



(12) 发明专利

(10) 授权公告号 CN 107436759 B

(45) 授权公告日 2020.12.04

(21) 申请号 201710464813.X

G06F 16/248 (2019.01)

(22) 申请日 2017.06.19

G06F 16/22 (2019.01)

(65) 同一申请的已公布的文献号
申请公布号 CN 107436759 A

(56) 对比文件
US 2015347478 A1, 2015.12.03

(43) 申请公布日 2017.12.05

审查员 温兰兰

(73) 专利权人 上海高顿教育培训有限公司
地址 200083 上海市虹口区西江湾路388号
B栋8层

(72) 发明人 钱升

(74) 专利代理机构 上海科盛知识产权代理有限公司 31225

代理人 叶敏华

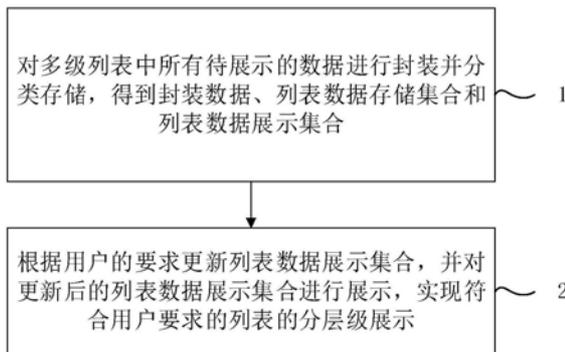
(51) Int. Cl.
G06F 9/451 (2018.01)
G06F 16/245 (2019.01)

权利要求书1页 说明书4页 附图2页

(54) 发明名称
一种用于安卓系统的多级列表展示方法

(57) 摘要

本发明涉及一种用于安卓系统的多级列表展示方法,所述方法包括下列步骤:对多级列表中所有待展示的数据进行封装并分类存储,得到封装数据、列表数据存储集合和列表数据展示集合;根据用户的要求更新列表数据展示集合,并对更新后的列表数据展示集合进行展示,实现符合用户要求的列表的分层级展示。与现有技术相比,本发明具有展示速度快、展示效率高以及实现方便等优点。



1. 一种用于安卓系统的多级列表展示方法,其特征在于,所述方法包括下列步骤:

1) 对多级列表中所有待展示的数据进行封装并分类存储,得到封装数据、列表数据存储集合和列表数据展示集合;

2) 根据用户的要求更新列表数据展示集合,并对更新后的列表数据展示集合进行展示,实现符合用户要求的列表的分层级展示;

所述步骤1) 具体为:

11) 对多级列表中所有待展示的数据进行封装,得到所有封装数据;

12) 将所有封装数据存储至同一个集合,得到列表数据存储集合;

13) 将第一级的封装数据存储至同一个集合,作为列表数据展示集合的初始集合;

所述步骤11) 具体为:

111) 定义子列表指向参数与子列表显示参数,并为子列表显示参数设置初始值,所述子列表指向参数和子列表显示参数的数量与待展示的数据数量一致;

112) 分别将每一个待展示的数据与子列表指向参数和子列表显示参数封装在一起,作为封装数据;

所述步骤2) 具体为:

21) 根据用户的要求,修改相应的封装数据的参数值,更新列表数据存储集合;

22) 根据所有封装数据的参数值,对列表数据展示集合进行更新;

23) 展示更新后的列表数据展示集合内所有的待展示的数据,实现符合用户要求的列表的分层级展示;

所述步骤22) 具体为:

221) 删除列表数据展示集合中的所有非第一级的封装数据;

222) 从列表数据存储集合中,得到所有子列表显示参数值为显示的封装数据;

223) 对步骤222) 中得到的封装数据,读取相应的子列表指向参数;

224) 根据步骤223) 中子列表指向参数指向的位置,从列表数据存储集合中提取相应的封装数据,添加至列表数据展示集合中;

所述子列表显示参数用于表明对应的封装数据是否显示下一级数据,所述子列表显示参数的初始值为不显示。

2. 根据权利要求1所述的用于安卓系统的多级列表展示方法,其特征在于,所述子列表指向参数用于指向对应的封装数据的下一级数据在列表数据存储集合中的位置,在对应的待展示数据不存在下一级数据时,所述子列表指向参数指向空集。

3. 根据权利要求1所述的用于安卓系统的多级列表展示方法,其特征在于,所述步骤21) 具体为:

211) 根据用户的要求,读取需要展示下级的封装数据的子列表指向参数;

212) 根据读取结果,判断需要展示下级的封装数据是否存在下级,若是则进入步骤213),若否则结束操作;

213) 将需要展示下级的封装数据中的子列表显示参数值改为反向值,更新列表数据存储集合。

一种用于安卓系统的多级列表展示方法

技术领域

[0001] 本发明涉及系统开发领域,尤其是涉及一种用于安卓系统的多级列表展示方法。

背景技术

[0002] 随着互联网时代的发展,业务上的需求层出不穷,而APP端现有的控件已经基本满足不了现在的开发需求。内容变得越来越复杂,于是多层级的展现形式也是越来越多,层级最多可打达到5、6级。

[0003] 安卓系统本身提供的开发工具只适用于2级列表的展示,为了达到多级展示的目的,现有的方法多是通过使用普通的系统嵌套来进行多级列表的展示,这种展示方式是将多级列表按照本身的层级进行层层嵌套并存储,在展示时需要将所有层级从上至下遍历直至寻找到待展示的层级,这种方式导致降低了展示的速度和效率,同时,在对多级列表本身的内容进行修改时,也由于层级的嵌套而大大增加了修改的难度和复杂程度。

发明内容

[0004] 本发明的目的是针对上述问题提供一种用于安卓系统的多级列表展示方法。

[0005] 本发明的目的可以通过以下技术方案来实现:

[0006] 一种用于安卓系统的多级列表展示方法,所述方法包括下列步骤:

[0007] 1) 对多级列表中所有待展示的数据进行封装并分类存储,得到封装数据、列表数据存储集合和列表数据展示集合;

[0008] 2) 根据用户的要求更新列表数据展示集合,并对更新后的列表数据展示集合进行展示,实现符合用户要求的列表的分层级展示。

[0009] 所述步骤1) 具体为:

[0010] 11) 对多级列表中所有待展示的数据进行封装,得到所有封装数据;

[0011] 12) 将所有封装数据存储至同一个集合,得到列表数据存储集合;

[0012] 13) 将第一级的封装数据存储至同一个集合,作为列表数据展示集合的初始集合。

[0013] 所述步骤11) 具体为:

[0014] 111) 定义子列表指向参数与子列表显示参数,并为子列表显示参数设置初始值,所述子列表指向参数和子列表显示参数的数量与待展示的数据数量一致;

[0015] 112) 分别将每一个待展示的数据与子列表指向参数和子列表显示参数封装在一起,作为封装数据。

[0016] 所述子列表指向参数用于指向对应的封装数据的下一级数据在列表数据存储集合中的位置,在对应的待展示数据不存在下一级数据时,所述子列表指向参数指向空集。

[0017] 所述子列表显示参数用于表明对应的封装数据是否显示下一级数据,所述子列表显示参数的初始值为不显示。

[0018] 所述步骤2) 具体为:

[0019] 21) 根据用户的要求,修改相应的封装数据的参数值,更新列表数据存储集合;

- [0020] 22) 根据所有封装数据的参数值,对列表数据展示集合进行更新;
- [0021] 23) 展示更新后的列表数据展示集合内所有的待展示的数据,实现符合用户要求的列表的分层级展示。
- [0022] 所述步骤21) 具体为:
- [0023] 211) 根据用户的要求,读取需要展示下级的封装数据的子列表指向参数;
- [0024] 212) 根据读取结果,判断需要展示下级的封装数据是否存在下级,若是则进入步骤213),若否则结束操作;
- [0025] 213) 将需要展示下级的封装数据中的子列表显示参数值改为反向值,更新列表数据存储集合。
- [0026] 所述步骤22) 具体为:
- [0027] 221) 删除列表数据展示集合中的所有非第一级的封装数据;
- [0028] 222) 从列表数据存储集合中,得到所有子列表显示参数值为显示的封装数据;
- [0029] 223) 对步骤222) 中得到的封装数据,读取相应的子列表指向参数;
- [0030] 224) 根据步骤223) 中子列表指向参数指向的位置,从列表数据存储集合中提取相应的封装数据,添加至列表数据展示集合中。
- [0031] 与现有技术相比,本发明具有以下有益效果:
- [0032] (1) 通过将待展示的数据进行封装,从而将数据的层级通过子列表指向参数的方式表达出来,比起直接按照层级进行层层嵌套相比,这种利用参数进行指向的方式大大降低了数据之间的关联性,从而避免了在分层级展示的过程中需要将所有层级从上之下遍历的过程,大大降低了展示过程的复杂程度,提高了展示速度和效率。
- [0033] (2) 将所有的封装数据均存储于列表数据存储集合中,将待展示的数据存储至列表数据展示集合中,在用户更改展示层级时,通过修改封装数据的相应参数,作为列表数据展示集合的更新依据,这种存储方式与嵌套式存储相比,更为简单且便于开发人员修改,提高展示效率的同时也降低了开发人员的工作强度。
- [0034] (3) 在对封装数据的参数进行修改时,首先读取待修改的封装数据的子列表指向参数,并在子列表指向参数指向空集时直接结束操作,对子列表指向参数不指向空集的封装数据进行子列表显示参数的修改,通过一次预判断,避免了对不存在下一级的子列表进行无意义的操作,提高了展示效率。
- [0035] (4) 在更新列表数据展示集合的过程中,保留所有第一级的封装数据,这种方式表明了总是会显示第一级的数据,从而使得后续的操作不受层级的限制,举例来说,在有三级数据存在时,若想显示第三级数据,只需考虑指向第三级的第二级数据的子列表指向参数,这是由于第一级数据的持续保留,保证了之前的操作一定会由第一级数据指向第二级数据,而不会发生第二级数据未添加至列表数据展示集合而只添加了第三级数据的情况,从而进一步降低了各层级之间的嵌套程度,提高了展示效率。

附图说明

- [0036] 图1为本发明的方法流程图;
- [0037] 图2为本实施例中各层级的指向关系图;
- [0038] 图3为本实施例中的多级列表展示流程图。

具体实施方式

[0039] 下面结合附图和具体实施例对本发明进行详细说明。本实施例以本发明技术方案为前提进行实施,给出了详细的实施方式和具体的操作过程,但本发明的保护范围不限于下述的实施例。

[0040] 如图1所示,本实施例提出了一种用于安卓系统的多级列表展示方法,所述方法包括下列步骤:

[0041] 1) 对多级列表中所有待展示的数据进行封装并分类存储,得到封装数据、列表数据存储集合和列表数据展示集合:

[0042] 11) 对多级列表中所有待展示的数据进行封装,得到所有封装数据:

[0043] 111) 定义子列表指向参数与子列表显示参数(子列表指向参数用于指向对应的封装数据的下一级数据在列表数据存储集合中的位置,在对应的待展示数据不存在下一级数据时,子列表指向参数指向空集;子列表显示参数用于表明对应的封装数据是否显示下一级数据,子列表显示参数的初始值为不显示),并为子列表显示参数设置初始值,所述子列表指向参数和子列表显示参数的数量与待展示的数据数量一致;

[0044] 112) 分别将每一个待展示的数据与子列表指向参数和子列表显示参数封装在一起,作为封装数据;

[0045] 12) 将所有封装数据存储至同一个集合,得到列表数据存储集合;

[0046] 13) 将第一级的封装数据存储至同一个集合,作为列表数据展示集合的初始集合;

[0047] 2) 根据用户的要求更新列表数据展示集合,并对更新后的列表数据展示集合进行展示,实现符合用户要求的列表的分层级展示,具体为:

[0048] 21) 根据用户的要求,修改相应的封装数据的参数值,更新列表数据存储集合:

[0049] 211) 根据用户的要求,读取需要展示下级的封装数据的子列表指向参数;

[0050] 212) 根据读取结果,判断需要展示下级的封装数据是否存在下级,若是则进入步骤213),若否则结束操作;

[0051] 213) 将需要展示下级的封装数据中的子列表显示参数值改为反向值,更新列表数据存储集合;

[0052] 22) 根据所有封装数据的参数值,对列表数据展示集合进行更新:

[0053] 221) 删除列表数据展示集合中的所有非第一级的封装数据;

[0054] 222) 从列表数据存储集合中,得到所有子列表显示参数值为显示的封装数据;

[0055] 223) 对步骤222)中得到的封装数据,读取相应的子列表指向参数;

[0056] 224) 根据步骤223)中子列表指向参数指向的位置,从列表数据存储集合中提取相应的封装数据,添加至列表数据展示集合中;

[0057] 23) 展示更新后的列表数据展示集合内所有的待展示的数据,实现符合用户要求的列表的分层级展示。

[0058] 本实施例中的展示列表层级如图2所示,依据上述步骤对图2中的展示列表进行分层级展示的过程如图3所示,具体为:

[0059] 首先创建两个集合 O, S 。其中 O 集合为原始数据, S 集合为显示数据。假设 x 为集合中的每一个元素, x 中包含数据如下: $level$:当前视图的层级序号, $level$ 会随着业务的层级递增,如第一层级: $level=0$;第二层级: $level=1$,以此类推。 $hasChild$:是否有子视图;

showChild:是否显示子视图;假设 $O=[x_1, x_2, x_3 \cdots x_n]$,并且 x_1, x_{10}, x_{11} 中 $level=0$,为第一层级;默认显示第一级,则 $S=[x_1, x_{10}, x_{11}]$ 。 x_2 为 x_1 的子视图,则 $x_2.level=1$; $x_3, x_4 \dots x_9$ 为 x_2 的子视图, $x_3.level=2, x_4.level=2 \cdots x_9.level=2$ 。

[0060] 假设用户点击了 x_1 ,即可以获取 x_1 中所有数据,则可以在 O 集合(原始数据)中找到 x_1 ,并且判断 x_1 是否存在子视图,如果存在,则将先 x_1 的showChild设置为显示(showChild=true),清除原来显示的 S 集合,再将当前显示(showChild=true)的视图添加到 S 集合中,则集合中数据为: $S=[x_1, x_2, x_{10}, x_{11}]$;当点击集合中 x_2 时,再次去 O 集合找到 x_2 ,判断如果有子视图,则设置为显示(showChild=true),并且清除 S 集合,将所有的showChild=true的视图添加到 S 集合中,所以 $S=[x_1, x_2, x_3 \cdots x_9, x_{10}, x_{11}]$,这样就完成了三级目录的显示。如果当前视图已经为显示的(showChild=true),那么则设置不显示,showChild=false。

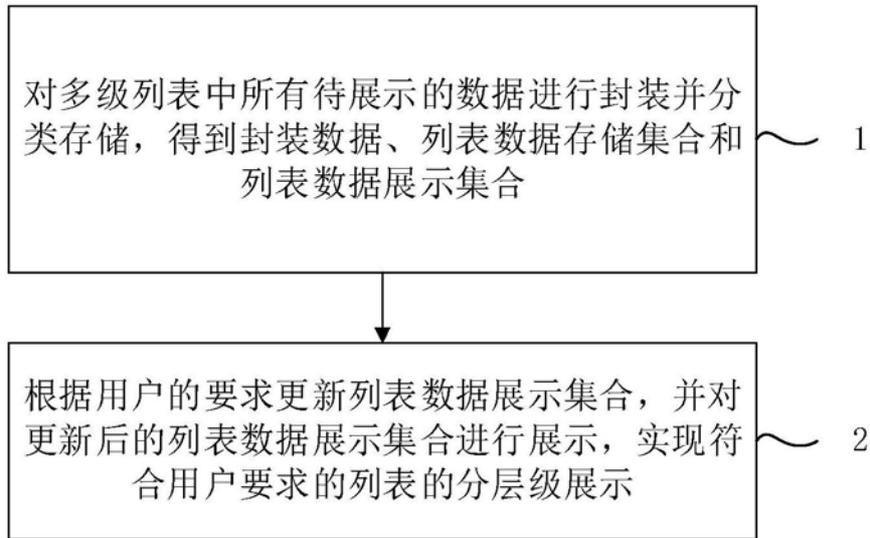


图1

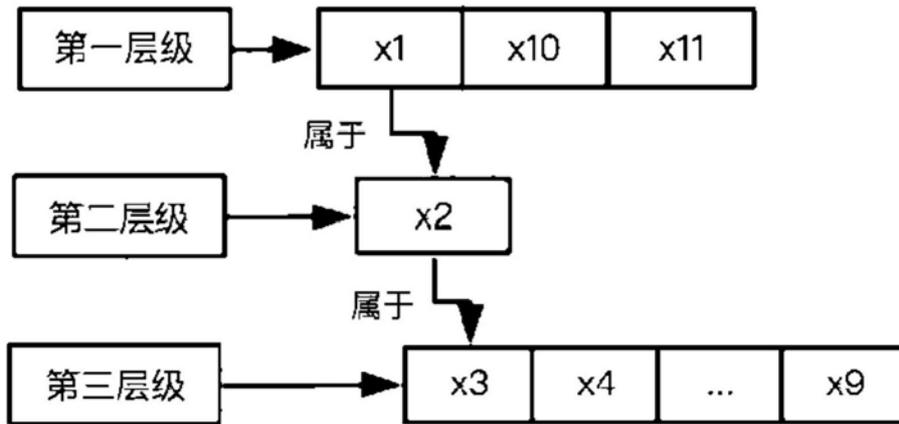


图2

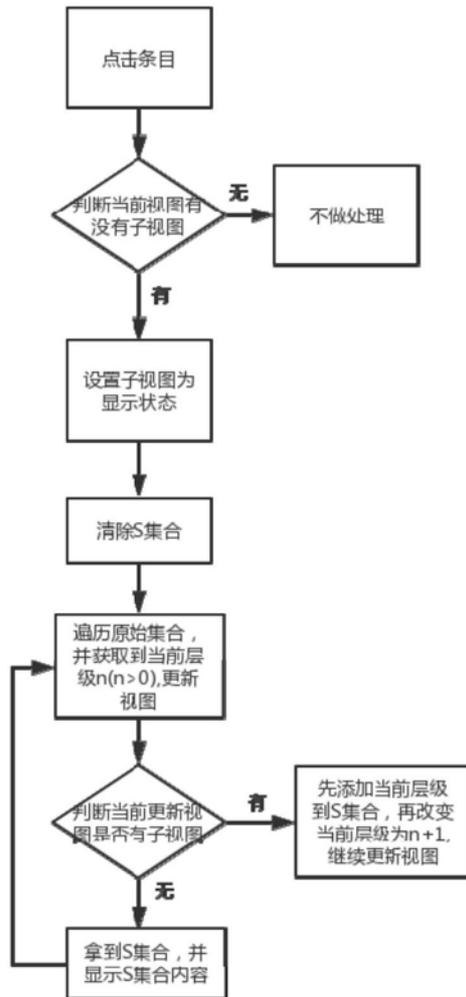


图3