**说 明 书**

一种配置虚拟人物的换装资源的方法、装置和设备

技术领域

本公开的各实施例涉及虚拟人物技术领域，尤其涉及一种配置虚拟人物的换装资源的技术。

背景技术

现有的游戏换装系统开发较为复杂，需要考虑以下问题：

1.开发复杂性：实现一个完整的游戏换装系统需要处理不同模型之间的适配、动画过渡、碰撞体调整等问题。这需要额外的编程工作和调试，增加了开发的复杂性和时间成本。

2.动画和物理交互问题：游戏换装系统可能会引入动画和物理交互的问题。不同模型的尺寸、比例和关节结构可能不同，可能需要额外的调整和匹配动画，以确保换装后的角色能够正确地进行动作和交互。

3.美术资源制作：创建大量的服装和配件需要艺术家进行建模、纹理贴图和动画制作等工作。这对于游戏开发团队来说可能是一个庞大的任务，需要大量的时间和资源来完成。

4.用户体验一致性：游戏换装系统需要确保在不同的角色模型和服装组合下，用户体验的一致性和流畅性。这可能涉及到处理模型之间的过渡、遮挡、贴图对齐等问题，以确保换装的过程和结果都能够让玩家感到自然和无缝。

发明内容

本公开的各实施例的目的是提供

根据本公开的一个方面，提供了

根据一个实施例，

根据本公开的一个方面，还提供了

本公开的各实施例通过提供换装管线设计，即从换装美术资源到换装开发的流程管线，来制作Unity预制件，提高了整个流水线的效率。具体地，本公开的各实施例中，

1.设计良好的架构：采用良好的架构设计，例如组件化、模块化等，以简化换装系统的开发和维护。

2.工具支持：使用可视化工具或编辑器插件来简化换装系统的配置和调整，减少编程工作的复杂性。

3.文档和示例：提供清晰的文档和示例代码，以帮助开发者理解和使用换装系统，降低学习和实现的难度。

4.动画和物理交互问题：1）角色模型的一致性：确保不同服装和配件的角色模型具有相似的骨骼结构和关节层次，以便动画能够适应各种组合。2）动画匹配和混合：针对不同服装组合创建适当的动画过渡和混合，确保角色在换装时能够自然地过渡和过渡动作。

5.美术资源制作：1）流程优化：优化美术资源的制作流程，例如使用可重用的模型部件、贴图共享等，以减少重复工作和资源浪费。2）高效的工具和流程：使用适当的3D建模、纹理制作和动画工具，提高美术资源制作的效率和质量。

6.用户体验一致性：过渡效果和动画：为角色换装过程添加过渡动画和特效，使切换服装时的过渡更加平滑和自然。

附图说明

通过阅读参照以下附图所作的对非限制性实施例所作的详细描述，本公开的其它特征、目的和优点将会变得更明显：

图1示出根据本公开的一个实施例的方法流程图；

图3示出根据本公开的一个实施例的装置的示意图。

附图中相同或相似的附图标记代表相同或相似的部件。

具体实施方式

以下将结合附图来进一步描述本公开的具体实施例。

在更加详细地讨论示例性实施例之前应当提到的是，本公开的一些示例性实施例被描述为由方框图表述的装置和由流程图表述的过程或方法｡虽然流程图将本公开的各实施例的操作过程描述成顺序的处理，但是其中的许多操作可以被并行地、并发地或者同时实施。此外，各项操作的顺序可以被重新安排。本公开的各实施例的过程可在其操作执行完毕时被终止，但也可包括未在所述流程图中示出的额外步骤｡本公开的各实施例的过程可以对应于方法、功能、规程、子例程、子程序等｡

以下讨论的由流程图示出的方法和由方框图示出的装置，可以通过硬件、软件、固件、中间件、微代码、硬件描述语言、或其任意组合实现｡当以软件、固件、中间件或微代码实现时，执行必要任务的程序代码或代码段可被存储于机器或诸如存储介质的计算机可读介质｡（一个或多个）处理器可以执行所述必要任务｡

类似地，还将理解任何流程表、流程图、状态转换图，诸如此类，表示各种过程，其可以被充分地描述为存储于计算机可读介质内的程序代码并因此被计算机设备或处理器执行，无论这些计算机设备或处理器是否被明确示出。

本文中，术语“存储介质”可以表示一个或多个用于存储数据的设备，包括只读存储器（ROM），随机存取存储器（RAM），磁性RAM，内核存储器，磁盘存储介质，光存储介质，闪存设备和/或其他用于存储信息的机器可读介质｡术语“计算机可读介质”可包括但不限于，便携的或固定的存储设备，光存储设备，及各种其他能够存储和/或包含指令和/或数据的介质｡

代码段可表示规程、功能、子程序、程序、例程、子例程、模块、软件包、类，或指令、数据结构或程序描述的任一组合｡一个代码段可以通过传递和/或接收信息、数据、自变量、参数或存储内容，与另一个代码段或硬件电路相耦合｡信息、自变量、参数、数据等，可以经由包括存储共享、信息传递、令牌传递、网络传输等任一合适方式，被传递、转发或发射｡

在上下文中所称“计算机设备”，是指可以通过运行预定程序或指令来执行数值计算和/或逻辑计算等预定处理过程的电子设备，其至少可以包括处理器与存储器，其中由处理器执行在存储器中预存的程序指令来执行预定处理过程，或是由ASIC、FPGA、DSP等硬件执行预定处理过程，或是由上述二者组合来实现。

上述“计算机设备”通常以通用计算机设备的形式表现，其组件可以包括但不限于：一个或者多个处理器或者处理单元、系统存储器。系统存储器可以包括易失性存储器形式的计算机可读介质，例如随机存取存储器（RAM）和/或高速缓存存储器。“计算机设备”可以进一步包括其它可移动/不可移动的、易失性/非易失性计算机可读存储介质。存储器可以包括至少一个计算机程序产品，该计算机程序产品具有一组（例如至少一个）程序模块，这些程序模块被配置以执行本公开各实施例的功能和/或方法。处理器通过运行存储在存储器中的程序，从而执行各种功能应用以及数据处理。

例如，存储器中存储有用于执行本公开多个实施例的各项功能和处理的计算机程序，处理器执行相应计算机程序时，本公开的多个实施例被实现。

典型地，计算机设备例如可以是用户设备或网络设备，甚至是两者的集合。其中，所述用户设备包括但不限于个人计算机（PC）、笔记本电脑、移动终端等，所述移动终端包括但不限于智能手机、平板电脑等；所述网络设备包括但不限于单个网络服务器、多个网络服务器组成的服务器组或基于云计算（Cloud Computing）的由大量计算机或网络服务器构成的云，其中，云计算是分布式计算的一种，由一群松散耦合的计算机集组成的一个超级虚拟计算机。其中，所述计算机设备可单独运行来实现本公开的各实施例，也可接入网络并通过与网络中的其他计算机设备的交互操作来实现本公开的各实施例。其中，所述计算机设备所处的网络包括但不限于互联网、广域网、城域网、局域网、VPN网络等。

需要说明的是，所述用户设备、网络设备和网络等仅为举例，其他现有的或今后可能出现的计算设备或网络如可适用于本公开的实施例，也应包含在本公开的保护范围以内，并以引用方式包含于此。

这里所公开的具体结构和功能细节仅仅是代表性的，并且是用于描述本公开的示例性实施例的目的。但是本公开的各实施例可以通过许多替换形式来具体实现，并且不应当被解释成仅仅受限于这里所阐述的实施例。

应当理解的是，虽然在这里可能使用了术语“第一”、“第二”等等来描述各个单元，但是这些单元不应当受这些术语限制。使用这些术语仅仅是为了将一个单元与另一个单元进行区分。举例来说，在不背离示例性实施例的范围的情况下，第一单元可以被称为第二单元，并且类似地第二单元可以被称为第一单元。这里所使用的术语“和/或”包括其中一个或更多所列出的相关联项目的任意和所有组合。

这里所使用的术语仅仅是为了描述具体实施例而不意图限制示例性实施例。除非上下文明确地另有所指，否则这里所使用的单数形式“一个”、“一项”还意图包括复数。还应当理解的是，这里所使用的术语“包括”和/或“包含”规定所陈述的特征、整数、步骤、操作、单元和/或组件的存在，而不排除存在或添加一个或更多其他特征、整数、步骤、操作、单元、组件和/或其组合。

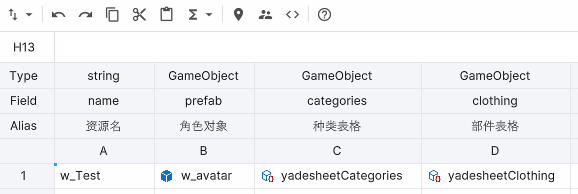
还应当提到的是，在一些替换实现方式中，所提到的功能/动作可以按照不同于附图中标示的顺序发生。举例来说，取决于所涉及的功能/动作，相继示出的两幅图实际上可以基本上同时执行或者有时可以按照相反的顺序来执行。

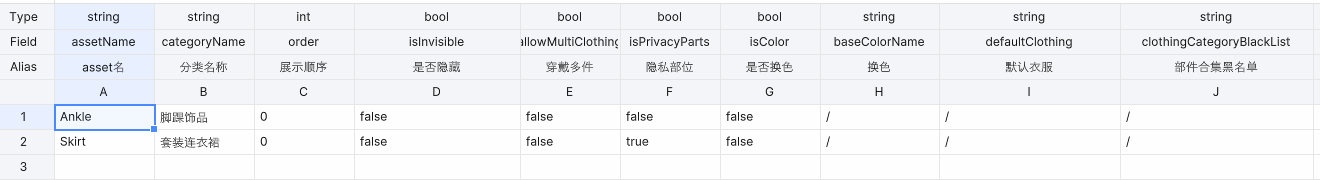
参阅图1，其中示出根据本公开的一个实施例的一种配置虚拟人物的换装资源的方法流程图。

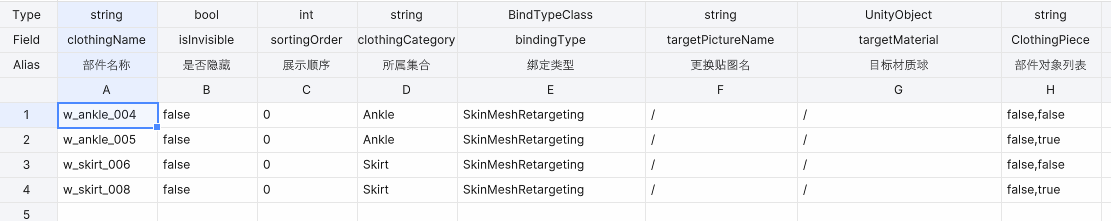
如图1所示，在步骤S1中，计算机设备获取换装资源的配置数据，其中，所述配置数据包括角色构建表，角色部件种类表和角色部件细节表；在步骤S2中，计算机设备根据所述配置数据，在可视化编辑工具中生成资源对象的预制件；在步骤S3中，计算机设备根据所述资源对象的预制件的列表，生成相应角色的模板。

具体地，在步骤S1中，计算机设备获取换装资源的配置数据。

在此，根据一个示例，换装资源可以预先由人工配置，并将其配置数据存储为Excel格式的角色构建表、角色部件种类表以及角色部件细节表。







在步骤S2中，计算机设备根据换装资源的配置数据，在可视化编辑工具中生成资源对象的预制件。

配合参阅图2示出的角色构建表，计算机设备根据该角色构件表，把每行元素生成对应的角色预制件，并把角色构件表中在该行定义的属性映射到所生成的角色预制件上。

配合参阅图3示出的角色部件种类表，计算机设备根据该角色部件种类表，把每行元素生成对应的种类预制件，并把角色部件种类表中在该行定义的属性映射到所生成的种类预制件上。

配合参阅图4示出的角色部件细节表，计算机设备根据该角色部件细节表，把每行元素生成对应的部件预制件，并把角色部件细节表中在该行定义的属性映射到所生成的部件预制件上。

根据一个示例，可视化编辑工具为Unity编辑器。计算机设备根据配置表里的对应关系批量把fbx文件生成符合标准的Unity预制件。

在步骤S3中，计算机设备根据资源对象的预制件的列表，生成相应角色的模板。

根据角色预制件的列表、种类预制件的列表以及部件预制件的列表，计算机设备生成至少一个角色的模板。

根据本公开的一个示例，图1所示过程还包括步骤S4。在步骤S4中，计算机设备根据第一角色的模板，在可视化编辑工具中实例化该第一角色，并在可视化编辑工具中对该第一角色的一个或多个部件进行编辑操作。

根据本公开的一个示例，Unity编辑器根据第一角色对应的种类配置和部件配置，获得相应部件的模板，并对各部件的模板进行组合，以获得第一角色的模板。随后，Unity编辑器还可以对这些部件进行替换、删除等编辑操作。

需要注意的是，本公开的各实施例可在软件和/或软件与硬件的组合体中被实施，例如，可采用专用集成电路（ASIC）、通用目的计算机或任何其他类似硬件设备来实现。在一个实施例中，本公开的各实施例的软件程序可以通过处理器执行以实现上文所述步骤或功能。同样地，本公开的各实施例的软件程序（包括相关的数据结构）可以被存储到计算机可读记录介质中，例如，RAM存储器，磁或光驱动器或软磁盘及类似设备。另外，本公开的各实施例的一些步骤或功能可采用硬件来实现，例如，作为与处理器配合从而执行各个步骤或功能的电路。

另外，本公开的各实施例的至少一部分可被应用为计算机程序产品，例如计算机程序指令，当其被计算设备执行时，通过该计算设备的操作，可以调用或提供根据本公开的各实施例的方法和/或技术方案。而调用/提供本公开的各实施例的方法的程序指令，可能被存储在固定的或可移动的记录介质中，和/或通过广播或其他信号承载媒体中的数据流而被传输，和/或被存储在根据所述程序指令运行的计算设备的工作存储器中。

对于本领域技术人员而言，显然本公开的各实施例不限于上述示范性实施例的细节，而且在不背离本公开的各实施例的精神或基本特征的情况下，能够以其他的具体形式实现本公开的各实施例。因此，无论从哪一点来看，均应将实施例看作是示范性的，而且是非限制性的，本公开的各实施例的范围由所附权利要求而不是上述说明限定，因此旨在将落在权利要求的等同要件的含义和范围内的所有变化涵括在本公开的各实施例内。不应将权利要求中的任何附图标记视为限制所涉及的权利要求。此外，显然“包括”一词不排除其他单元或步骤，单数不排除复数。系统权利要求中陈述的多个单元或装置也可以由一个单元或装置通过软件或者硬件来实现。第一，第二等词语用来表示名称，而并不表示任何特定的顺序。

**权 利 要 求 书**

1. 一种配置虚拟人物的换装资源的方法，其中，该方法包括以下步骤：

配置换装资源的模型数据，其中，所述模型数据包括种类配置数据和部件配置数据；

根据所述模型数据构建换装配置关系表，其中，所述换装配置关系表包括角色构建表、种类构建表和模型部件构建表；

根据所述换装配置关系表，在可视化编辑工具中生成资源对象的模板，其中，所述模板包括角色模板和部件模板。

2. 根据权利要求1所述的方法，其中，该方法还包括以下步骤：

根据第一角色的模板，在所述可视化编辑工具中实例化所述第一角色；

在所述可视化编辑工具中对所述第一角色的一个或多个部件进行编辑操作。

3. 根据权利要求1所述的方法，其中，所述可视化编辑工具为Unity编辑器。

4. 根据权利要求3所述的方法，其中，所述构建换装配置关系表的步骤具体包括：

将所述模型数据转换为所述Unity编辑器能够识别的fab文件格式的换装配置关系表。

5. 根据权利要求3所述的方法，其中，所述资源对象的模板为prefab预制件。

6. 根据权利要求1至5中任一项所述的方法，其中，所述资源对象的模板进一步包括角色配置表、种类配置表和部件配置表。

7. 根据权利要求1至6中任一项所述的方法，其中，所述实例化的步骤具体包括：

根据所述第一角色对应的种类配置和部件配置，获得相应部件的模板，并对所述部件的模板进行组合，以获得所述第一角色的模板。

21. 一种设备，其中，该设备包括处理器和存储器，所述存储器包括计算机可执行指令，当所述计算机可执行指令被所述处理器执行时，该设备被配置来执行如权利要求1至6中任一项所述的方法。

**说 明 书 摘 要**

本公开的各实施例的目的是

**说 明 书 附 图**

图 1

图 2（a） 图 2（b）

图 3

**摘 要 附 图**

（图1）