(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112882950 A
(43)申请公布日 2021.06.01

(21)申请号 202110289363.1

(22)申请日 2021.03.18

(71)申请人 北京字节跳动网络技术有限公司
地址 100041 北京市石景山区实兴大街30号院3号楼2层B-0035房间

(72)发明人 周暄承 陈尹 潘林

(74)专利代理机构 北京锺维联合知识产权代理有限公司 11579
代理人 王越

(51)Int.Cl.
G06F 11/36(2006.01)

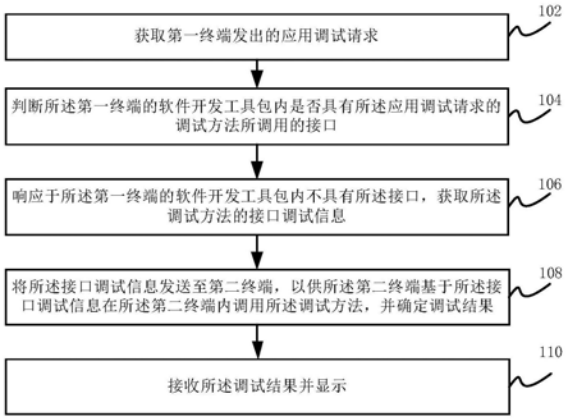
权利要求书3页 说明书11页 附图4页

(54)发明名称

应用调试方法及装置、电子设备和计算机可读存储介质

(57)摘要

本发明提出了一种应用调试方法及装置、电子设备和计算机可读存储介质，其中，该方法包括：获取第一终端发出的应用调试请求；判断所述第一终端的软件开发工具包内是否具有所述应用调试请求的调试方法所调用的接口；响应于所述第一终端的软件开发工具包内不具有所述接口，获取所述调试方法的接口调试信息；将所述接口调试信息发送至第二终端，以供所述第二终端基于所述接口调试信息在所述第二终端内调用所述调试方法，并确定调试结果；接收所述调试结果并显示。通过本发明的技术方案，减少了应用调试过程的人工参与，降低了调试出错率，提升了调试效率。



CN 112882950 A

1.一种应用调试方法,其特征在于,包括:

获取第一终端发出的应用调试请求;

判断所述第一终端的软件开发工具包内是否具有所述应用调试请求的调试方法所调用的接口;

响应于所述第一终端的软件开发工具包内不具有所述接口,获取所述调试方法的接口调试信息;

将所述接口调试信息发送至第二终端,以供所述第二终端基于所述接口调试信息在所述第二终端内调用所述调试方法,并确定调试结果;

接收所述调试结果并显示。

2.根据权利要求1所述的应用调试方法,其特征在于,

所述第一终端为PC端,所述第二终端为移动端,所述软件开发工具包为运行在所述移动端的软件开发工具包。

3.根据权利要求2所述的应用调试方法,其特征在于,所述获取所述调试方法的接口调试信息,包括:

通过所述第一终端中移动端应用的引擎桥接层,获取所述调试方法的接口调试信息;或者

通过hook函数获取所述调试方法的接口调试信息,其中,所述hook函数替换所述应用调试请求的调试方法,用于获取所述接口调试信息和用于被调用后得到所述调试结果。

4.根据权利要求2所述的应用调试方法,其特征在于,

所述接口调试信息包括:所述软件开发工具包所涉及的方法名称、所述软件开发工具包所涉及的方法所用参数名称、参数个数、参数类型和参数值。

5.根据权利要求2至4中任一项所述的应用调试方法,其特征在于,所述将所述接口调试信息发送至第二终端,包括:

通过所述第一终端中移动端应用的通信代理模块,对所述接口调试信息进行序列化处理和发送序列化处理后的接口调试信息;

所述接收所述调试结果并显示,包括:

通过所述通信代理模块,接收所述调试结果和对所述调试结果进行反序列化处理;

向所述第一终端中所述移动端应用的引擎桥接层发送反序列化处理后的调试结果,使所述移动端应用获得反序列化处理后的调试结果并显示。

6.根据权利要求1所述的应用调试方法,其特征在于,还包括:

响应于所述第一终端的软件开发工具包内不具有所述接口,在获取所述调试方法的接口调试信息之同时,暂停当前调试进程;

在接收所述调试结果之后,还包括:

基于所述调试结果,继续所述当前调试进程。

7.一种应用调试方法,其特征在于,包括:

获取接口调试信息,所述接口调试信息为第一终端发出的应用调试请求的调试方法之接口调试信息;

基于所述接口调试信息和第二终端的软件开发工具包,在所述第二终端内调用所述调试方法,并确定调试结果,其中,所述第二终端的软件开发工具包具有所述应用调试请求的

调试方法所调用的接口；

发送所述调试结果。

8.根据权利要求7所述的应用调试方法，其特征在于，

所述第一终端为PC端，所述第二终端为移动端，所述软件开发工具包为运行在移动端的软件开发工具包。

9.根据权利要求8所述的应用调试方法，其特征在于，所述获取接口调试信息，包括：

通过所述第二终端的通信代理模块，接收所述接口调试信息和对所述接口调试信息进行反序列化处理；

所述发送所述调试结果，包括：

通过所述通信代理模块，对所述调试结果进行序列化处理和发送序列化处理后的调试结果。

10.根据权利要求8所述的应用调试方法，其特征在于，

所述接口调试信息包括：所述软件开发工具包所涉及的方法名称、所述软件开发工具包所涉及的方法所用参数名称、参数个数、参数类型和参数值。

11.根据权利要求8至10中任一项所述的应用调试方法，其特征在于，所述发送所述调试结果，包括：

基于所述第二终端的软件开发工具包所指示的发送方式，向所述第一终端发送所述调试结果，其中，

所述第二终端的软件开发工具包所指示的发送方式包括同步方式或异步方式。

12.一种应用调试装置，其特征在于，包括：

第一获取单元，用于获取第一终端发出的应用调试请求；

判断单元，用于判断所述第一终端的软件开发工具包内是否具有所述应用调试请求的调试方法所调用的接口；

第二获取单元，用于响应于所述第一终端的软件开发工具包内不具有所述接口，获取所述调试方法的接口调试信息；

发送单元，用于将所述接口调试信息发送至第二终端，以供所述第二终端基于所述接口调试信息在所述第二终端内调用所述调试方法，并确定调试结果；

接收单元，用于接收所述调试结果并显示。

13.一种应用调试装置，其特征在于，包括：

获取单元，用于获取接口调试信息，所述接口调试信息为第一终端发出的应用调试请求的调试方法之接口调试信息；

调试单元，用于基于所述接口调试信息和第二终端的软件开发工具包，在所述第二终端内调用所述调试方法，并确定调试结果，其中，所述第二终端的软件开发工具包具有所述应用调试请求的调试方法所调用的接口；

发送单元，用于发送所述调试结果。

14.一种电子设备，其特征在于，包括：至少一个处理器；以及，与所述至少一个处理器通信连接的存储器；

其中，所述存储器存储有可被所述至少一个处理器执行的指令，所述指令被设置为用于执行上述权利要求1至11中任一项所述的方法。

15.一种计算机可读存储介质,其特征在于,存储有计算机可执行指令,所述计算机可执行指令用于执行如权利要求1至11中任一项所述的方法流程。

应用调试方法及装置、电子设备和计算机可读存储介质

【技术领域】

[0001] 本发明涉及应用调试技术领域，尤其涉及一种应用调试方法及装置、电子设备和计算机可读存储介质。

【背景技术】

[0002] 随着移动端技术的发展，用于移动端的游戏、社交、理财、购物等手机应用诞生。应用制作方首先需要将移动端的手机应用在PC端进行调试，调试完成后，再导出移动端的手机应用至移动端或移动端模拟器，在移动端或移动端模拟器对该手机应用进行验收。

[0003] 然而，移动端的手机应用需要使用移动端专用的SDK(Software Development Kit，软件开发工具包)，而移动端专用的SDK是无法在PC端进行自动调试的，这就需要应用制作方对移动端专用的SDK进行人工调试或人工将其导出至移动端或移动端模拟器后再调试。这样做导致调试难度大、消耗时间长，严重影响了调试效率。

[0004] 因此，如何提升调试效率，成为目前亟待解决的技术问题。

【发明内容】

[0005] 本发明实施例提供了一种应用调试方法及装置、电子设备和计算机可读存储介质，旨在解决相关技术中对移动端专用的软件开发工具包进行人工调试或人工导出后再调试的方案所导致的调试效率低下的技术问题。

[0006] 第一方面，本发明实施例提供了一种应用调试方法，包括：获取第一终端发出的应用调试请求；判断所述第一终端的软件开发工具包内是否具有所述应用调试请求的调试方法所调用的接口；响应于所述第一终端的软件开发工具包内不具有所述接口，获取所述调试方法的接口调试信息；将所述接口调试信息发送至第二终端，以供所述第二终端基于所述接口调试信息在所述第二终端内调用所述调试方法，并确定调试结果；接收所述调试结果并显示。

[0007] 在本发明上述实施例中，可选地，所述第一终端为PC端，所述第二终端为移动端，所述软件开发工具包为运行在所述移动端的软件开发工具包。

[0008] 在本发明上述实施例中，可选地，所述获取所述调试方法的接口调试信息，包括：通过所述第一终端中移动端应用的引擎桥接层，获取所述调试方法的接口调试信息；或者通过hook函数获取所述调试方法的接口调试信息，其中，所述hook函数替换所述应用调试请求的调试方法，用于获取所述接口调试信息和用于被调用后得到所述调试结果。

[0009] 在本发明上述实施例中，可选地，所述接口调试信息包括：所述软件开发工具包所涉及的方法名称、所述软件开发工具包所涉及的方法所用参数名称、参数个数、参数类型和参数值。

[0010] 在本发明上述实施例中，可选地，所述将所述接口调试信息发送至第二终端，包括：通过所述第一终端中移动端应用的通信代理模块，对所述接口调试信息进行序列化处理和发送序列化处理后的接口调试信息；所述接收所述调试结果并显示，包括：通过所述通

信代理模块，接收所述调试结果和对所述调试结果进行反序列化处理；向所述第一终端中所述移动端应用的引擎桥接层发送反序列化处理后的调试结果，使所述移动端应用获得反序列化处理后的调试结果并显示。

[0011] 在本发明上述实施例中，可选地，还包括：响应于所述第一终端的软件开发工具包内不具有所述接口，在获取所述调试方法的接口调试信息之同时，暂停当前调试进程；在接收所述调试结果之后，还包括：基于所述调试结果，继续所述当前调试进程。

[0012] 第二方面，本发明实施例提供了一种应用调试方法，包括：获取接口调试信息，所述接口调试信息为第一终端发出的应用调试请求的调试方法之接口调试信息；基于所述接口调试信息和第二终端的软件开发工具包，在所述第二终端内调用所述调试方法，并确定调试结果，其中，所述第二终端的软件开发工具包具有所述应用调试请求的调试方法所调用的接口；发送所述调试结果。

[0013] 在本发明上述实施例中，可选地，所述第一终端为PC端，所述第二终端为移动端，所述软件开发工具包为运行在移动端的软件开发工具包。

[0014] 在本发明上述实施例中，可选地，所述获取接口调试信息，包括：通过所述第二终端的通信代理模块，接收所述接口调试信息和对所述接口调试信息进行反序列化处理；所述发送所述调试结果，包括：通过所述通信代理模块，对所述调试结果进行序列化处理和发送序列化处理后的调试结果。

[0015] 在本发明上述实施例中，可选地，所述接口调试信息包括：所述软件开发工具包所涉及的方法名称、所述软件开发工具包所涉及的方法所用参数名称、参数个数、参数类型和参数值。

[0016] 在本发明上述实施例中，可选地，所述发送所述调试结果，包括：基于所述第二终端的软件开发工具包所指示的发送方式，向所述第一终端发送所述调试结果，其中，所述第二终端的软件开发工具包所指示的发送方式包括同步方式或异步方式。

[0017] 第三方面，本发明实施例提供了一种应用调试装置，包括：第一获取单元，用于获取第一终端发出的应用调试请求；判断单元，用于判断所述第一终端的软件开发工具包内是否具有所述应用调试请求的调试方法所调用的接口；第二获取单元，用于响应于所述第一终端的软件开发工具包内不具有所述接口，获取所述调试方法的接口调试信息；发送单元，用于将所述接口调试信息发送至第二终端，以供所述第二终端基于所述接口调试信息在所述第二终端内调用所述调试方法，并确定调试结果；接收单元，用于接收所述调试结果并显示。

[0018] 在本发明上述实施例中，可选地，所述第一终端为PC端，所述第二终端为移动端，所述软件开发工具包为运行在所述移动端的软件开发工具包。

[0019] 在本发明上述实施例中，可选地，所述第二获取单元用于：通过所述第一终端中移动端应用的引擎桥接层，获取所述调试方法的接口调试信息；或者通过hook函数获取所述调试方法的接口调试信息，其中，所述hook函数替换所述应用调试请求的调试方法，用于获取所述接口调试信息和用于被调用后得到所述调试结果。

[0020] 在本发明上述实施例中，可选地，所述接口调试信息包括：所述软件开发工具包所涉及的方法名称、所述软件开发工具包所涉及的方法所用参数名称、参数个数、参数类型和参数值。

[0021] 在本发明上述实施例中，可选地，所述发送单元用于：通过所述第一终端中移动端应用的通信代理模块，对所述接口调试信息进行序列化处理和发送序列化处理后的接口调试信息；所述接收单元用于：通过所述通信代理模块，接收所述调试结果和对所述调试结果进行反序列化处理；向所述第一终端中所述移动端应用的引擎桥接层发送反序列化处理后的调试结果，使所述移动端应用获得反序列化处理后的调试结果并显示。

[0022] 在本发明上述实施例中，可选地，还包括：进程控制单元，用于响应于所述第一终端的软件开发工具包内不具有所述接口，在获取所述调试方法的接口调试信息之同时，暂停当前调试进程，以及用于在接收所述调试结果之后，还包括：基于所述调试结果，继续所述当前调试进程。

[0023] 第四方面，本发明实施例提供了一种应用调试装置，包括：获取单元，用于获取接口调试信息，所述接口调试信息为第一终端发出的应用调试请求的调试方法之接口调试信息；调试单元，用于基于所述接口调试信息和第二终端的软件开发工具包，在所述第二终端内调用所述调试方法，并确定调试结果，其中，所述第二终端的软件开发工具包具有所述应用调试请求的调试方法所调用的接口；发送单元，用于发送所述调试结果。

[0024] 在本发明上述实施例中，可选地，所述第一终端为PC端，所述第二终端为移动端，所述软件开发工具包为运行在移动端的软件开发工具包。

[0025] 在本发明上述实施例中，可选地，所述获取单元用于：通过所述第二终端的通信代理模块，接收所述接口调试信息和对所述接口调试信息进行反序列化处理；所述发送单元用于：通过所述通信代理模块，对所述调试结果进行序列化处理和发送序列化处理后的调试结果。

[0026] 在本发明上述实施例中，可选地，所述接口调试信息包括：所述软件开发工具包所涉及的方法名称、所述软件开发工具包所涉及的方法所用参数名称、参数个数、参数类型和参数值。

[0027] 在本发明上述实施例中，可选地，所述发送单元用于：基于所述第二终端的软件开发工具包所指示的发送方式，向所述第一终端发送所述调试结果，其中，所述第二终端的软件开发工具包所指示的发送方式包括同步方式或异步方式。

[0028] 第五方面，本发明实施例提供了一种电子设备，包括：至少一个处理器；以及，与所述至少一个处理器通信连接的存储器；其中，所述存储器存储有可被所述至少一个处理器执行的指令，所述指令被设置为用于执行上述第一方面和/或第二方面中任一项所述的方法。

[0029] 第六方面，本发明实施例提供了一种计算机可读存储介质，存储有计算机可执行指令，所述计算机可执行指令用于执行上述第一方面和/或第二方面中任一项所述的方法。

[0030] 以上技术方案，针对相关技术中应用调试效率低下的问题，可在需要于PC端调试移动端专用的软件开发工具包时，自动将调试该软件开发工具包所需的接口调试信息发送至移动端进行调试，再直接获得移动端提供的调试结果。由此，减少了应用调试过程的人工参与，降低了调试出错率，提升了调试效率。

【附图说明】

[0031] 为了更清楚地说明本发明实施例的技术方案，下面将对实施例中所需要使用的附

图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其它的附图。

[0032] 图1示出了根据本发明的一个实施例的应用调试方法的流程图；

[0033] 图2示出了根据本发明的另一个实施例的应用调试方法的流程图；

[0034] 图3示出了根据本发明的又一个实施例的应用调试方法的流程图；

[0035] 图4示出了根据本发明的一个实施例的应用调试的示意图；

[0036] 图5示出了根据本发明的另一个实施例的应用调试的示意图；

[0037] 图6示出了根据本发明的一个实施例的应用调试的时序图；

[0038] 图7示出了根据本发明的一个实施例的应用调试装置的框图；

[0039] 图8示出了根据本发明的另一个实施例的应用调试装置的框图；

[0040] 图9示出了根据本发明的一个实施例的电子设备的框图。

【具体实施方式】

[0041] 为了更好的理解本发明的技术方案，下面结合附图对本发明实施例进行详细描述。

[0042] 应当明确，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其它实施例，都属于本发明保护的范围。

[0043] 在本发明实施例中使用的术语是仅仅出于描述特定实施例的目的，而非旨在限制本发明。在本发明实施例和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式，除非上下文清楚地表示其他含义。

[0044] 实施例一

[0045] 图1示出了根据本发明的一个实施例的应用调试方法的流程图。

[0046] 如图1所示，根据本发明的一个实施例的应用调试方法，包括：

[0047] 步骤102，获取第一终端发出的应用调试请求。

[0048] 步骤104，判断所述第一终端的软件开发工具包内是否具有所述应用调试请求的调试方法所调用的接口。

[0049] 其中，第一终端为PC端，所述软件开发工具包为运行在所述移动端的软件开发工具包，而运行在所述移动端的软件开发工具包无法在PC端进行自动调试。当在PC端调试移动端应用时，PC端会发出应用调试请求，为及时识别软件开发工具包可能无法在PC端进行自动调试的问题，获取应用调试请求后，需判断应用调试请求所请求调试的内容是否涉及调试这种运行在所述移动端的软件开发工具包(以下简称软件开发工具包)。

[0050] 具体地，软件开发工具包无法在PC端进行自动调试，也就是说，第一终端发出的应用调试请求所述的、该软件开发工具包所涉及的调试方法和接口，是PC端所无法提供的。此时，可判断第一终端的该软件开发工具包内是否具有所述应用调试请求的调试方法所调用的接口。若判断结果为是，可直接调用该调试方法及对应接口执行应用调试请求所请求调试的内容，反之，若判断结果为否，说明第一终端确实没有调试该软件开发工具包的能力，即无法在第一终端直接调试应用调试请求所请求调试的内容，此时，如下步骤所示，可为其提供有效的调试方式。

[0051] 步骤106,响应于所述第一终端的软件开发工具包内不具有所述接口,获取所述调试方法的接口调试信息。

[0052] 步骤108,将所述接口调试信息发送至第二终端,以供所述第二终端基于所述接口调试信息在所述第二终端内调用所述调试方法,并确定调试结果。

[0053] 步骤110,接收所述调试结果并显示。

[0054] 需要知晓,本技术方案的主体为第一终端本身、与第一终端相连接的调试设备或第一终端内设的调试设备(以下将本技术方案的主体统称为第一终端),那么可将接口调试信息发送至第二终端,由于第二终端为移动端或移动端模拟器,具有调试运行在所述移动端的软件开发工具包的能力,因此,可直接通过第二终端基于所述接口调试信息调用对应的调试方法,最终再将调试结果发送至第一终端,实现第一终端作为PC端对运行在所述移动端的软件开发工具包的自动调试。

[0055] 其中,调试方法的接口调试信息即调试该软件开发工具包所需的信息,包括但不限于所述软件开发工具包所涉及的方法名称、所述软件开发工具包所涉及的方法所用参数名称、参数个数、参数类型和参数值等。

[0056] 实施例二

[0057] 图2示出了根据本发明的另一个实施例的应用调试方法的流程图。

[0058] 如图2所示,根据本发明的另一个实施例的应用调试方法,其流程包括:

[0059] 步骤202,获取第一终端发出的应用调试请求。

[0060] 步骤204,判断所述第一终端的软件开发工具包内是否具有所述应用调试请求的调试方法所调用的接口。

[0061] 步骤206,响应于所述第一终端的软件开发工具包内不具有所述接口,通过所述第一终端中移动端应用的引擎桥接层,获取所述调试方法的接口调试信息;或者,通过hook函数获取所述调试方法的接口调试信息,其中,所述hook函数替换所述应用调试请求的调试方法,用于获取所述接口调试信息和用于被调用后得到所述调试结果。

[0062] 在一种可能的设计中,获取所述调试方法的接口调试信息的方式包括:通过所述第一终端中移动端应用的引擎桥接层,获取所述调试方法的接口调试信息。具体地,第一终端中移动端应用的引擎桥接层(Unity Bridge)可通过插件拦截软件开发工具包的接口调试信息,可快捷实现接口调试信息的获取。

[0063] 在另一种可能的设计中,通过hook函数获取所述调试方法的接口调试信息,其中,所述hook函数替换所述应用调试请求的调试方法,用于获取所述接口调试信息和用于被调用后得到所述调试结果。具体地,通过hook函数替换应用调试请求所述的调试方法,那么hook函数在实现该调试方法的功能同时还能够兼具获取用于该调试方法的接口调试信息功能。由此,通过hook方式即可快捷实现接口调试信息的获取。

[0064] 步骤208,通过所述第一终端中移动端应用的通信代理模块,对所述接口调试信息进行序列化处理和发送序列化处理后的接口调试信息。

[0065] 可选地,通信代理模块为Proxy(代理类),Proxy工作在应用层,用于将待远程传输的信息进行备份,对备份信息进行第一终端不想或不能进行的其他操作。具体地,Proxy可将第一终端不能调试的软件开发工具包的接口调试信息进行备份,并将该备份进行序列化处理后发送至第二终端。。

[0066] 另外，将接口调试信息通过Proxy进行代理联调，还能够减少对第一终端的资源的占用，可提高第一终端的内部访问速度，还能够提升第一终端的安全性，避免针对移动端专用软件开发工具包之接口的非法攻击直接作用于第一终端本身。

[0067] 步骤210，通过所述通信代理模块，接收所述调试结果和对所述调试结果进行反序列化处理。

[0068] 步骤212，向所述第一终端中所述移动端应用的引擎桥接层发送反序列化处理后的调试结果，使所述移动端应用获得反序列化处理后的调试结果并显示。

[0069] 在实际场景中，第一终端和第二终端均设置有多种软件开发工具包方法，因此，无需将软件开发工具包方法在两者间进行传递，而是只将第一终端中运行在移动端的软件开发工具包的接口调试信息发送至第二终端即可。在调用过程中，软件开发工具包方法相当于一个框架，接口调试信息相当于框架的具体设置信息，这样，第二终端即可基于接口调试信息调用对应的调试方法，实现对该软件开发工具包的调试。最终，第一终端通过第二终端对接口调试信息执行调试功能，得到调试结果。

[0070] 在这一过程中，为提升信息安全水平，第一终端通过所述第一终端中移动端应用的通信代理模块向外发送接口调试信息时，可对接口调试信息进行序列化处理，最终发送序列化处理后的接口调试信息。同理，第二终端向所述第一终端反馈调试结果时也是将调试结果进行了序列化处理以保证安全的，具体地，第二终端可通过GBridge(桥接层)调用自身具有的、与所述接口调试信息对应的接口，得到调试结果，再通过GBridge对调试结果进行序列化处理，返回给第一终端。

[0071] 移动端应用的引擎桥接层与移动端应用的引擎相关联，基于上述内容可知，接口调试信息是在移动端应用的引擎桥接层被拦截的，那么，在接收到反序列化处理后的调试结果后，可将其反馈至移动端应用的引擎桥接层。这样，相当于移动端应用的引擎桥接层基于软件开发工具包的接口调试信息自动获取了针对该接口调试信息的调试结果，减少了应用调试过程的人工参与，降低了调试出错率，提升了调试效率。

[0072] 对此，第一终端可通过所述通信代理模块接收所述调试结果和对所述调试结果进行反序列化处理。最终，第一终端将反序列化处理后的调试结果通过自身中移动端应用的引擎桥接层提供给该移动端应用，使该移动端应用最终获得获得反序列化处理后的调试结果并显示，完成对软件开发工具包的自动调试。

[0073] 以上技术方案，可在需要调试移动端专用的软件开发工具包时自动将该软件开发工具包的接口调试信息发送至移动端进行调试，再直接获得移动端提供的调试结果。由此，无需人工操作即可实现在PC端对移动端专用的软件开发工具包的调试，减少了应用调试过程的人工参与，降低了调试出错率，提升了调试效率。

[0074] 实施例三

[0075] 图3示出了根据本发明的又一个实施例的应用调试方法的流程图。

[0076] 如图3所示，根据本发明的又一个实施例的应用调试方法用于第二终端，其流程包括：

[0077] 步骤302，获取接口调试信息，所述接口调试信息为第一终端发出的应用调试请求的调试方法之接口调试信息。

[0078] 步骤304，基于所述接口调试信息和第二终端的软件开发工具包，在所述第二终端

内调用所述调试方法，并确定调试结果，其中，所述第二终端的软件开发工具包具有所述应用调试请求的调试方法所调用的接口。

[0079] 步骤306，发送所述调试结果。

[0080] 第二终端为移动端或移动端模拟器，具有调试运行在移动端的软件开发工具包的能力，因此，可直接通过第二终端基于第一终端发出的应用调试请求的调试方法之接口调试信息来调用对应的调试方法，得到调试结果，最终再将调试结果发送至第一终端，实现第一终端作为PC端对运行在所述移动端的软件开发工具包的自动调试。

[0081] 具体地，步骤302包括：通过所述第二终端的通信代理模块，接收所述接口调试信息和对所述接口调试信息进行反序列化处理；步骤306包括：通过所述通信代理模块，对所述调试结果进行序列化处理和发送序列化处理后的调试结果。

[0082] 第一终端中的Proxy需要与第二终端的通信代理模块进行交互，才能实现对该接口调试信息的联调，因此，第二终端配置有通信代理模块，第二终端通过自身的通信代理模块接收该接口调试信息后，再对该接口调试信息进行调试。

[0083] 可选地，第二终端的通信代理模块为MagicBox，MagicBox作为Proxy的对接对象，具有中继功能，同时，具有降低信号干扰的能力，能够准确有效地为第二终端获接收自第一终端中的接口调试信息。

[0084] 第一终端通过所述第一终端中移动端应用的Proxy向外发送接口调试信息时，可对接口调试信息进行序列化处理，最终发送序列化处理后的接口调试信息。同理，第二终端向所述第一终端反馈调试结果时也是将调试结果进行了序列化处理以保证安全的，具体地，第二终端可通过GBridge(桥接层)调用自身具有的、与所述接口调试信息对应的接口，得到调试结果，再通过GBridge对调试结果进行序列化处理，返回给第一终端。

[0085] 由此，无需人工操作即可实现在PC端对移动端专用的软件开发工具包的调试，减少了应用调试过程的人工参与，降低了调试出错率，提升了调试效率。

[0086] 如图4所示，在游戏调试PC端调试移动端的游戏应用时，遇到调试移动端的游戏应用之移动端软件开发工具包的需求。此时，游戏调试PC端(即第一终端)中的Proxy模块可在游戏调试PC端的引擎桥接层拦截移动端的游戏应用之移动端软件开发工具包的接口调试信息，并将该接口调试信息进行序列化处理后发送至本地/远程移动端(即第二终端)的MagicBox模块。

[0087] 本地/远程移动端的MagicBox模块对接收到的信息进行反序列化处理，得到接口调试信息。进一步地，本地/远程移动端基于接口调试信息调用自身的移动端软件开发工具包接口执行调试，得到调试结果。本地/远程移动端的MagicBox模块对调试结果进行序列化处理，并将序列化处理后的调试结果发送至游戏调试PC端的Proxy模块。

[0088] 其中，游戏调试PC端与远程移动端的通信可通过服务器转发实现。在一种可能的设计中，远程移动端可提供控制路径，外部研发人员可通过该控制路径查看远程移动端基于待调试的移动端软件开发工具包的接口调试信息进行调试的过程和调试结果，增加了移动端软件开发工具包调试过程的可控性，方便了游戏调试PC端的移动端软件开发工具包调试工作，降低了游戏调试的整体难度，提升了游戏调试的整体效率。

[0089] 如图5所示，在游戏调试PC端调试移动端的游戏应用时，遇到调试移动端的游戏应用之移动端软件开发工具包的需求。此时，游戏调试PC端(即第一终端)中的Proxy模块可在

游戏调试PC端的引擎桥接层拦截移动端的游戏应用之移动端软件开发工具包的接口调试信息，并将该接口调试信息进行序列化处理后发送至移动端模拟器(即第二终端)的MagicBox模块。

[0090] 移动端模拟器的MagicBox模块对接收到的信息进行反序列化处理，得到接口调试信息。进一步地，移动端模拟器基于接口调试信息调用自身的移动端软件开发工具包接口执行调试，得到调试结果。移动端模拟器的MagicBox模块对调试结果进行序列化处理，并将序列化处理后的调试结果发送至游戏调试PC端的Proxy模块。

[0091] 在上述任一技术方案的基础上，最终，游戏调试PC端的Proxy模块对接收到的内容进行反序列化处理，得到调试结果，并将调试结果返回至拦截接口调试信息的引擎桥接层。这样，对于引擎桥接层来说，其基于该接口调试信息自动获取了对该接口调试信息的调试结果。

[0092] 需要补充的是，本申请的技术方案可包括：响应于所述第一终端的软件开发工具包内不具有所述接口，在获取所述调试方法的接口调试信息之同时，暂停当前调试进程；在接收所述调试结果之后，还包括：基于所述调试结果，继续所述当前调试进程。

[0093] 具体来说，对于第一终端来说，其用于调试移动端应用的当前调试进程在遭遇自身无法调试的移动端专用软件开发工具包接口时，在将调试工作转移给外部的第二终端的同时，暂停当前调试进程。最终，在接收到来自第二终端的调试结果时，再重启当前调试进程。由此，当前调试进程相当于直接基于待调试的移动端软件开发工具包的接口调试信息获得了基于该接口调试信息的调试结果，避免了应用调试进程因无法调试移动端软件开发工具包接口而遭受负面影响，有助于提升调试结果的可靠性。

[0094] 在上述任一技术方案的基础上，参照图6所示，在游戏调试PC端环境中，PC端通过接口methodA连接Unity Bridge，Unity Bridge通过插件在游戏引擎Unity/Unreal等中拦截待调试的移动端软件开发工具包的接口调试信息，并在PC端将该接口调试信息发送至Proxy，Proxy对其进行序列化并通过服务器Server将序列化的信息转发至移动端环境中的MagicBox。MagicBox调用GBridge对来自Proxy的信息进行处理后，调用Native Module(原生模组)得到调试结果，并将调试结果同步返回或异步返回至PC端。

[0095] 具体地，第二终端可基于所述第二终端的软件开发工具包所指示的发送方式，向所述第一终端发送所述调试结果，其中，所述第二终端的软件开发工具包所指示的发送方式包括同步方式或异步方式。

[0096] 基于此，同步方式指的是将第一终端向第二终端发送待调试的移动端软件开发工具包的接口调试信息至第二终端向第一终端返回调试结果作为一个完整的进程，可保证进程的完整性，降低进程出错的几率。

[0097] 第一终端向第二终端发送待调试的移动端软件开发工具包的接口调试信息和第二终端向第一终端返回调试结果，这两个步骤之间，第二终端需要一定的时间进行调试操作，若将两个步骤连续为一个进程，则进行调试的时间会导致第一终端的应用调试发生卡顿。因此，异步方式将第一终端向第二终端发送待调试的移动端软件开发工具包的接口调试信息和第二终端向第一终端返回调试结果作为两个进程，以提升第一终端进行应用调试的流畅性。

[0098] 图7示出了根据本发明的一个实施例的应用调试装置的框图。

[0099] 如图7所示，根据本发明的一个实施例的应用调试装置700，包括：第一获取单元702，用于获取第一终端发出的应用调试请求；判断单元704，用于判断所述第一终端的软件开发工具包内是否具有所述应用调试请求的调试方法所调用的接口；第二获取单元706，用于响应于所述第一终端的软件开发工具包内不具有所述接口，获取所述调试方法的接口调试信息；发送单元708，用于将所述接口调试信息发送至第二终端，以供所述第二终端基于所述接口调试信息在所述第二终端内调用所述调试方法，并确定调试结果；接收单元710，用于接收所述调试结果并显示。

[0100] 在本发明上述实施例中，可选地，所述第一终端为PC端，所述第二终端为移动端，所述软件开发工具包为运行在所述移动端的软件开发工具包。

[0101] 在本发明上述实施例中，可选地，所述第二获取单元706用于：通过所述第一终端中移动端应用的引擎桥接层，获取所述调试方法的接口调试信息；或者通过hook函数获取所述调试方法的接口调试信息，其中，所述hook函数替换所述应用调试请求的调试方法，用于获取所述接口调试信息和用于被调用后得到所述调试结果。

[0102] 在本发明上述实施例中，可选地，所述接口调试信息包括：所述软件开发工具包所涉及的方法名称、所述软件开发工具包所涉及的方法所用参数名称、参数个数、参数类型和参数值。

[0103] 在本发明上述实施例中，可选地，所述发送单元708用于：通过所述第一终端中移动端应用的通信代理模块，对所述接口调试信息进行序列化处理和发送序列化处理后的接口调试信息；所述接收单元710用于：通过所述通信代理模块，接收所述调试结果和对所述调试结果进行反序列化处理；向所述第一终端中所述移动端应用的引擎桥接层发送反序列化处理后的调试结果，使所述移动端应用获得反序列化处理后的调试结果并显示。

[0104] 在本发明上述实施例中，可选地，还包括：进程控制单元，用于响应于所述第一终端的软件开发工具包内不具有所述接口，在获取所述调试方法的接口调试信息之同时，暂停当前调试进程，以及用于在接收所述调试结果之后，还包括：基于所述调试结果，继续所述当前调试进程。

[0105] 该应用调试装置700使用上述任一实施例所述的方案，因此，具有上述所有技术效果，在此不再赘述。

[0106] 图8示出了根据本发明的另一个实施例的应用调试装置的框图。

[0107] 如图8所示，根据本发明的另一个实施例的应用调试装置800，包括：获取单元802，用于获取接口调试信息，所述接口调试信息为第一终端发出的应用调试请求的调试方法之接口调试信息；调试单元804，用于基于所述接口调试信息和第二终端的软件开发工具包，在所述第二终端内调用所述调试方法，并确定调试结果，其中，所述第二终端的软件开发工具包具有所述应用调试请求的调试方法所调用的接口；发送单元806，用于发送所述调试结果。

[0108] 在本发明上述实施例中，可选地，所述第一终端为PC端，所述第二终端为移动端，所述软件开发工具包为运行在移动端的软件开发工具包。

[0109] 在本发明上述实施例中，可选地，所述获取单元802用于：通过所述第二终端的通信代理模块，接收所述接口调试信息和对所述接口调试信息进行反序列化处理；所述发送单元806用于：通过所述通信代理模块，对所述调试结果进行序列化处理和发送序列化处理

后的调试结果。

[0110] 在本发明上述实施例中，可选地，所述接口调试信息包括：所述软件开发工具包所涉及的方法名称、所述软件开发工具包所涉及的方法所用参数名称、参数个数、参数类型和参数值。

[0111] 在本发明上述实施例中，可选地，所述发送单元806用于：基于所述第二终端的软件开发工具包所指示的发送方式，向所述第一终端发送所述调试结果，其中，所述第二终端的软件开发工具包所指示的发送方式包括同步方式或异步方式。

[0112] 该应用调试装置800使用上述任一实施例所述的方案，因此，具有上述所有技术效果，在此不再赘述。

[0113] 图9示出了根据本发明的一个实施例的电子设备的框图。

[0114] 如图9所示，本发明的一个实施例的电子设备900，包括至少一个存储器902；以及，与所述至少一个存储器902通信连接的处理器904；其中，所述存储器存储有可被所述至少一个处理器904执行的指令，所述指令被设置为用于执行上述任一实施例中所述的方案。因此，该电子设备900具有和上述任一实施例中相同的技术效果，在此不再赘述。

[0115] 本发明实施例的电子设备以多种形式存在，包括但不限于：

[0116] (1)移动通信设备：这类设备的特点是具备移动通信功能，并且以提供话音、数据通信为主要目标。这类终端包括：智能手机(例如iPhone)、多媒体手机、功能性手机，以及低端手机等。

[0117] (2)超移动个人计算机设备：这类设备属于个人计算机的范畴，有计算和处理功能，一般也具备移动上网特性。这类终端包括：PDA、MID和UMPC设备等，例如iPad。

[0118] (3)便携式娱乐设备：这类设备可以显示和播放多媒体内容。该类设备包括：音频、视频播放器(例如iPod)，掌上游戏机，电子书，以及智能玩具和便携式车载导航设备。

[0119] (4)服务器：提供计算服务的设备，服务器的构成包括处理器、硬盘、内存、系统总线等，服务器和通用的计算机架构类似，但是由于需要提供高可靠的服务，因此在处理能力、稳定性、可靠性、安全性、可扩展性、可管理性等方面要求较高。

[0120] (5)其他具有数据交互功能的电子装置。

[0121] 另外，本发明实施例提供了一种计算机可读存储介质，存储有计算机可执行指令，所述计算机可执行指令用于执行上述任一实施例中所述的方法流程。

[0122] 以上结合附图详细说明了本发明的技术方案，针对相关技术中应用调试效率低下的问题，可在需要调试软件开发工具包时自动将待调试的移动端软件开发工具包的接口调试信息发送至外部调试主体进行调试，再直接获得外部调试主体提供的调试结果。由此，减少了应用调试过程的人工参与，降低了调试出错率，提升了调试效率。

[0123] 应当理解，本文中使用的术语“和/或”仅仅是一种描述关联对象的关联关系，表示可以存在三种关系，例如，A和/或B，可以表示：单独存在A，同时存在A和B，单独存在B这三种情况。另外，本文中字符“/”，一般表示前后关联对象是一种“或”的关系。

[0124] 应当理解，尽管在本发明实施例中可能采用术语第一、第二等来描述终端，但这些终端不应限于这些术语。这些术语仅用来将终端彼此区分开。例如，在不脱离本发明实施例范围的情况下，第一终端也可以被称为第二终端，类似地，第二终端也可以被称为第一终端。

[0125] 取决于语境，如在此所使用的词语“如果”可以被解释成为“在……时”或“当……时”或“响应于确定”或“响应于检测”。类似地，取决于语境，短语“如果确定”或“如果检测(陈述的条件或事件)”可以被解释成为“当确定时”或“响应于确定”或“当检测(陈述的条件或事件)时”或“响应于检测(陈述的条件或事件)”。

[0126] 在本发明所提供的几个实施例中，应该理解到，所揭露的系统、装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如，多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，装置或单元的间接耦合或通信连接，可以是电性，机械或其它的形式。

[0127] 另外，在本发明各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用硬件加软件功能单元的形式实现。

[0128] 上述以软件功能单元的形式实现的集成的单元，可以存储在一个计算机可读取存储介质中。上述软件功能单元存储在一个存储介质中，包括若干指令用以使得一台计算机装置(可以是个人计算机，服务器，或者网络装置等)或处理器(Processor)执行本发明各个实施例所述方法的部分步骤。而前述的存储介质包括：U盘、移动硬盘、只读存储器(Read-Only Memory，ROM)、随机存取存储器(Random Access Memory，RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0129] 以上所述仅为本发明的较佳实施例而已，并不用以限制本发明，凡在本发明的精神和原则之内，所做的任何修改、等同替换、改进等，均应包含在本发明保护的范围之内。

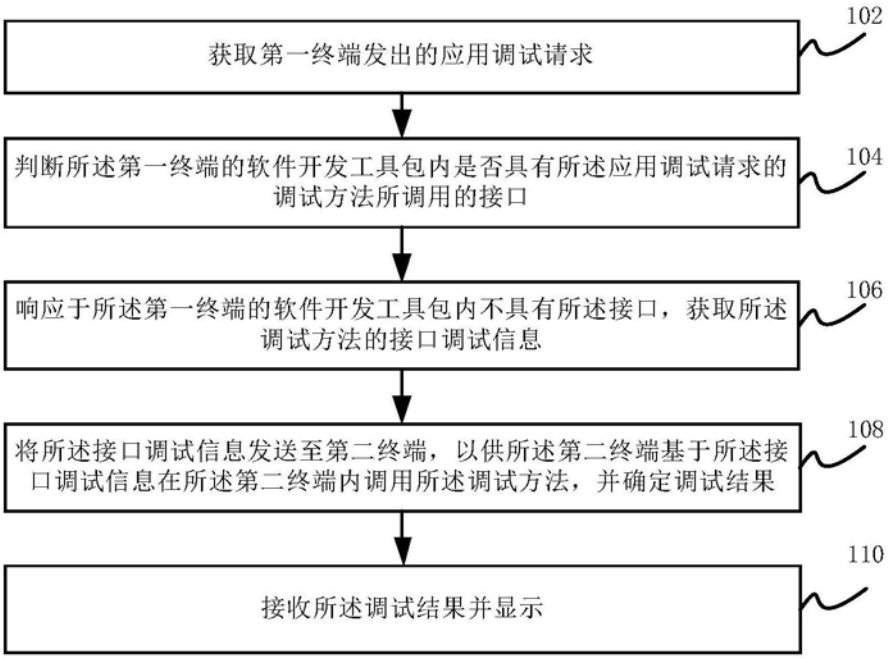

获取第一终端发出的应用调试请求
102
判断所述第一终端的软件开发工具包内是否具有所述应用调试请求的调试方法所调用的接口
104
响应于所述第一终端的软件开发工具包内不具有所述接口，获取所述调试方法的接口调试信息
106
将所述接口调试信息发送至第二终端，以供所述第二终端基于所述接口调试信息在所述第二终端内调用所述调试方法，并确定调试结果
108
接收所述调试结果并显示
110

图1

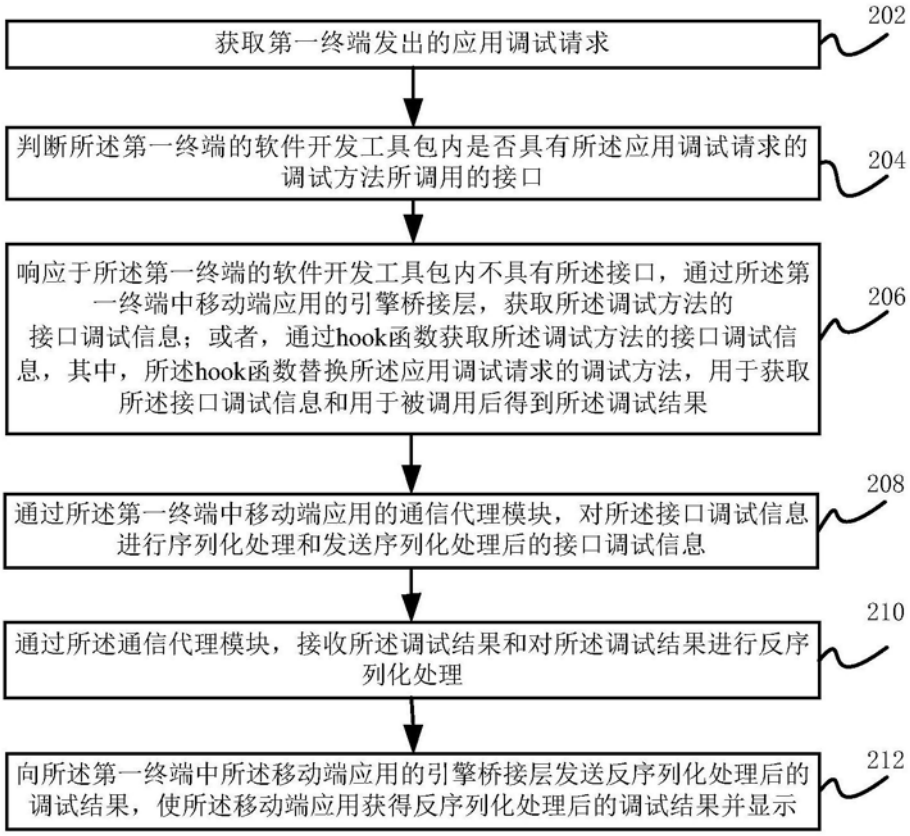

获取第一终端发出的应用调试请求
202
判断所述第一终端的软件开发工具包内是否具有所述应用调试请求的调试方法所调用的接口
204
响应于所述第一终端的软件开发工具包内不具有所述接口，通过所述第一终端中移动端应用的引擎桥接层，获取所述调试方法的接口调试信息；或者，通过hook函数获取所述调试方法的接口调试信息，其中，所述hook函数替换所述应用调试请求的调试方法，用于获取所述接口调试信息和用于被调用后得到所述调试结果
206
通过所述第一终端中移动端应用的通信代理模块，对所述接口调试信息进行序列化处理和发送序列化处理后的接口调试信息
208
通过所述通信代理模块，接收所述调试结果和对所述调试结果进行反序列化处理
210
向所述第一终端中所述移动端应用的引擎桥接层发送反序列化处理后的调试结果，使所述移动端应用获得反序列化处理后的调试结果并显示
212

图2

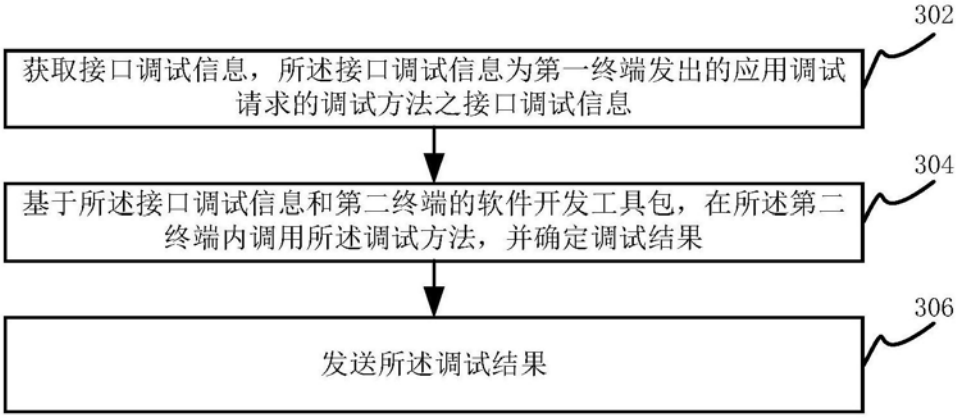

302
获取接口调试信息，所述接口调试信息为第一终端发出的应用调试请求的调试方法之接口调试信息
304
基于所述接口调试信息和第二终端的软件开发工具包，在所述第二终端内调用所述调试方法，并确定调试结果
306
发送所述调试结果

图3

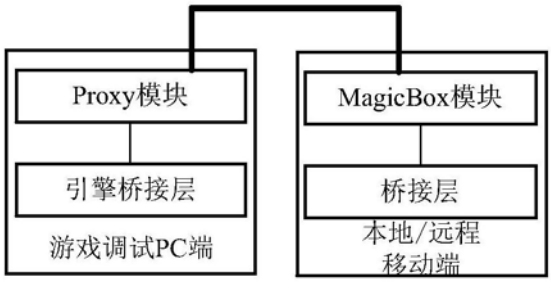

Proxy模块
引擎桥接层
游戏调试PC端
MagicBox模块
桥接层
本地/远程
移动端

图4

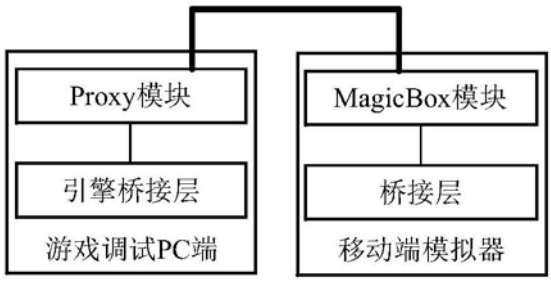

Proxy模块
引擎桥接层
游戏调试PC端
MagicBox模块
桥接层
移动端模拟器

图5

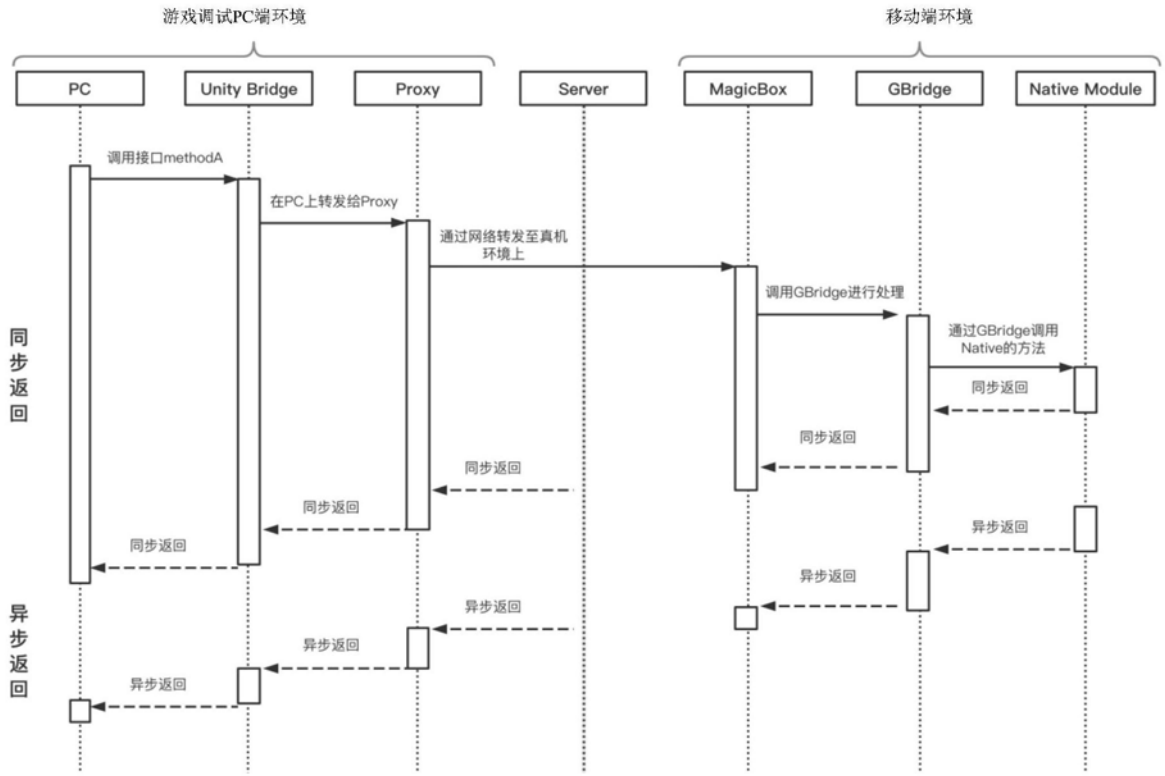

游戏调试PC端环境
移动端环境
PC
Unity Bridge
Proxy
Server
MagicBox
GBridge
Native Module
调用接口methodA
在PC上转发给Proxy
通过网络转发至真机环境上
调用GBridge进行处理
通过GBridge调用Native的方法
同步返回
同步返回
同步返回
同步返回
同步返回
同步返回
异步返回
异步返回
异步返回
异步返回
异步返回
同步返回
异步返回

图6

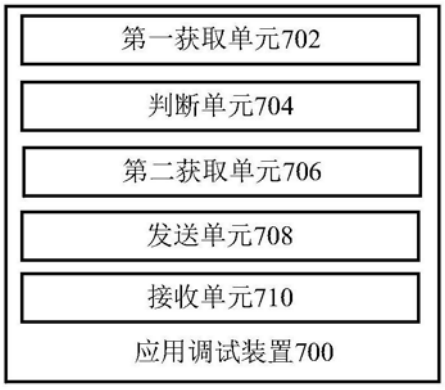

第一获取单元702
判断单元704
第二获取单元706
发送单元708
接收单元710
应用调试装置700

图7

图8

图9