|  |  |
| --- | --- |
| 交底书名称 | 订单公益项目-序列帧动画组件的实现 |
| 技术联系人姓名 | 李永健 |
| 技术联系人电话 | 13269971637 |
| 技术联系人Email | liyongjian@jd.com |

（技术联系人信息用于与外部代理沟通，发明人信息在ERP系统中填写）

注意事项：

1、代理人并不是技术专家，交底书要使代理人能看懂，尤其是“3.2 本发明的完整技术方案”部分一定要写得全面、清楚、通俗易懂。

2、在与专利代理人沟通时，应积极配合（禁止通过私人邮箱与代理人沟通）。

3、常用检索网站：http://www.innojoy.com/search/index.html（大为搜索引擎）。

# 1. 现有技术

/\* 应记载某个应用场景或者解决某个技术问题当前所采用的技术，可以概述，也可以仅给出参考文献的链接或相关专利号。

对于toC领域的前端业务来说，怎样通过技术手段提升用户体验和增强用户粘性，是除了业务逻辑外，首要解决的问题，而利用前端动画来提升用户交互体验是解决该痛点的一个有效方案。目前，实现前端动画的主流方案有原生CSS方案、原生JS方案、Canvas方案和第三方动画框架，在性能优化上，有些方案在细节上处理的不是很完善，所以在订单公益项目的序列帧动画组件的实现中加入性能优化的细节和CSS的新属性，以达到优化动画组件性能的诉求。

# 2. 现有技术的缺点

/\* 需要指出现有技术存在的缺点，本申请也不能克服的缺点无需提供。

/\* 应根据现有技术的实现过程，针对性地说明缺点产生的原因。

1. 资源浪费；大量引用第三方动画框架，造成不必要的资源占用和资源浪费，很多场景下可以通过原生CSS或JS手段来自行实现；
2. 社区维护水平参差不齐：在使用第三方动画框架时，在遇到问题时，我们通常是自己主动排查或是查询社区issue，但是有很大一部分的第三方框架或组件的社区维护较低，致使某些极端问题得不到解决，调用者只能通过排查源码来解决；
3. 不必要的性能浪费：有些开发者在开发动画时不会关注页面运行的性能，从而导致不必要的性能浪费，如果页面复杂度较高的话，会占用大量的GPU资源、CPU资源和其他内存资源，直接或间接导致其他页面bug。

# 3. 本发明技术方案

## 3.1 本发明所要解决的技术问题（即发明目的）

/\* 描述本发明所要解决的技术问题，与“2.现有技术的缺点”部分指出的缺点相对应。

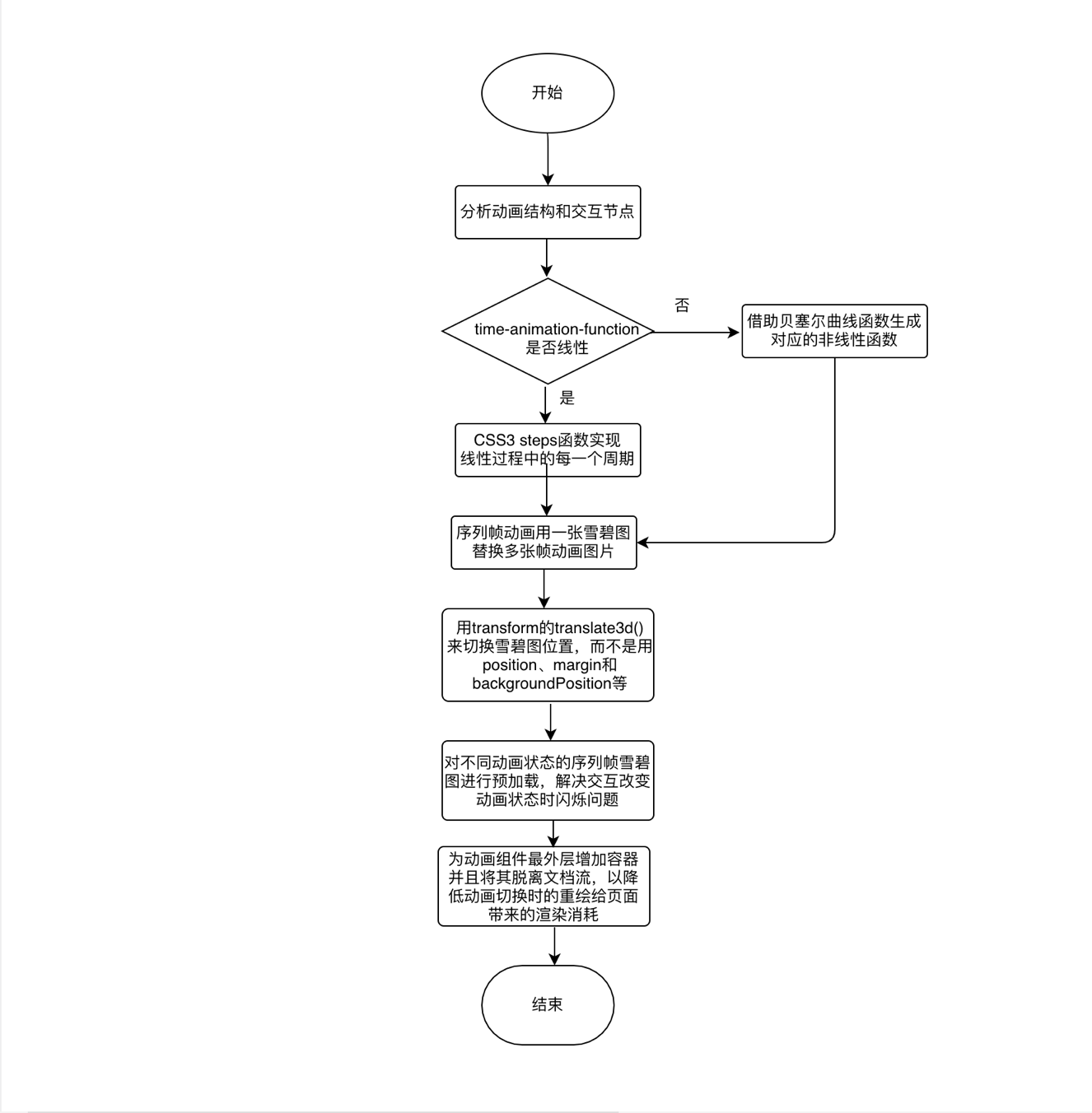
1. 降低对第三方动画框架的依赖，提升项目的独立性；
2. 降低不必要的性能浪费，提升页面性能。

## 3.2 本发明的完整技术方案

### 3.2.1 系统原理图、结构说明图或流程图

/\*如果要保护的是一个方法，应通过附图说明方法的实现步骤或数据流向；如果保护的是一个装置，应通过附图说明装置的结构和组成部件。

（图1）动画组件前端设计流程图



（图2）订单公益动画实例（示例）



（图3）订单公益动态海报组件实例（示例）



### 3.2.2 技术方案详细描述

/\* 这是本文档最重要的部分，需要详细完整的阐述；不能仅有原理没有实现细节，也不能仅有功能性介绍或操作说明；同时，为了相关人员能够更好地理解技术方案，应对除关键技术之外的、实现本技术方案必不可少的步骤或部件进行适当的说明。

/\* 在描述具体的技术方案时，必须结合附图进行说明，每个附图都应当有对应的文字描述。

序列帧动画的特点是能够定义每个动画周期的效果，而不是整个动画周期的效果，并且能够解放设计师，让设计师充分发挥，尽最大可能还原动画原型。

订单公益项目中的序列帧动画在实现时，采用CSS3规范的steps函数来作为time-animation-function，相比于采用JS，CSS3的steps函数使得动画在表现和行为上得到分离，并且不用再借助JS的定时器任务来实现每个动画周期。

1. 分析动画结构和交互节点，规划动画的关键帧和CSS3动画API；
2. 判断动画time-animation-function是否为线性，如果为线性则直接使用steps函数处理，否则需要借助贝塞尔曲线函数cubic-bezier来实现非线性变化；
3. 采用将序列帧做成雪碧图和更改雪碧图位移和位置的方案替换通过改变序列帧图片src的方式，以减少http请求数、解决图片加载闪烁的问题和方便文件管理；
4. 采用CSS3 transform: translate3d()来替换background- position、position或margin等方式切换序列帧雪碧图位置的方案，降低GPU和CPU的消耗；
5. 对动画不同状态的序列帧雪碧图进行预加载，其中通过img的src和dom的背景图引入时的预加载方式不同，避免动画切换时两种动画衔接处闪烁问题；
6. 将动画组件外层用容器包裹并脱离文档流，已减少动画切换引起的重绘导致的页面GPU性能损耗等。

## 3.3 本发明希望保护的技术创新点

/\* 指出技术方案中希望保护的技术关键点，并概括说明该关键点的技术原理。

1. 降低对第三方动画框架的依赖：降低对第三方动画框架的依赖，提升项目的独立性；
2. 性能提升：降低不必要的性能浪费，提升页面性能。

## 3.4 针对3.3中的技术方案，是否还有别的替代方案同样能完成发明目的？

/\* 替代方案可以是完整技术方案的替代，也可以是部分结构或者步骤的替代。

（1）HTML5的Canvas API可以实现更多复杂的异性动画，但是针对不是非常复杂的动画，本方案足矣，并且不会造成性能浪费，简单动画使用Canvas可能会造成不必要的性能浪费。

## 3.5交底书中技术术语的名词解释

/\* 记载交底书中出现的专业技术术语、缩写、外文的解释。

CSS3：前端动画规范；

steps：CSS3提供的阶跃函数，实现对每个动画关键帧周期的控制；

cubic-bezier：贝塞尔曲线函数，CSS3提供的实现非线性动画的方法；

transform：CSS3中提供的进行2d、3d变形、缩放、位移和旋转等的API；

translate3d()：CSS3提供的位移API，相比于position、margin等方式，能够降低GPU性能损耗；

预加载：前端将资源文件进行预先加载的一种方式；

脱离文档流：前端页面中，脱离文档流的元素的重绘不会赢影响到其他元素；

Canvas：HTML5规范中的画布API，能够进行丰富的绘图操作。

setTimeout：Javascript原生提供的定时器；