4581)
```

```
CS.LPCFramework.UITesterManager.AddClick("_n13.n4")
```

```
--等待 8360 毫秒
```

```
CS.LPCFramework.UITesterManager.AddInterval(8360)
```

```
--检查文本框_n295.n33.n21 的文字是否是"[盟友]玩家名字 7 个字"
```

```
CS.LPCFramework.UITesterManager.CheckText("_n295.n33.n21","[盟友]玩家名字 7 个字")
```

```
--检查文本框_n295.n33.n19.n2 的文字是否是"选择目标"
```

```
CS.LPCFramework.UITesterManager.CheckText("_n295.n33.n19.n2","选择目标")
```

```
--等待 2341 毫秒后点击_n887.n7
```

```
CS.LPCFramework.UITesterManager.AddInterval(2341)
```

```
CS.LPCFramework.UITesterManager.AddClick("_n887.n7")
```

```
CS.LPCFramework.UITesterManager.FinishTestCommandQueue()
```

```
end
```