

专利申请申报书

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 申报  部门 | | 芒果超媒/快乐阳光/商业化技术中心 | | | 日期 | | 2025.6.27 |
| 申报  名称 | | 一种通用性增加人机交互的按压反馈动效实现 | | | 申报类别 | | √发明 |
| 第一发明人 | | 廖荣升 | | |
| 第一发明人身份证号码 | | 432524199202011455 | | |
| 应用于哪些  载体（例如计算机、手机、电视等） | | 手机 | | | | | |
| 1. 技术来源（请在下列对应的口中选择）：  √ 独创；口 参考已知技术+创新；口 仿制；口 仿制+改进；口 引进；口 引进+改进。   1. 设计基础：   目前主流应用用户与手机的点击交互往往没有点击效果，较为生硬，为了增加人机的互动性，google按钮控件会增加一层阴影层的反馈效果，来改善体验，但这仅仅是针对于正常按钮，太过单一，并且互动性也不够强。  本专利旨在封装与业务解耦的通用按压伸缩效果反馈控件，不受使用场景与控件类型影响，在尽可能不增加开发成本的情况下，增加人机互动性，提高用户交互粘度。  3. 设计部位：  所有需要增加交互伸缩效果的页面  4. 工艺方法或结构型式否是独一无二的（请在下列对应的口中选择）：  √ 独家所有；口 其它公司也有类似的； 口 未发现其它公司有。 | | | | | | | |
| 技术联系人 | 廖荣升 | | | 技术联系人电话 | |  | |
| 发明人名单（作为专利发明人上报）： | | | 刘比奥、杨亮波、廖荣升 | | | | |
| 实现流程图：  流程图 | | | | | | | |
| **·设计要点说明：**（结合上述附图指出设计要点）   1. 点击时增加伸缩反馈，增加人机的交互感，改善用户体验，提高用户点击粘度； 2. 按压动效与业务页面完全解耦，业务方无需修改页面逻辑来适配动效，也不用管动效是怎么实现的，什么时候销毁，提高通用性，减少调用方的开发时间； 3. 调用方可以自定义场景，是否触摸屏幕就动效还是接下来不滑动才动效，以及自定义按压缩放过程中的阴影效果，缩放圆角，按压缩放程度，哪些页面区域需要动效等； 4. 链式调用，调用方式足够简洁,添加效果的开发量少，且不影响业务功能。 | | | | | | | |
| **功能说明：**（结合附图说明交互工作过程，及能实现的相应效果）  1、按压反馈效果可用于两种场景，一中场景是只要触摸屏幕就执行动效，另一种场景是触摸屏幕后，接下来是不滑动才动效，如果接下来是滑动，则不动效；  2、对应的布局被点击时，有一个渐渐缩放效果，并且阴影层逐渐加深，执行完成时，有震动反馈，说明按压到底了。  3、松手时，缩放效果逐渐还原，且阴影层逐渐消失  效果事例图1：  右上方区域为按压后的效果，会动画缩放。需要执行缩放的页面内容有标注出来的1，2，3，  如果有其它页面场景，只需通过链式调用将需要动画的页面元素传递给按压动画控件，无需修改布局或者业务逻辑来适配动画。    按压效果事例图  **技术实现：**   1. 一个完整的按压伸缩动画过程由两个部分组成，按下时的缩放动画，松手时的放大还原动画，   不管时按下还是松手时的动画，实现原理一样，动画中包含多个view，每个view包含X,Y方向的平移与伸缩四个动画组成模块，将多个view的多个动画合并成一个动画实现，具体实现原理：  对于某一个view来说，X,Y轴平移与缩放的每个动画对应一个PropertyValuesHolder，此时会有四个PropertyValuesHolder，通过调用属性动画的ofPropertyValuesHolder方法，把当前的View，以及对应的四个动画PropertyValuesHolder作为入参传入，合并成一个ObjectAnimator属性动画，这样，一个view的多个动画就组成了一个动画ObjectAnimator，多个View对应多个ObjectAnimator，再通过AnimatorSet类的playTogether方法，将多个属性动画对象ObjectAnimator合并成一个AnimatorSet，这样，就保证了多个view的多个动画合并同步执行。  实现示意图如下：  最终动画的组成   1. X,Y轴方向上的平移动画的偏移量计算：    1. 在所有需要动画的view的集合中，取宽\*高最大的view最为参考目标view，view的中心点为参考点;    2. 中心点的计算：中心点X轴坐标Ox = 当前控件在屏幕中的X轴位置 + 当前控件宽度width的一半， 中心点Y轴坐标Oy = 当前控件在屏幕中的Y轴位置 + 当前控件高度height的一半;   2.3. 再缩放的过程中，不同大小的view，缩放同样的百分比，实际缩放的距离并不一样；同时，到目标View之间的间距，随着缩放动画的执行，也应该同比增大或者减少，要从视觉上保证动画过程中所有view的相对位置不变，在每一个缩放过程中的时间节点，需要对各个view相对于参考目标view的这两个因素位移差同步做X轴方向以及Y轴方向上的平移动画，来还实现偏移纠正。  2.4. 相对与参考目标view,其它view的相对位置可以分为四个方向：左上，右上，左下，右下。不同方向在每一个时间点的位移差，计算方式不同，X轴跟Y轴方向上的位移差由两个部分组成，  2.5. 按下动作执行缩放效果时，取一个控件，控件下标系数暂时定义为1，X轴动画从0到offsetX,  Y轴动画从0到offsetY, 计算分析如下：  用到的参数说明：  缩放系数scaling：按压时的最终缩放大小占起始状态的百分比，由调用方根据不同的业务场景来设置，默认为起始状态的0.95  marginLeft：表示当前控件到目标控件的左侧间距  marginTop：表示当前控件到目标控件的上侧间距  marginRight:表示当前控件到目标控件的右侧间距  marginBottom:表示当前控件到目标控件的下侧间距  2.5.1 X轴方向的偏移量计算  2.5.1.1如果当前控件中心点在参考点的左侧  第一部分：缩放时当前控件与目标控件宽度的最大偏差的负数，既缩放完成时的宽度偏差量的负数，计算方式如下：  offsetX1 = -缩放系数scaling \* (目标控件的宽度Ow-当前控件的宽度Ow1) / 2，既 offsetX1 = -scaling \* (Ow -Ow1) / 2  第二部分，表示当前控件到目标控件左侧的间距：公式为offsetX2 = -marginLeft  最终结果 offsetX = offsetX1 + offsetX2 = -scaling \* (Ow -Ow1) / 2 - marginLeft  2.5.1.2如果当前控件中心点在参考点右侧时  第一部分：缩放时当前控件与目标控件宽度的最大偏差，既缩放完成时的偏差量，计算方式如下：  offsetX1 = 缩放系数scaling \* (当前控件中心点Ox1到参考点Ox的距离) / 2，既 offsetX1 = scaling \* (Ox -Ox1) / 2  最终结果 offsetX = offsetX1 + offsetX2 = scaling \* (Ox -Ox1) / 2 + marginRight  2.5.2 Y轴方向的偏移量计算  2.5.2.1 如果当前控件中心点在参考点的上侧  第一部分：缩放时当前控件与目标控件高度的最大偏差，既缩放完成时的高度偏差量，计算方式如下：  offsetY1 = 缩放系数scaling \*（目标控件的高度Oh - 当前控件的 高度Oh1）/2 , 既 offsetY1 = scaling \*(Oh-Oh1)/2  第二部分：表示当前控件到目标控件上侧的间距的负数，公式为offsetY2 = -marginTop  最终结果：offsetY = offsetY1 +offsetY2 = scaling \*(Oh-Oh1)/2 -marginTop  2.5.2.2 如果当前控件中心点在参考点的下侧  缩放时当前控件与目标控件高度的最大偏差的负数，既缩放完成时的高度偏差量的负数，计算方式如下：  offetY1 = -缩放系数scaling \*（目标控件的高度Oh - 当前控件的 高度Oh1）/2 , 既 offsetY1 = -scaling \*(Oh-Oh1)/2  第二部分：表示为当前控件到目标控件下侧的间距的二分之一，公式为offsetY2 = marginBottom/2  最终结果：offsetY = offsetY1 +offsetY2 = -scaling \* (Oh-Oh1)/2 + marginBottom/2  2.6. 松手时，还原扩大动画执行时,平移动画的计算与步骤5相同，但动画反向，X轴方向的平移动画从offsetX到0，Y轴方向的动画从offsetY到0  2.7.具体相对于参考控件不同位置的其它控件对应的计算如下图：  偏移量计算   1. 按下与松手的动画设置贝塞尔差值器来控制执行程度的快慢，开始时缓慢，中间过程速度逐渐加快，快结束时，逐渐平缓减速到完成，使反馈效果动画不是因为线性执行而显得突兀，视觉效果更加平滑，贝塞尔曲线系数由UI设计师去不断测试调整，确定视觉效果最好的系数。   3.1 动画时间贝塞尔曲线，按下时的曲线系数(0.33, 0, 0.67, 1)：  **down**  3.2 动画时间贝塞尔曲线，松手时的曲线系数(0.2, 0, 0,2, 1)：    up  动画过程的遮罩阴影层，调用方通过链式调用传入的系列参数动态生成的前景foregroundDrawable来实现遮罩层渐变功能，如setNeedForeground方法设置是否需要遮罩，setCircular方法设置是否圆形遮罩，setColor方法设置遮罩颜色，setCornerRadius方法设置遮罩指定角的圆角度数，或者setForegroundDrawable直接设置遮罩,内部具体实现为如果直接设置的setForegroundDrawable则直接使用，如果调用的是其它方式，  如果是圆角，则通过ShapeDrawable的OvalShape动态生成一个圆角foregroundDrawable，  如果遮罩设置的是圆角，则通过RoundRectShape的outerRadii数组动态生成对应圆角的foregroundDrawable   1. 监听动画的执行过程，根据动画的执行程度，动态修改前景的透明度0-255之间变化，达到按下时随着动画的执行，遮罩阴影层颜色越来越深，松手时，颜色越来越浅的动态变化效果   **优势：**   1. 不限定于按钮控件，页面的所有内容都能实现点击反馈效果，且比普通按钮交互性更强； 2. 可实现滑动时没有效果，点击时才有效果的场景； 3. 独立于业务页面，低耦合，可复用性高。 | | | | | | | |