

一种任务调度方法、游戏引擎、设备及存储介质

本申请要求于 2021 年 08 月 16 日提交中国专利局、申请号 202110941214.9、申请名称为“一种任务调度方法、游戏引擎、设备及存储介质”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

5 **技术领域**

本申请涉及计算机技术领域，尤其涉及一种任务调度方法、游戏引擎、设备及存储介质。

背景技术

10 游戏引擎通常是指已编写好的可编辑游戏系统、或者实时交互式的图像应用程序的核心组件，游戏引擎可以为游戏设计者提供编写游戏程序所需的各种工具，从而便于游戏设计者更快地做出游戏程式。游戏引擎通常可支持实现以下功能：物理计算、确定光影效果、碰撞检测、渲染游戏元素、生成音效、运行游戏脚本等等。游戏引擎运行时，其支持的各项功能将相互配合，以实现游戏的正常运行。

15 目前比较常见的游戏引擎在多核中央处理器(Central Processing Unit, CPU)上运行时，通常难以整体控制 CPU 的并行情况，存在某些时段 CPU 的并行度高、而某些时段 CPU 的并行度低的情况，而这极易导致游戏帧率降低、CPU 发热消耗增多等问题发生。

发明内容

本申请实施例提供了一种任务调度方法、游戏引擎、设备及存储介质，能够实现对于 CPU 并行情况的整体控制，提高游戏帧率，减少 CPU 发热消耗。

20 有鉴于此，本申请第一方面提供了一种任务调度方法，所述方法由终端设备执行，所述终端设备用于支持游戏引擎运行，所述游戏引擎包括任务调度系统和多个任务处理系统；所述多个任务处理系统各自负责处理的任務不同，每个所述任务处理系统配置有对应的任务处理阶段、以及其在所对应的任务处理阶段中的运行关联关系，所述运行关联关系用于表征所述任务处理系统与所述游戏引擎中其它任务处理系统之间的运行依赖关系；所述方法包括：

25 当所述游戏引擎工作在待执行阶段时，各特定任务处理系统分别生成各自负责处理的待执行任务，并将所述待执行任务传输给所述任务调度系统；所述特定任务处理系统是所述游戏引擎中所配置的任务处理阶段为所述待执行阶段的任务处理系统；

30 针对来自各所述特定任务处理系统的所述待执行任务，所述任务调度系统基于中央处理器 CPU 中各处理线程的负载情况和所述特定任务处理系统在所述待执行阶段中的运行关联关系，为所述待执行任务分配所述 CPU 中的处理线程。

35 本申请第二方面提供了一种游戏引擎，所述游戏引擎运行在终端设备上，所述游戏引擎包括任务调度系统和多个任务处理系统；所述多个任务处理系统各自负责处理的任務不同，每个所述任务处理系统配置有对应的任务处理阶段、以及其在所对应的任务处理阶段中的运行关联关系，所述运行关联关系用于表征所述任务处理系统与所述游戏引擎中其它任务处理系统之间的运行依赖关系；

所述任务处理系统，用于当所述游戏引擎工作在其对应的任务处理阶段时，生成其负责处理的任务，并将所述任务传输给所述任务调度系统；

所述任务调度系统，用于当所述游戏引擎工作在待执行阶段时，针对来自各特定任务处理系统的待执行任务，基于 CPU 中各处理线程的负载情况和所述特定任务处理系统在所述待执行阶段中的运行关联关系，为所述待执行任务分配所述 CPU 中的处理线程；所述特定任务处理系统是所述游戏引擎中所配置的任务处理阶段为所述待执行阶段的任务处理系统，所述待执行任务是所述特定任务处理系统生成的其负责处理的任务。

本申请第三方面提供了一种设备，所述设备包括处理器以及存储器：

所述存储器用于存储计算机程序；

所述处理器用于根据所述计算机程序，执行上述第一方面所述的任务调度方法的步骤。

本申请第四方面提供了一种计算机可读存储介质，所述计算机可读存储介质用于存储计算机程序，所述计算机程序用于执行上述第一方面所述的任务调度方法的步骤。

本申请第五方面提供了一种计算机程序产品或计算机程序，该计算机程序产品或计算机程序包括计算机指令，该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令，处理器执行该计算机指令，使得该计算机设备执行上述第一方面所述的任务调度方法的步骤。

从以上技术方案可以看出，本申请实施例具有以下优点：

本申请实施例提供了一种任务调度方法，该方法提出了游戏引擎层级的处理资源调度方式，即在游戏运行过程所涉及的每个任务处理阶段内，综合考虑游戏引擎在该任务处理阶段内产生的各个任务之间的关联关系，进而据此合理地针对各个任务分配处理资源。本申请实施例提供的任务调度方法，应用于基于 ECS（Entity-Component-System）架构搭建的游戏引擎，该种游戏引擎将游戏中的各种处理逻辑解耦开来，利用若干独立的任务处理系统分别执行不同的任务；本申请创新性地针对该种游戏引擎中的每个任务处理系统配置了其适用的任务处理阶段、以及其在所适用的任务处理阶段中的运行关联关系，并且还在该种游戏引擎中增设了用于综合调度处理资源的任务调度系统。当游戏引擎工作在某个任务处理阶段例如待执行阶段时，任务调度系统可以根据适用于该待执行阶段的特定任务处理系统的运行关联关系，将 CPU 中的处理线程合理地分配给适用于该待执行阶段的特定任务处理系统生成的待执行任务，实现从游戏引擎层面出发的处理资源配置。如此，综合考虑一个任务处理系统与游戏引擎中其它任务处理系统之间的运行依赖关系来分配处理资源，可以实现对 CPU 并行情况的整体控制，从而有助于提高游戏帧率，降低 CPU 的发热损耗。

附图说明

图 1 为本申请实施例提供的任务调度方法的应用场景示意图；

图 2 为本申请实施例提供的游戏引擎的工作原理示意图；

图 3 为本申请实施例提供的预更新阶段、更新阶段和更新完成阶段的执行顺序示意图；

图 4 为本申请实施例提供的任务调度方法的流程示意图；

图 5 为本申请实施例提供的任务调度系统的执行过程示意图；

图 6 为本申请实施例提供的游戏引擎基于本申请实施例提供的任务调度方法运行时对于 CPU 处理资源的占用情况示意图；

图 7 为本申请实施例提供的游戏引擎的结构示意图；

图 8 为本申请实施例提供的终端设备的结构示意图。

具体实施方式

为了使本技术领域的人员更好地理解本申请方案，下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”、“第三”、“第四”等（如果存在）是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换，以便这里描述的本申请的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外，术语“包括”和“具有”以及他们的任何变形，意图在于覆盖不排他的包含，例如，包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元，而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

相关技术中，游戏引擎运行时，通常只关注任务层级上的处理资源调度情况，即仅针对当前执行的任务中的子任务进行处理资源的配置，而不从游戏引擎整体运行的角度进行处理资源的配置。如此，无法对 CPU 的并行情况进行整体控制，容易导致游戏帧率降低、CPU 的发热损耗增多等问题发生。

针对上述相关技术存在的问题，本申请实施例提供了一种任务调度方法，该方法可以实现对于 CPU 的并行情况的整体控制，从而提高游戏帧率，降低 CPU 的发热消耗。

示例性的，本申请实施例提供的任务调度方法应用于 ECS 架构的游戏引擎，该游戏引擎中包括任务调度系统和多个任务处理系统；这多个任务处理系统各自负责处理的任务不同，且每个任务处理系统均配置有其对应的任务处理阶段、以及其在所对应的任务处理阶段中的运行关联关系，该运行关联关系用于表征任务处理系统与游戏引擎中其它任务处理系统之间的运行依赖关系。当游戏引擎工作在待执行阶段时，各特定任务处理系统分别生成各自负责处理的待执行任务，并将所生成的待执行任务传输给任务调度系统，此处的特定任务处理系统是游戏引擎中所对应的任务处理阶段为该待执行阶段的任务处理系统；针对来自各特定任务处理系统的待执行任务，任务调度系统可以基于 CPU 中各处理线程的负载情况和特定任务处理系统在该待执行阶段中的运行关联关系，为待执行任务分配该 CPU 中的处理线程。

上述任务调度方法提出了游戏引擎层级的处理资源调度方式，即在游戏运行过程所涉及的每个任务处理阶段内，综合考虑游戏引擎在该任务处理阶段内产生的各个任务之间的运行关联关系，进而，据此合理地针对各个任务分配处理资源。该任务调度方法应用于基于 ECS 架构搭建的游戏引擎，该种游戏引擎将游戏中的各种处理逻辑解耦开来，利用若干独立的任务处理系统分别执行不同的任务；本申请在该种游戏引擎的基础上，创新性地针对其中每个任务处理系统配置了其适用的任务处理阶段、以及其在所适用的任务处理阶段中的运行关联关系，并且还在该种游戏引擎中增设了用于综合调度处理资源的任务调度系统。当游戏引擎工作在某个任务处理阶段例如待执行阶段时，任务调度系统可以根据适用

于该待执行阶段的特定任务处理系统的运行关联关系，将 CPU 中的处理线程合理地分配给适用于该待执行阶段的特定任务处理系统生成的待执行任务，实现从游戏引擎层面出发的处理资源配置。如此，综合考虑一个任务处理系统与游戏引擎中其它任务处理系统之间的运行依赖关系来分配处理资源，可以实现对 CPU 并行情况的整体控制，从而有助于提高游戏帧率，降低 CPU 的发热损耗。

应理解，本申请实施例提供的任务调度方法可以应用于支持游戏引擎运行的终端设备，该终端设备具体可以为智能手机、计算机、平板电脑、个人数字助理（Personal Digital Assistant, PDA）、车载终端等等，本申请在此不对该任务调度方法所适用的终端设备做具体限定。

为了便于理解本申请实施例提供的任务调度方法，下面对本申请实施例提供的任务调度方法的应用场景进行示例性介绍。

参见图 1，图 1 为本申请实施例提供的任务调度方法的应用场景示意图。如图 1 所示，该应用场景中包括终端设备 110，终端设备 110 可以是智能手机、计算机等可支持游戏应用程序运行的设备，应理解，图 1 中示出的终端设备 110 仅为示例。终端设备 110 中运行有游戏应用程序，该游戏应用程序运行时其游戏引擎中的各个部件将相互配合，以实现该游戏应用程序的各项功能。

终端设备 110 中运行的游戏应用程序的游戏引擎可以包括任务调度系统 111 和多个任务处理系统 112。其中，任务调度系统 111 用于在游戏引擎运行时为其中各个任务处理系统 112 生成的任务分配 CPU 中的处理线程；任务处理系统 112 用于在游戏引擎运行时生成其负责处理的任務，各个任务处理系统 112 各自负责处理的任務不同。在一种可能的实现方式中，终端设备 110 中运行的游戏应用程序的游戏引擎可以是 ECS 架构的游戏引擎。

在本申请实施例中，在游戏引擎中注册任务处理系统 112 时，需要针对任务处理系统 112 配置其适用的任务处理阶段，作为该任务处理系统 112 对应的任务处理阶段。示例性的，任务处理阶段可以根据对于游戏中游戏帧的处理过程划分得到，例如，可以将游戏帧的处理过程划分为预更新（PreUpdate）阶段、更新（Update）阶段和更新完成（PostUpdate）阶段，相应地，在游戏引擎中注册任务处理系统 112 时，可以配置该任务处理系统 112 具体适用于预更新阶段、更新阶段和更新完成阶段中的哪个阶段或者哪些阶段。此外，在游戏引擎中注册任务处理系统 112 时，还需要针对任务处理系统 112 配置其在对应的任务处理阶段中的运行关联关系，该运行关联关系用于表征该任务处理系统 112 与游戏引擎中其它任务处理系统 112 之间的运行依赖关系，例如，对于用于执行渲染任务的任务处理系统，其需要依赖用于执行物理计算任务的任务处理系统的处理结果执行渲染任务，相应地在游戏引擎中注册用于执行渲染任务的任务处理系统时，可以基于上述工作关系配置该任务处理系统的运行关联关系。

在游戏应用程序实际运行的过程中，当游戏引擎工作至待执行阶段时，各特定任务处理系统可以分别生成各自负责处理的待执行任务，并将生成的待执行任务传输给任务调度系统 111。此处的特定任务处理系统可以是游戏引擎中所配置的任务处理阶段为该待执行阶段的任务处理系统 112，例如，当游戏引擎工作至某游戏帧对应的预更新阶段（即待执行阶

段)时,游戏引擎中所配置的任务处理阶段为预更新阶段的任务处理系统即为特定任务处理系统。

任务调度系统 111 接收到来自各特定任务处理系统的各待执行任务后,针对来自各特定任务处理系统的待执行任务,任务调度系统 111 可以基于终端设备 110 中用于支持游戏应用程序运行的 CPU 中的各处理线程的负载情况,以及特定任务处理系统在该待执行阶段中的运行关联关系,为待执行任务分配该 CPU 中的处理线程。需要说明的是,当支持游戏应用程序运行的 CPU 为多核同构 CPU 时,任务调度系统 111 可以基于 CPU 中的处理线程负载均衡的原则,为待执行任务分配该 CPU 中的处理线程;当支持游戏应用程序运行的 CPU 为多核异构 CPU 时,任务调度系统 111 可以综合考虑 CPU 中各内核各自的数据处理能力,为待执行任务分配该 CPU 中的处理线程。

应理解,图 1 所示的应用场景仅为示例,在实际应用中,本申请实施例提供的任务调度方法还可以应用于其它场景,在此不对本申请实施例提供的任务调度方法适用的应用场景做任何限定。

为了便于理解本申请实施例提供的任务调度方法,下面在介绍本申请实施例提供的任务调度方法前,先对该任务调度方法适用的游戏引擎进行介绍。参见图 2,图 2 为本申请实施例提供的任务调度方法所适用的游戏引擎的工作原理示意图。

如图 2 所示,本申请实施例提供的任务调度方法所适用的游戏引擎为 ECS 架构的游戏引擎,ECS 架构的模式遵循组合优于继承的原则。在 ECS 架构中,游戏中的每个基本单元均可以被视为一个实体(Entity),此处游戏中的基本单元可以理解为游戏中每个独立的虚拟元素,虚拟元素可以是指游戏世界中可能出现的各种元素,如游戏世界中的虚拟角色、虚拟道具、虚拟装饰元素等等。每个实体可以包括至少一种组件(Component)数据,每种组件数据实质上是其所属的实体对应的虚拟元素的功能属性数据,组件数据通常只包括对于功能属性的描述数据,而不包括任何方法逻辑数据。ECS 架构中的系统(System),也即本申请实施例中的任务处理系统,是用于根据实体包括的组件数据操控实体对应的虚拟元素的工具,其只拥有行为,而不拥有任何状态数据。

ECS 架构中的实体其实是一种概念上的定义,其用于指代游戏中存在的虚拟元素,其中包括至少一种组件数据。为了区分游戏中不同的虚拟元素,在代码层面可以针对每个实体配置对应的唯一标识(Identity document, ID),该实体中包括的所有组件数据也会被该实体对应的 ID 标记,以便明确组件数据与实体之间的所属关系。如图 2 所示,在游戏引擎中,通常由实体管理器(Entity Manager)管理游戏引擎中的各个实体,执行上述配置 ID 的操作。

由于 ECS 架构中的实体是组件数据的集合,因此,游戏引擎运行时可以很方便地对实体中的组件数据进行动态编辑,如为实体增加新的组件数据,或者将组件数据从实体中移除。例如,假设玩家控制的虚拟角色因某些原因丧失了移动能力,则游戏引擎可以简单地将用于描述移动功能的组件数据从该虚拟角色对应的实体中移除,如此即可达到使该虚拟角色无法移动的效果。

ECS 架构中的组件数据是其所属的实体对应的虚拟元素的功能属性数据，其只用于存储状态，而不表征任何行为。实体中包括的组件数据可以描述该实体对应的虚拟元素所具备的功能。例如，假设一个实体包括位置组件数据，则表示该实体对应的虚拟元素在游戏世界中占据一定的区域位置，又例如，假设一个实体包括与移动功能相关的组件数据，则表示该实体对应的虚拟元素具备移动能力。如图 2 所示，在游戏引擎中，通常由组件管理器（Component Manager）管理游戏引擎中各实体的组件数据，其可以为组件数据分配内存及控制器，并且管理游戏引擎中所有的组件数据。

ECS 架构中的系统（也即本申请实施例中的任务处理系统）是游戏逻辑的执行工具，其可以操控拥有组件数据的实体，其只能执行行为，而不存储任何状态数据。在 ECS 架构中为了便于独立地实现各种游戏逻辑，其将游戏所支持的各种游戏逻辑解耦开来，分配到不同的任务处理系统上，即由不同的任务处理系统执行不同的游戏逻辑。任务处理系统执行其负责处理的任务时，可以遍历游戏引擎中各实体的组件数据，确定与其负责处理的任务相关的实体，进而基于此类实体中的组件数据执行对应的任务。例如，对于用于实现移动功能的任务处理系统，其可以遍历游戏引擎中的所有实体，以从中查找到拥有与移动功能相关的组件数据的实体，进而基于此类实体中与移动功能相关的组件数据，控制该实体对应的虚拟元素移动。如图 2 所示，在游戏引擎中，通常由系统管理器（System Manager）管理游戏引擎中的任务处理系统，其可以控制各任务处理系统相应地执行对应的游戏逻辑，基于实体中包括的组件数据操控游戏中虚拟元素。此外，该游戏引擎还可以通过事件管理器（Event Manager）实现各任务处理系统间消息的交互。

本申请实施例在上文介绍的 ECS 架构的游戏引擎的基础上，进一步增设了用于为游戏引擎运行过程中生成的任务调度处理资源的任务调度系统，该任务调度系统的具体工作方式将在下文的方法实施例中详细介绍。

此外，本申请实施例还在上文介绍的 ECS 架构的游戏引擎的基础上，为该游戏引擎中的任务处理系统配置了对应的任务处理阶段、以及该任务处理系统在其对应的任务处理阶段中的运行关联关系，该运行关联关系能够表征该任务处理系统与游戏引擎中其它任务处理系统之间的运行依赖关系。

在一种可能的是实现方式中，本申请实施例可以根据游戏引擎的实际工作方式，将游戏引擎的处理过程划分为若干任务处理阶段。在游戏引擎中注册任务处理系统时，可以确定任务处理系统负责处理的任務所适用的任务处理阶段，作为该任务处理系统对应的任务处理阶段。

示例性的，本申请实施例中的任务处理阶段可以根据对于游戏帧的处理过程划分，所划分出的任务处理阶段可以包括游戏帧对应的预更新（PreUpdate）阶段、更新（Update）阶段以及更新完成（PostUpdate）阶段；其中，预更新阶段用于执行更新游戏帧所需的准备任务，更新阶段用于执行对于游戏帧的更新任务，更新完成阶段用于执行更新完游戏帧后所需的收尾任务。

考虑到游戏正常运行时游戏引擎通常以游戏帧为单位执行相关处理任务，因此，本申请实施例可以将对于一个游戏帧的处理过程划分为预更新阶段、更新阶段和更新完成阶段，

这三个阶段的执行顺序如图 3 所示,即先执行预更新阶段,再执行更新阶段,最后执行更新完成阶段。

在游戏帧对应的预更新阶段内,游戏引擎可以执行更新该游戏帧所需的准备任务;例如,游戏引擎中用于执行脚本逻辑的任务处理系统可以执行触发更新物体位置、触发播放特定动画效果等操作,游戏引擎中用于执行物理计算任务的任务处理系统可以执行计算虚拟元素的运动参数的操作,等等。在游戏帧对应的更新阶段,游戏引擎可以执行对于该游戏帧的更新任务;例如,游戏引擎中用于执行渲染任务的任务处理系统可以执行对于虚拟元素的渲染操作,等等。在游戏帧对应的更新完成阶段,游戏引擎可以执行更新完该游戏帧后所需的收尾工作;例如,游戏引擎中用于执行脚本逻辑的任务处理系统可以执行对于该游戏帧内的相关数据的统计操作,等等。

在所划分出的任务处理阶段包括预更新阶段、更新阶段和更新完成阶段的情况下,在游戏引擎中注册任务处理系统时,可以根据该任务处理系统所负责处理的任务的执行时机,相应地为该任务处理系统配置对应的任务处理阶段。例如,针对用于执行渲染任务的任务处理阶段,可以配置其对应的任务处理阶段为更新阶段;针对用于执行物理计算任务的任务处理系统,可以配置其对应的任务处理阶段为预更新阶段;针对用于执行脚本逻辑的任务处理系统,可以配置其对应的任务处理阶段为预更新阶段和更新完成阶段。

应理解,针对一个任务处理系统,可以根据其负责处理的任务的执行时机,为其配置一个或者多个对应的任务处理阶段,本申请在此不对一个任务处理系统对应的任务处理阶段的数目做任何限定。

如此,按照对于游戏帧的处理过程划分任务处理阶段,并相应地为任务处理系统配置对应的任务处理阶段,可以保证针对任务处理系统配置的任务处理阶段比较合理,并且为针对任务处理系统配置运行关联关系提供了较好的配置基础条件,可以更明确所配置的运行关联关系具体适用的时机。

本申请实施例除了针对任务处理系统配置对应的任务处理阶段外,还会针对任务处理系统配置其在所对应的任务处理阶段中的运行关联关系,运行关联关系能够表征任务处理系统与游戏引擎中其它的任务处理系统之间的运行依赖关系。示例性的,若任务处理系统 A 在其对应的任务处理阶段内需要依赖任务处理系统 B 的处理结果执行其负责处理的业务,则可以针对该任务处理系统 A 配置用于表征其需要依赖任务处理系统 B 的处理结果执行任务的运行关联关系;如此,指定每个任务处理系统在其对应的任务处理阶段内的运行关联关系,可以方便任务调度系统根据该运行关联关系,合理地为任务处理阶段内各任务处理系统生成的任务分配处理线程。

应理解,若某任务处理系统在其对应的任务处理阶段内与其它任务处理系统之间不具有运行依赖关系,即该任务处理系统可以独立地执行其负责的任务,而不需要依赖其它任务处理系统的处理结果,则可以针对该任务处理系统配置用于表征其不需要依赖其它任务处理系统的运行关联关系。本申请在此不对针对任务处理系统配置的运行关联关系做任何限定。

在了解了本申请实施例适用的游戏引擎、以及针对该游戏引擎中各任务处理系统配置的基础信息的情况下，下面通过方法实施例对本申请实施例提供的任务调度方法进行详细介绍。参见图 4，图 4 为本申请实施例提供的任务调度方法的流程示意图。如图 4 所示，该任务调度方法包括以下步骤：

5 步骤 401：当所述游戏引擎工作在待执行阶段时，各特定任务处理系统分别生成各自负责处理的待执行任务，并将所述待执行任务传输给所述任务调度系统；所述特定任务处理系统是所述游戏引擎中所配置的任务处理阶段为所述待执行阶段的任务处理系统。

10 在游戏运行的过程中，当游戏引擎工作至待执行阶段时，游戏引擎中的各特定任务处理系统将分别生成其各自负责处理的待执行任务，并将生成的待执行任务传送给游戏引擎中的任务调度系统。此处的特定任务处理系统是指游戏引擎中所配置的任务处理阶段为待执行阶段的任务处理系统。

15 示例性的，假设本申请实施例中的任务处理阶段是根据对于游戏帧的处理过程划分的，包括预更新阶段、更新阶段以及更新完成阶段，则当游戏引擎工作至某游戏帧对应的预更新阶段时，即当游戏引擎工作至需要执行更新某游戏帧所需的准备工作的阶段时，游戏引擎中任务处理阶段为预更新阶段的任务处理系统皆为特定任务处理系统，此时这些特定任务处理系统需要相应地生成各自负责处理的待执行任务，并将所生成的待执行任务传输给任务调度系统。例如，游戏引擎中任务处理阶段为预更新阶段的任务处理系统包括任务处理系统 A、任务处理系统 B 和任务处理系统 C，当游戏引擎工作至某游戏帧对应的预更新阶段时，任务处理系统 A、任务处理系统 B 和任务处理系统 C 可以作为特定任务处理系统，
20 分别生成待执行任务 a、待执行任务 b 和待执行任务 c，任务处理系统 A 可以将其生成的待执行任务 a 传送给任务调度系统，任务处理系统 B 可以将其生成的待执行任务 b 传送给任务调度系统，任务处理系统 C 可以将其生成的待执行任务 c 传送给任务调度系统。

25 应理解，在实际应用中，适用于待执行阶段的特定任务处理系统可以包括一个，也可以包括多个，本申请在此不对游戏引擎中在待执行阶段工作的特定任务处理系统的数量做任何限定。

30 在一种可能的实现方式中，本申请实施例可以为任务处理系统提供便捷的应用程序接口（Application Programming Interface，API），以便任务处理系统可以通过该 API 向游戏引擎中的任务调度系统传输其生成的任务。即各特定任务处理系统生成其各自负责处理的待执行任务后，可以通过本申请实施例提供的 API，将所生成的待执行任务传输给任务调度系统。

35 本申请实施例可以在游戏引擎中新增任务调度系统的基础上，进一步针对游戏引擎中的各个任务处理系统，配置用于支持其与任务调度系统进行通信的 API。示例性的，可以针对游戏引擎中的各个任务处理系统配置其各自对应的 API，每个任务处理系统对应的 API 专用于支持该任务处理系统与任务调度系统进行通信；或者，也可以针对游戏引擎中的多个任务处理系统配置一个 API，该 API 可以支持这些任务处理系统与任务调度系统进行通信。本申请在此不对 API 的配置方式做任何限定。

针对游戏引擎中的各个任务处理系统配置对应的 API 后,任务处理系统一旦生成所需处理的任务,即可通过其对应的 API 将所生成的任务传输给任务调度系统。如此,提高了任务处理系统与任务调度系统之间通信的便利性,并且每个任务处理系统通过 API 将其生成的任务传输给任务调度系统,无需考虑自身与其它任务处理系统之间的复杂的依赖关系。

- 5 步骤 402: 针对来自各所述特定任务处理系统的所述待执行任务,所述任务调度系统基于 CPU 中各处理线程的负载情况和所述特定任务处理系统在所述待执行阶段中的运行关联关系,为所述待执行任务分配所述 CPU 中的处理线程。

- 10 任务调度系统接收到特定任务处理系统传输过来的待执行任务后,可以基于支持游戏运行的 CPU 中的各个处理线程当前的负载情况和特定任务处理系统在待执行阶段中的运行关联关系,为待执行任务分别分配该 CPU 中的处理线程。

- 示例性的,假设任务调度系统接收到任务处理系统 A、任务处理系统 B 和任务处理系统 C 分别传来的待执行任务 a、待执行任务 b 和待执行任务 c,则任务调度系统可以根据当前支持游戏运行的 CPU 中各个处理线程的负载情况、以及任务处理系统 A、任务处理系统 B 和任务处理系统 C 各自在当前所处的待执行阶段中的运行关联关系,为待执行任务 a、待执行任务 b 和待执行任务 c 分别分配对应的处理线程。例如,假设任务处理系统 B 在待执行阶段内需要依赖任务处理系统 A 的处理结果执行其负责处理的待执行任务 b,任务处理系统 C 在待执行阶段内无需依赖其它任务处理系统的处理结果,则任务调度系统可以先为待执行任务 a 和待执行任务 c 分配对应的处理线程,并预估待执行任务 a 完成后 CPU 中处理线程的负载情况,再据此为待执行任务 b 分配处理线程。

- 20 正如上文所介绍的,游戏引擎中还包括与游戏中的虚拟元素对应的实体,且实体包括组件数据,组件数据用于描述该实体对应的虚拟元素具备的功能属性实体包括的组件数据均配置有该实体的标识。在该种架构的游戏引擎中,特定任务处理系统在生成待执行任务时,还需要根据该游戏引擎中各实体各自包括的组件数据,确定与其负责处理的待执行任务相关的实体作为任务实体,并将任务实体的标识传输给任务调度系统。相应地,任务调度系统针对来自该特定任务处理系统的待执行任务分配完处理线程后,还需要将任务实体的标识传输给所分配的处理线程,以使该处理线程基于该任务实体的标识获取任务实体包括的组件数据来执行待执行任务。其中,实体包括的组件数据可以是一种或多种,本申请实施例对此不做限定。

- 30 示例性的,假设特定任务处理系统为用于实现移动功能的任务处理系统,该特定任务处理系统生成待执行任务时,可以遍历游戏引擎中包括的各个实体,以从中查找出包括与移动功能相关的组件数据的实体作为任务实体,并将该任务实体的标识通过自身对应的 API 传输给任务调度系统。任务调度系统针对该特定任务处理系统传输过来的待执行任务分配好对应的处理线程后,可以将该特定任务处理系统传输过来的任务实体的标识提供给所分配的处理线程;如此,该处理线程可以在执行该特定任务处理系统生成的待执行任务时,根据该任务实体的标识确定执行待执行任务时依据的组件数据,并基于该任务实体包括的与移动功能相关的组件数据,实现该任务实体对应的虚拟元素的移动功能,如确定该任务实体对应的虚拟元素移动后所处的位置。

需要说明的是，特定任务处理系统确定任务实体时，会遍历游戏引擎中包括的所有实体，游戏引擎中与该特定任务处理系统负责处理的待执行任务相关的所有实体，均会被视为任务实体。

5 应理解，在实际应用中，特定任务处理系统可以在向任务调度系统传输待执行任务的同时传输任务实体的标识，也可以在完成待执行任务的传输后、再传输任务实体的标识，还可以先传输任务实体的标识、后传输待执行任务，本申请在此不对待执行任务与任务实体的标识的传输顺序做任何限定。

10 如此，在 ECS 架构的游戏引擎中，通过上述方式确定待执行任务所需处理的组件数据，并在基于所确定的组件数据执行待执行任务，保证了任务执行的完整性，实现了对于游戏引擎中组件数据的灵活调用、以及对于实体对应的虚拟元素的灵活控制。

15 在一种可能的实现方式中，在游戏引擎中注册任务处理系统时，还可以针对任务处理系统配置其对应的优先级，该优先级通常是根据任务处理系统负责处理的任务的重要程度、和/或任务处理系统负责处理的任务所需占用的处理资源确定的。在该种情况下，任务调度系统针对来自特定任务处理系统的待执行任务分配处理线程时，可以基于 CPU 中处理线程的负载情况、特定任务处理系统对应的优先级、以及特定任务处理系统在待执行阶段中的运行关联关系，为待执行任务分配 CPU 中的处理线程。

20 可以理解的是，在游戏引擎中注册任务处理系统时，可以根据该任务处理系统负责处理的任务的重要程度、和/或该任务处理系统负责处理的任务所需占用的处理资源，来为该任务处理系统配置对应的优先级。应理解，任务处理系统负责处理的任务越重要、所需占用的处理资源越多，任务处理系统对应的优先级应当越高。

25 示例性的，在游戏引擎中，负责处理渲染任务的任务处理系统执行渲染任务时所需耗费的资源通常比较多，并且渲染任务对于游戏的整体实现来说比较重要，因此，可以针对负责处理渲染任务的任务处理系统配置较高的优先级；负责处理粒子效果的任务处理系统、以及负责处理音效的任务处理系统执行其负责处理的任务时所需耗费的资源通常比较少，并且粒子效果和音效对于游戏的整体实现来说重要程度较低，因此，可以针对负责处理粒子效果的任务处理系统、以及负责处理音效的任务处理系统配置较低的优先级。应理解，上述任务处理系统的优先级的配置方式仅为示例，在实际应用中，可以根据游戏引擎的实际运行情况，为其中的任务处理系统配置对应的优先级，本申请在此不对优先级的配置方式做任何限定。

30 在游戏引擎中各任务处理系统均配置有优先级的情况下，任务调度系统针对来自各特定任务处理系统的各待执行任务分配处理线程时，还需要综合考虑各特定任务处理系统各自对应的优先级。例如，优先针对来自高优先级的特定任务处理系统的待执行任务，分配性能较优、和/或资源占用率较低的处理线程。

35 作为一种示例，当 CPU 为多核异构 CPU 时，针对优先级高于第一预设级别的特定任务处理系统，任务调度系统可以为来自该特定任务处理系统的待执行任务，分配 CPU 中由最优内核支持运行的处理线程，该最优内核是该 CPU 中处理性能最优的内核。

目前,很多终端设备(例如移动终端)中的CPU由规模不同、频率不同的多个内核组成,为了使本申请实施例提供的任务调度方法能够较好地适配于此类移动终端,任务调度系统可以在针对来自任务处理系统的任务分配处理线程时,将比较重要的任务锁定在CPU中处理性能较优的内核上执行,而对于重要度较低的任务可以使其在CPU中处理性能一般的内核上执行。

基于此,可以预先设置第一预设级别,将优先级高于第一预设级别的任务处理系统负责处理的任务视为重要度较高的任务,将优先级不高于第一预设级别的任务处理系统负责处理的任务视为重要度较低的任务。当接收到优先级高于第一预设级别的任务处理系统传输过来的任务时,任务调度系统可以为该任务分配由最优内核(即CPU中性能最优的内核)支持运行的处理线程。例如,接收到负责处理渲染任务的任务处理系统传输过来的渲染任务时,任务调度系统可以将该渲染任务锁定在由最优内核支持运行的处理线程上执行。反之,当接收到优先级不高于第一预设级别的任务处理系统传输过来的任务时,任务调度系统可以为该任务分配由除最优内核外的其它内核支持运行的处理线程;例如,接收到负责处理粒子效果的任务处理系统传输过来的粒子效果处理任务时,任务调度系统可以将该粒子效果处理任务放置到由性能一般的内核支持运行的处理线程上执行。

应理解,在CPU中包括的内核的处理性能更多样的情况下,本申请实施例还可以适应性地针对游戏引擎中的各任务处理系统配置更多样的优先级,并且设置更多对应于不同处理性能的内核的级别区间;根据任务处理系统所对应的优先级所处的级别区间,为来自该任务处理系统的任务配置对应的处理内核。

如此,通过上述方式,实现适配于多核异构CPU的任务调度方案,保证将较为重要的任务放置在CPU中处理性能较优的内核上执行,从而保证游戏能够更稳定的运行。

作为一种示例,当CPU为多核同构CPU时,任务调度系统可以根据CPU中各处理线程的负载情况,确定该CPU中占用率满足预设条件的处理线程,作为优选处理线程。进而针对优先级高于第二预设级别的特定任务处理系统,任务调度系统可以为来自该特定任务处理系统的待执行任务,分配此类优选处理线程。

目前,很多终端设备(例如个人电脑端)中的CPU由规模相同、频率相同的多个内核组成,当本申请实施例提供的游戏引擎运行在此类电脑端时,任务调度系统可以基于CPU负载均衡的原则,为任务处理系统生成的任务分配CPU中的处理线程;对于来自所对应的优先级较高的任务处理系统的任务,任务调度系统可以为其优先分配CPU中占用率较低的处理线程。

基于此,可以预先设置第二预设级别,将优先级高于第二预设级别的任务处理系统负责处理的任务视为重要度较高的任务,将优先级不高于第二预设级别的任务处理系统负责处理的任务视为重要度较低的任务。

任务调度系统针对来自各特定任务处理系统的各待执行任务分配处理线程时,可以先根据支持游戏运行的CPU中各个处理线程的负载情况,确定该CPU中占用率满足预设条件的处理线程,作为优选处理线程。此处可以将占用率低于预设阈值作为预设条件,也可以将对CPU中各个处理线程的占用率进行升序排序,确定其中排序靠前的若干个处理线程

满足预设条件，本申请在此不对该预设条件做任何限定。进而，针对来自优先级高于第二预设级别的特定任务处理系统的待执行任务，分配所确定的优选处理线程。

如此，通过上述方式，实现适配于多核同构 CPU 的任务调度方案，保证将较为重要的任务放置在 CPU 中处理资源较丰富的处理线程上执行，从而保证游戏能够更稳定的运行。

5 在实际应用中，任务处理系统负责处理的任务可能由若干具有关联关系的子任务组成，即特定任务处理系统向任务调度系统传输的待执行任务中可能包括多个待执行子任务，以及这多个待执行子任务各自对应的执行关联关系，此处的执行关联关系用于表征其对应的待执行子任务与待执行任务中其它待执行子任务之间的执行依赖关系。此时，任务调度系统可以通过以下方式针对待执行任务以及其中的待执行子任务配置处理线程：

10 针对来自各特定任务处理系统的待执行任务，基于 CPU 中各处理线程的负载情况和特定任务处理系统在待执行阶段中的运行关联关系，为待执行任务分配对应的处理线程集合。进而，针对每个待执行任务，基于该待执行任务对应的处理线程集合中各处理线程的负载情况，以及该待执行任务中各待执行子任务各自对应的执行关联关系，为待执行任务中各待执行子任务分配处理线程集合中的处理线程。

15 示例性的，特定任务处理系统生成的待执行任务实质上可以由多个具有关联关系的待执行子任务组成，特定任务处理系统将其生成的待执行任务传输给任务调度系统，实质上即是将其生成的待执行任务中的多个待执行子任务以及这多个待执行子任务各自对应的执行关联关系传输给任务调度系统。此处待执行子任务对应的执行关联关系能够表征该待执行子任务与待执行任务中其它的待执行子任务之间的执行依赖关系，例如，假设待执行子任务 1 的执行需要依赖待执行子任务 2 的处理结果，则可以基于此生成待执行子任务 1 对应的执行关联关系。

20 任务调度系统分配处理线程时，可以先在游戏引擎层级上针对所接收的各个待执行任务分配对应的处理线程集合，所分配的处理线程集合中可以包括一个处理线程，也可以包括多个处理线程。在游戏引擎层级上针对待执行任务分配处理线程集合的实现方式，实质上与上文介绍的针对待执行任务分配处理线程的实现方式相同，详细可参见上文的相关介绍内容，此处不再赘述。

25 任务调度系统分配得到待执行任务对应的处理线程集合后，可以进一步在任务层级上针对待执行任务中的各个待执行子任务分配对应的处理线程。在一种可能的实现方式中，任务调度系统可以根据待执行任务中各个待执行子任务各自对应的执行关联关系，结合待执行任务对应的处理线程集合中各处理线程当前的负载情况，为待执行任务中的各个待执行子任务分配对应的处理线程。例如，假设待执行任务包括待执行子任务 1、待执行子任务 2 和待执行子任务 3，并且待执行子任务 2 的执行需要依赖待执行子任务 1 的处理结果，而待执行子任务 3 可以独立执行，不依赖其它待执行子任务的执行结果，则任务调度系统可以先为待执行子任务 1 和待执行子任务 3 分配对应的处理线程，并预估待执行子任务 1 30 完成后处理线程集合中处理线程的负载情况，再据此为待执行子任务 2 分配处理线程。

35 如此，通过上述方式，在游戏引擎层级和任务层级这两个层级上实现双层级的处理资源分配，既保证了针对每个所需处理的任务合理分配处理资源，又保证了针对任务中的每

个子任务合理分配处理资源,更细化地实现了对于处理资源的合理分配,进一步提高了 CPU 的并行度。

在一种可能的实现方式中,待执行任务中各个待执行子任务也可以有其对应的优先级,即特定任务处理系统在生成待执行任务时,可以根据该待执行任务中待执行子任务的重要程度、和/或所需占用的处理资源,配置该待执行子任务对应的优先级。在该种情况下,任务调度系统针对待执行任务中的待执行子任务分配处理线程时,可以根据该待执行任务对应的处理线程集合中各处理线程的负载情况,确定该处理线程集合中占用率最低的处理线程。进而针对优先级高于第三预设级别的待执行子任务,任务调度系统可以为该待执行子任务分配该占用率最低的处理线程。

特定任务处理系统生成待执行任务时,可以根据该待执行任务中各待执行子任务的重要程度、和/或该待执行任务中各待执行子任务所需占用的处理资源,为该待执行任务中的各待执行子任务配置对应的优先级。应理解,待执行任务中重要程度越高、所需占用的处理资源越多的待执行子任务,所对应的优先级应当越高。在待执行任务中各待执行子任务配置有对应的优先级的情况下,任务调度系统针对待执行任务中的各待执行子任务分配处理线程时,还需要综合考虑各待执行子任务各自对应的优先级。例如,针对所对应的优先级较高的待执行子任务,分配处理线程集合中资源占用率较低的处理线程。

示例性的,可以预先设置第三预设级别,将优先级高于该第三预设级别的待执行子任务视为重要度较高的子任务,将优先级不高于该第三预设级别的待执行子任务视为重要度较低子任务。任务调度系统针对待执行任务中各待执行子任务分配处理线程时,可以先根据该待执行任务对应的处理线程集合中各处理线程的负载情况,确定其中占用率最低的处理线程。进而,针对优先级高于第三预设级别的待执行子任务,分配该占用率最低的处理线程。

如此,通过上述方式,实现对于任务层级的处理资源分配方式的进一步优化,使得比较重要的子任务可以在占用率较低的处理线程上执行,从而保证待执行任务能够较好地执行。

上述任务调度方法提出了游戏引擎层级的处理资源调度方式,即在游戏运行过程所涉及的每个任务处理阶段内,综合考虑游戏引擎在该任务处理阶段内产生的各个任务之间的运行关联关系,进而,据此合理地针对各个任务分配处理资源。该任务调度方法应用于基于 ECS 架构搭建的游戏引擎,该种游戏引擎将游戏中的各种处理逻辑解耦开来,利用若干独立的任务处理系统分别执行不同的任务;本申请在该种游戏引擎的基础上,创新性地针对其中每个任务处理系统配置了其适用的任务处理阶段、以及其在所适用的任务处理阶段中的运行关联关系,并且还在该种游戏引擎中增设了用于综合调度处理资源的任务调度系统。当游戏引擎工作在某个任务处理阶段例如待执行阶段时,任务调度系统可以根据适用于该待执行阶段的特定任务处理系统的运行关联关系,将 CPU 中的处理线程合理地分配给适用于该待执行阶段的特定任务处理系统生成的待执行任务,实现从游戏引擎层面出发的处理资源配置。如此,综合考虑一个任务处理系统与游戏引擎中其它任务处理系统之间的

运行依赖关系来分配处理资源，可以实现对 CPU 并行情况的整体控制，从而有助于提高游戏帧率，降低 CPU 的发热损耗。

为了便于进一步理解本申请实施例提供的任务调度方法，下面对该任务调度方法进行整体示例性介绍。

- 5 考虑到游戏运行时通常需要执行若干帧内的任务，因此本申请将游戏中对于一个游戏帧的处理过程划分为 PreUpdate、Update 和 PostUpdate 三个阶段。对于游戏引擎中的每个任务处理系统，可以根据其负责处理的任务的执行时机，确定其具体适用于这三个阶段中的哪个阶段或者哪些阶段，并据此配置其对应的任务处理阶段。例如，对于用于执行脚本逻辑的任务处理系统，其需要在游戏帧对应的 PreUpdate 阶段运行一些脚本逻辑（如更新物体位置、播放动画等），因此可以配置其对应的任务处理阶段为 PreUpdate 阶段；又例如，对于用于执行渲染任务的任务处理系统，其需要在游戏帧对应的 Update 阶段执行生成渲染命令、渲染裁剪等计算任务，因此可以配置其对应的任务处理阶段为 Update 阶段。

- 10 在上述 PreUpdate、Update 和 PostUpdate 中的每个阶段中，适用于该阶段的任务处理系统均可以通过其对应的 API，将其生成的所需执行的任务传输给任务调度系统，以使该任务调度系统为其生成的任务分配 CPU 中的处理线程。此处，任务处理系统所生成的任务中可以包括若干子任务，这些子任务彼此之间可以存在依赖关系，也可以任意执行，任务处理系统可以在创建任务时指定该任务中包括的子任务间的执行关联关系，例如，指定其中前置的子任务、后置的子任务等，进而将这些子任务间的执行关联关系传输给任务调度系统。

- 20 在本申请实施例中，在每个任务处理系统的注册阶段，还会配置其在所适用的任务处理阶段中与其它任务处理系统之间的运行关联关系，即配置其是否需要依赖其它任务处理系统的处理结果执行其负责处理的任務。例如，用于执行渲染任务的任务处理系统，会在其适用的任务处理阶段中依赖用于执行物理计算任务的任务处理系统的处理结果、以及用于执行动画任务的任务处理系统的处理结果，因此可以基于此配置用于执行渲染任务的任务处理系统的运行关联关系。此外，在每个任务处理系统的注册阶段，还可以根据该任务处理系统负责处理的任务的重要程度、和/或该任务处理系统负责处理的任務所需占用的处理资源，为该任务处理系统配置对应的优先级。

- 30 如此，在指定了游戏引擎中任务处理系统所适用的任务处理阶段、在所适用的任务处理阶段中的运行关联关系以及优先级后，任务调度系统可以基于 CPU 中各处理线程的负载情况、任务处理系统的优先级和运行关联关系，合理地安排任务到相应的处理线程上执行。本申请实施例中的任务调度系统在游戏运行时，针对游戏中的每个游戏帧，可以循环执行任务创建（Build Execution Graph）阶段和任务执行（Execute Task Graph）阶段，将所有游戏任务并行执行，如图 5 所示。

- 35 此外，考虑到移动终端中的 CPU 与个人电脑（Personal Computer, PC）端中的 CPU 架构不同，移动终端中的 CPU 通常由若干规模不同、频率不同的大小内核组成，基于此，任务调度系统并不能简单地将任务按照负载均衡的原则分配到各个处理线程上。在该种情况下，本申请实施例可以根据任务处理系统对应的优先级，将优先级较高的任务处理系统

生成的任务锁定在 CPU 中性能较优的内核上执行，例如，将渲染任务锁定在 CPU 中的大核上执行，而对于粒子、音效等任务可以放在性能较慢的小核上执行，如此更好地提高游戏在移动终端上的性能。

通过本申请实施例提供的任务调度方法，可以将所需执行的各个任务合理地分配到各个处理线程上，保证 CPU 具有较高的整体利用率。图 6 所示为游戏引擎基于本申请实施例提供的任务调度方法运行时对于 CPU 处理资源的占用情况，其中六条横向的线条分别代表 CPU 中六个不同的处理线程，线条上分布有纵向纹路的区域表示在该区域对应的时间段内该线条对应的处理线程得到了使用，线条上没有分布纵向纹路的区域表示在该区域对应的时间段内该线条对应的处理线程没有被使用。如图 6 所示，执行任务 1、任务 2、任务 3 和任务 4（此处的任务 1、任务 2、任务 3 和任务 4 可以是独立的任务，也可以是包括多个任务的集合）时，图 6 中每个线条上没有分布纵向纹路的区域非常少，说明了 CPU 中这六个处理线程都得到了充分地利用，CPU 的整体利用率不低于 70%。

针对上文描述的任务调度方法，本申请还提供了对应的游戏引擎，以使上述任务调度方法在实际中得以应用及实现。

参见图 7，图 7 是本申请实施例提供的游戏引擎的结构示意图。如图 7 所示，该游戏引擎 700 包括任务调度系统 701 和多个任务处理系统 702；所述多个任务处理系统 702 各自负责处理的任务不同，每个所述任务处理系统 702 配置有对应的任务处理阶段、以及其在所对应的任务处理阶段中的运行关联关系，所述运行关联关系用于表征所述任务处理系统 702 与所述游戏引擎中其它任务处理系统 702 之间的运行依赖关系；

所述任务处理系统 702，用于当所述游戏引擎工作在其对应的任务处理阶段时，生成其负责处理的任务，并将所述任务传输给所述任务调度系统 701；

所述任务调度系统 701，用于当所述游戏引擎工作在待执行阶段时，针对来自各特定任务处理系统的待执行任务，基于 CPU 中各处理线程的负载情况和所述特定任务处理系统在所述待执行阶段中的运行关联关系，为所述待执行任务分配所述 CPU 中的处理线程；所述特定任务处理系统是所述游戏引擎中所配置的任务处理阶段为所述待执行阶段的任务处理系统，所述待执行任务是所述特定任务处理系统生成的其负责处理的任务。

在一种可能的实现方式中，在图 7 所示的游戏引擎的基础上，所述游戏引擎还包括多个实体，所述实体包括组件数据，所述实体包括的组件数据均配置有所述实体的标识；

所述任务处理系统 702，还用于当所述游戏引擎工作在其对应的任务处理阶段时，根据所述游戏引擎中各实体包括的组件数据，确定与其负责处理的任务相关的实体，并将所述实体的标识传输给所述任务调度系统 701；

所述任务调度系统 701，还用于针对来自所述特定任务处理系统的所述待执行任务分配完处理线程后，将任务实体的标识传输给所述处理线程，以使所述处理线程基于所述任务实体的标识获取所述任务实体包括的组件数据来执行所述待执行任务；所述任务实体是与所述特定任务处理系统负责处理的所述待执行任务相关的实体。

在一种可能的实现方式中，在图 7 所示的游戏引擎的基础上，每个所述任务处理系统 702 还配置有其对应的优先级，；

所述任务调度系统 701, 具体用于针对来自各所述特定任务处理系统的所述待执行任务, 基于所述 CPU 中各处理线程的负载情况、所述特定任务处理系统对应的优先级、以及所述特定任务处理系统在所述待执行阶段中的运行关联关系, 为所述待执行任务分配所述 CPU 中的处理线程。

- 5 在一种可能的实现方式中, 当所述 CPU 为多核异构 CPU 时, 所述任务调度系统 701 具体用于:

 针对优先级高于第一预设级别的所述特定任务处理系统, 为来自所述特定任务处理系统的所述待执行任务, 分配所述 CPU 中由最优内核支持运行的处理线程; 所述最优内核为所述 CPU 中处理性能最优的内核。

- 10 在一种可能的实现方式中, 当所述 CPU 为多核同构 CPU 时, 所述任务调度系统 701 具体用于:

 根据所述 CPU 中各处理线程的负载情况, 确定所述 CPU 中占用率满足预设条件的处理线程, 作为优选处理线程; 针对优先级高于第二预设级别的所述特定任务处理系统, 为来自所述特定任务处理系统的所述待执行任务, 分配所述优选处理线程。

- 15 在一种可能的实现方式中, 所述优先级是根据所述任务处理系统负责处理的任务的重要程度、以及所述任务处理系统负责处理的任务所需占用的处理资源中的至少一种因素确定的。

- 在一种可能的实现方式中, 在图 7 所示的游戏引擎的基础上, 所述待执行任务包括多个待执行子任务以及所述多个待执行子任务各自对应的执行关联关系, 所述执行关联关系用于表征其对应的待执行子任务与所述待执行任务中其它的待执行子任务之间的执行依赖关系; 则所述任务调度系统 701 具体用于:

 针对来自各所述特定任务处理系统的所述待执行任务, 所述任务调度系统基于所述 CPU 中各处理线程的负载情况和所述特定任务处理系统在所述待执行阶段中的运行关联关系, 为所述待执行任务分配对应的处理线程集合;

- 25 针对每个所述待执行任务, 所述任务调度系统基于所述待执行任务对应的处理线程集合中各处理线程的负载情况, 以及所述待执行任务中各待执行子任务各自对应的执行关联关系, 为所述待执行任务中各待执行子任务分配所述处理线程集合中的处理线程。

 在一种可能的实现方式中, 所述待执行任务还包括所述多个待执行子任务各自对应的优先级; 则所述任务调度系统 701 具体用于:

- 30 根据所述处理线程集合中各处理线程的负载情况, 确定所述处理线程集合中占用率最低的处理线程; 针对优先级高于第三预设级别的所述待执行子任务, 为所述待执行子任务分配所述占用率最低的处理线程。

- 在一种可能的实现方式中, 在图 7 所示的游戏引擎的基础上, 所述任务处理阶段是根据对于游戏帧的处理过程划分的, 所述任务处理阶段包括游戏帧对应的预更新阶段、更新阶段和更新完成阶段; 所述预更新阶段用于执行更新所述游戏帧所需的准备任务, 所述更新阶段用于执行对于所述游戏帧的更新任务, 所述更新完成阶段用于执行更新完所述游戏帧后所需的收尾任务。

在一种可能的实现方式中，在图 7 所示的游戏引擎的基础上，所述任务处理系统 702 具体用于：

生成其负责处理的任務，并通过 API 将所述任务传输给所述任务调度系统 701。

上述任务调度装置提出了游戏引擎层级的处理资源调度方式，即在游戏运行过程所涉
5 及的每个任务处理阶段内，综合考虑游戏引擎在该任务处理阶段内产生的各个任务之间的
运行关联关系，进而，据此合理地针对各个任务分配处理资源。该任务调度方法应用于基
于 ECS 架构搭建的游戏引擎，该种游戏引擎将游戏中的各种处理逻辑解耦开来，利用若干
独立的任务处理系统分别执行不同的任务；本申请在该种游戏引擎的基础上，创新性地针
10 对其中每个任务处理系统配置了其适用的任务处理阶段、以及其在所适用的任务处理阶段
中的运行关联关系，并且还在该种游戏引擎中增设了用于综合调度处理资源的任务调度系
统。当游戏引擎工作在某个任务处理阶段例如待执行阶段时，任务调度系统可以根据适用
于该待执行阶段的特定任务处理系统的运行关联关系，将 CPU 中的处理线程合理地分配给
15 适用于该待执行阶段的特定任务处理系统生成的待执行任务，实现从游戏引擎层面出发的
处理资源配置。如此，综合考虑一个任务处理系统与游戏引擎中其它任务处理系统之间的
运行依赖关系来分配处理资源，可以实现对 CPU 并行情况的整体控制，从而有助于提高游
戏帧率，降低 CPU 的发热损耗。

本申请实施例还提供了一种用于实现上述任务调度方法的设备，该设备具体可以是终端设备，下面将从硬件实体化的角度对本申请实施例提供的终端设备进行介绍。

参见图 8，图 8 是本申请实施例提供的终端设备的结构示意图。如图 8 所示，为了便
20 于说明，仅示出了与本申请实施例相关的部分，具体技术细节未揭示的，请参照本申请实
施例方法部分。该终端设备可以为包括手机、平板电脑、个人数字助理、销售终端（Point of
Sales, POS）、车载电脑等任意终端设备，以终端设备为计算机为例：

图 8 示出的是与本申请实施例提供的终端设备相关的计算机的部分结构的框图。参考
图 8，计算机包括：射频（Radio Frequency, RF）电路 810、存储器 820、输入单元 830（其
25 中包括触控面板 831 和其他输入设备 832）、显示单元 840（其中包括显示面板 841）、传
感器 850、音频电路 860（其可以连接扬声器 861 和传声器 862）、无线保真（wireless fidelity,
WiFi）模块 870、处理器 880、以及电源 890 等部件。本领域技术人员可以理解，图 8 中示
出的计算机结构并不构成对计算机的限定，可以包括比图示更多或更少的部件，或者组合
某些部件，或者不同的部件布置。

存储器 820 可用于存储软件程序以及模块，处理器 880 通过运行存储在存储器 820 的
30 软件程序以及模块，从而执行计算机的各种功能应用以及数据处理。存储器 820 可主要包
括存储程序区和存储数据区，其中，存储程序区可存储操作系统、至少一个功能所需的应
用程序（比如声音播放功能、图像播放功能等）等；存储数据区可存储根据计算机的使用
所创建的数据（比如音频数据、电话本等）等。此外，存储器 820 可以包括高速随机存取
35 存储器，还可以包括非易失性存储器，例如至少一个磁盘存储器件、闪存器件、或其他易
失性固态存储器件。

处理器 880 是计算机的控制中心，利用各种接口和线路连接整个计算机的各个部分，通过运行或执行存储在存储器 820 内的软件程序和/或模块，以及调用存储在存储器 820 内的数据，执行计算机的各种功能和处理数据，从而对计算机进行整体监控。可选的，处理器 880 可包括一个或多个处理单元；优选的，处理器 880 可集成应用处理器和调制解调处理器，其中，应用处理器主要处理操作系统、用户界面和应用程序等，调制解调处理器主要处理无线通信。可以理解的是，上述调制解调处理器也可以不集成到处理器 880 中。

在本申请实施例中，该终端设备所包括的处理器 880 还具有以下功能：

支持游戏引擎的运行，所述游戏引擎包括任务调度系统和多个任务处理系统；所述多个任务处理系统各自负责处理的任务不同，每个所述任务处理系统配置有对应的任务处理阶段、以及其在所对应的任务处理阶段中的运行关联关系，所述运行关联关系用于表征所述任务处理系统与所述游戏引擎中其它任务处理系统之间的运行依赖关系；

当所述游戏引擎工作在待执行阶段时，各特定任务处理系统分别生成各自负责处理的待执行任务，并将所述待执行任务传输给所述任务调度系统；所述特定任务处理系统是所述游戏引擎中所配置的任务处理阶段为所述待执行阶段的任务处理系统；

针对来自各所述特定任务处理系统的所述待执行任务，所述任务调度系统基于中央处理器 CPU 中各处理线程的负载情况和所述特定任务处理系统在所述待执行阶段中的运行关联关系，为所述待执行任务分配所述 CPU 中的处理线程。

在一种可能的实现方式中，所述处理器 880 还用于执行本申请实施例提供的任务调度方法的任意一种实现方式的步骤。

本申请实施例还提供一种计算机可读存储介质，用于存储计算机程序，该计算机程序用于执行前述各个实施例所述的一种任务调度方法中的任意一种实施方式。

本申请实施例还提供了一种计算机程序产品或计算机程序，该计算机程序产品或计算机程序包括计算机指令，该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令，处理器执行该计算机指令，使得该计算机设备执行前述各个实施例所述的一种任务调度方法中的任意一种实施方式。

所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，上述描述的系统，装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

在本申请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。

所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

另外，在本申请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，也可以采用软件功能单元的形式实现。

5 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用，可以存储在一个计算机可读取存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来，该计算机软件产品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设备（可以是个人计算机，服务器，或者网络设备等）执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U 盘、移动硬盘、只读存储器（英文全称：Read-Only Memory，英文缩写：ROM）、随机存取存储器（英文全称：Random Access Memory，英文缩写：RAM）、磁碟或者光盘等各种可以存储计算机程序的介质。

15 应当理解，在本申请中，“至少一个（项）”是指一个或者多个，“多个”是指两个或两个以上。“和/或”，用于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：只存在 A，只存在 B 以及同时存在 A 和 B 三种情况，其中 A，B 可以是单数或者复数。字符“/”一般表示前后关联对象是一种“或”的关系。“以下至少一项（个）”或其类似表达，是指这些项中的任意组合，包括单项（个）或复数项（个）的任意组合。例如，a，b 或 c 中的至少一项（个），可以表示：a，b，c，“a 和 b”，“a 和 c”，“b 和 c”，或“a 和 b 和 c”，其中 a，b，c 可以是单个，也可以是多个。

20 以上所述，以上实施例仅用以说明本申请的技术方案，而非对其限制；尽管参照前述实施例对本申请进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围。

权 利 要 求

1、一种任务调度方法，所述方法由终端设备执行，所述终端设备用于支持游戏引擎运行，所述游戏引擎包括任务调度系统和多个任务处理系统；所述多个任务处理系统各自负责处理的任务不同，每个所述任务处理系统配置有对应的任务处理阶段、以及其在所对应的任务处理阶段中的运行关联关系，所述运行关联关系用于表征所述任务处理系统与所述游戏引擎中其它任务处理系统之间的运行依赖关系；所述方法包括：

当所述游戏引擎工作在待执行阶段时，各特定任务处理系统分别生成各自负责处理的待执行任务，并将所述待执行任务传输给所述任务调度系统；所述特定任务处理系统是所述游戏引擎中所配置的任务处理阶段为所述待执行阶段的任务处理系统；

针对来自各所述特定任务处理系统的所述待执行任务，所述任务调度系统基于中央处理器 CPU 中各处理线程的负载情况和所述特定任务处理系统在所述待执行阶段中的运行关联关系，为所述待执行任务分配所述 CPU 中的处理线程。

2、根据权利要求 1 所述的方法，所述游戏引擎还包括多个实体，所述实体包括组件数据，所述实体包括的组件数据均配置有所述实体的标识；所述方法还包括：

所述特定任务处理系统根据所述游戏引擎中各实体包括的组件数据，确定与其负责处理的所述待执行任务相关的实体作为任务实体，并将所述任务实体的标识传输给所述任务调度系统；

所述任务调度系统针对来自所述特定任务处理系统的所述待执行任务分配完处理线程后，将所述任务实体的标识传输给所述处理线程，以使所述处理线程基于所述任务实体的标识获取所述任务实体包括的组件数据来执行所述待执行任务。

3、根据权利要求 1 或 2 所述的方法，每个所述任务处理系统还配置有优先级；

针对来自各所述特定任务处理系统的所述待执行任务，所述任务调度系统基于中央处理器 CPU 中各处理线程的负载情况和所述特定任务处理系统在所述待执行阶段中的运行关联关系，为所述待执行任务分配所述 CPU 中的处理线程，包括：

针对来自各所述特定任务处理系统的所述待执行任务，所述任务调度系统基于所述 CPU 中各处理线程的负载情况、所述特定任务处理系统对应的优先级、以及所述特定任务处理系统在所述待执行阶段中的运行关联关系，为所述待执行任务分配所述 CPU 中的处理线程。

4、根据权利要求 3 所述的方法，当所述 CPU 为多核异构 CPU 时，所述针对来自各所述特定任务处理系统的所述待执行任务，所述任务调度系统基于所述 CPU 中各处理线程的负载情况、所述特定任务处理系统对应的优先级、以及所述特定任务处理系统在所述待执行阶段中的运行关联关系，为所述待执行任务分配所述 CPU 中的处理线程，包括：

针对优先级高于第一预设级别的所述特定任务处理系统，所述任务调度系统为来自所述特定任务处理系统的所述待执行任务，分配所述 CPU 中由最优内核支持运行的处理线程；所述最优内核为所述 CPU 中处理性能最优的内核。

5、根据权利要求 3 所述的方法，当所述 CPU 为多核同构 CPU 时，所述针对来自各所述特定任务处理系统的所述待执行任务，所述任务调度系统基于所述 CPU 中各处理线程的

负载情况、所述特定任务处理系统对应的优先级、以及所述特定任务处理系统在所述待执行阶段中的运行关联关系，为所述待执行任务分配所述 CPU 中的处理线程，包括：

所述任务调度系统根据所述 CPU 中各处理线程的负载情况，确定所述 CPU 中占用率满足预设条件的处理线程，作为优选处理线程；

- 5 针对优先级高于第二预设级别的所述特定任务处理系统，所述任务调度系统为来自所述特定任务处理系统的所述待执行任务，分配所述优选处理线程。

6、根据权利要求 3 所述的方法，所述优先级是根据所述任务处理系统负责处理的任务的重要程度、以及所述任务处理系统负责处理的任务所需占用的处理资源中的至少一种因素确定的。

- 10 7、根据权利要求 1 或 2 所述的方法，所述待执行任务包括多个待执行子任务以及所述多个待执行子任务各自对应的执行关联关系，所述执行关联关系用于表征其对应的待执行子任务与所述待执行任务中其它的待执行子任务之间的执行依赖关系；

所述针对来自各所述特定任务处理系统的所述待执行任务，所述任务调度系统基于中央处理器 CPU 中各处理线程的负载情况和所述特定任务处理系统在所述待执行阶段中的运行关联关系，为所述待执行任务分配所述 CPU 中的处理线程，包括：

- 15 针对来自各所述特定任务处理系统的所述待执行任务，所述任务调度系统基于所述 CPU 中各处理线程的负载情况和所述特定任务处理系统在所述待执行阶段中的运行关联关系，为所述待执行任务分配对应的处理线程集合；

- 20 针对每个所述待执行任务，所述任务调度系统基于所述待执行任务对应的处理线程集合中各处理线程的负载情况，以及所述待执行任务中各待执行子任务各自对应的执行关联关系，为所述待执行任务中各待执行子任务分配所述处理线程集合中的处理线程。

- 25 8、根据权利要求 7 所述的方法，所述待执行任务还包括所述多个待执行子任务各自对应的优先级；所述任务调度系统基于所述待执行任务对应的处理线程集合中各处理线程的负载情况，以及所述待执行任务中各待执行子任务各自对应的执行关联关系，为所述待执行任务中各待执行子任务分配所述处理线程集合中的处理线程，包括：

所述任务调度系统根据所述处理线程集合中各处理线程的负载情况，确定所述处理线程集合中占用率最低的处理线程；

针对优先级高于第三预设级别的所述待执行子任务，所述任务调度系统为所述待执行子任务分配所述占用率最低的处理线程。

- 30 9、根据权利要求 1 或 2 所述的方法，所述任务处理阶段是根据对游戏帧的处理过程划分的，所述任务处理阶段包括游戏帧对应的预更新阶段、更新阶段和更新完成阶段；所述预更新阶段用于执行更新所述游戏帧所需的准备任务，所述更新阶段用于执行对于所述游戏帧的更新任务，所述更新完成阶段用于执行更新完所述游戏帧后所需的收尾任务。

- 35 10、根据权利要求 1 或 2 所述的方法，所述各特定任务处理系统分别生成各自负责处理的待执行任务，并将所述待执行任务传输给所述任务调度系统，包括：

各所述特定任务处理系统分别生成各自负责处理的所述待执行任务，并通过应用程序接口 API，将所述待执行任务传输给所述任务调度系统。

11、一种游戏引擎，所述游戏引擎运行在终端设备上，所述游戏引擎包括任务调度系统和多个任务处理系统；所述多个任务处理系统各自负责处理的任务不同，每个所述任务处理系统配置有对应的任务处理阶段、以及其在所对应的任务处理阶段中的运行关联关系，所述运行关联关系用于表征所述任务处理系统与所述游戏引擎中其它任务处理系统之间的运行依赖关系；

所述任务处理系统，用于当所述游戏引擎工作在其对应的任务处理阶段时，生成其负责处理的任务，并将所述任务传输给所述任务调度系统；

所述任务调度系统，用于当所述游戏引擎工作在待执行阶段时，针对来自各特定任务处理系统的待执行任务，基于 CPU 中各处理线程的负载情况和所述特定任务处理系统在所述待执行阶段中的运行关联关系，为所述待执行任务分配所述 CPU 中的处理线程；所述特定任务处理系统是所述游戏引擎中所配置的任务处理阶段为所述待执行阶段的任务处理系统，所述待执行任务是所述特定任务处理系统生成的其负责处理的任务。

12、根据权利要求 11 所述的游戏引擎，其特征在于，所述游戏引擎还包括多个实体，所述实体包括组件数据，所述实体包括的组件数据均配置有所述实体的标识；

所述任务处理系统，还用于当所述游戏引擎工作在其对应的任务处理阶段时，根据所述游戏引擎中各实体包括的组件数据，确定与其负责处理的任务相关的实体，并将所述实体的标识传输给所述任务调度系统；

所述任务调度系统，还用于针对来自所述特定任务处理系统的所述待执行任务分配完处理线程后，将任务实体的标识传输给所述处理线程，以使所述处理线程基于所述任务实体的标识获取所述任务实体包括的组件数据来执行所述待执行任务；所述任务实体是与所述特定任务处理系统负责处理的所述待执行任务相关的实体。

13、根据权利要求 11 或 12 所述的游戏引擎，其特征在于，每个所述任务处理系统还配置有其对应的优先级；

所述任务调度系统，具体用于针对来自各所述特定任务处理系统的所述待执行任务，基于所述 CPU 中各处理线程的负载情况、所述特定任务处理系统对应的优先级、以及所述特定任务处理系统在所述待执行阶段中的运行关联关系，为所述待执行任务分配所述 CPU 中的处理线程。

14、根据权利要求 13 所述的游戏引擎，其特征在于，当所述 CPU 为多核异构 CPU 时，所述任务调度系统具体用于：

针对优先级高于第一预设级别的所述特定任务处理系统，为来自所述特定任务处理系统的所述待执行任务，分配所述 CPU 中由最优内核支持运行的处理线程；所述最优内核为所述 CPU 中处理性能最优的内核。

15、一种设备，所述设备包括处理器及存储器；

所述存储器用于存储计算机程序；

所述处理器用于根据所述计算机程序执行权利要求 1 至 10 中任一项所述的任务调度方法。

—23—

16、一种计算机可读存储介质，所述计算机可读存储介质用于存储计算机程序，所述计算机程序用于执行权利要求 1 至 10 中任一项所述的任务调度方法。

17、一种计算机程序产品，当所述计算机程序产品被执行时，使得所述处理器实现如权利要求 1 至 10 中任一项所述的任务调度方法。