

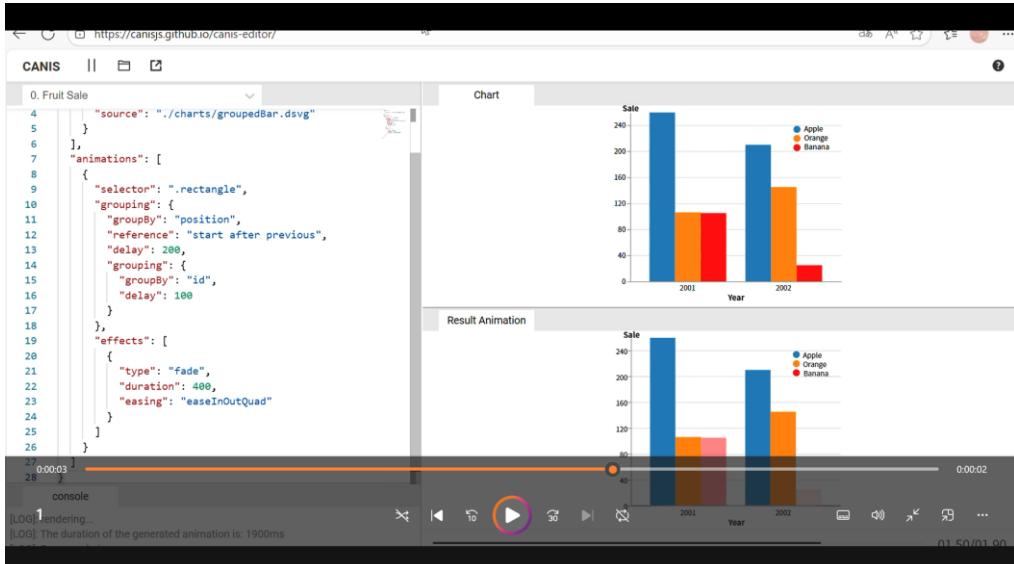
山东大学计算机科学与技术学院

大数据分析与实践课程实验报告

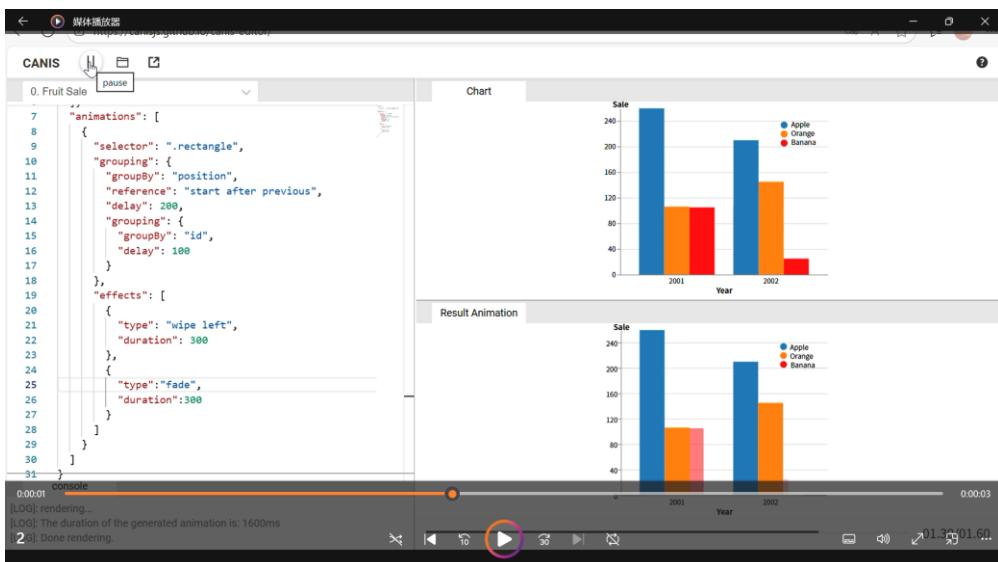
学号: 202320130203	姓名: 李姿含	班级: 数据 23
实验题目: Canis 实践		
实验学时: 2		实验日期: 20251112
实验目标:		
掌握 Canis 这一面向数据驱动图表动画的高级领域特定语言的核心语法与使用逻辑；学会利用 Canis 编辑器加载数据增强的 dSVG 图表文件，通过声明式规范定义图表元素的分组规则、动画效果及时间控制参数；成功实现多种自定义图表动画效果并验证其跨平台渲染可行性；深入理解 Canis 编译器的工作机制，对比不同动画参数对最终效果的影响。		
作品描述（实验背景、数据集来源、描述思路、实验背景）：		
1、实验背景： 在数据可视化领域，动态图表能够更直观地展现数据变化趋势和内在关联，但传统动画开发往往需要编写复杂的底层代码，门槛较高且复用性差。Canis 作为 EuroVis 2020 提出的声明式图表动画语言，通过封装底层渲染逻辑，允许开发者基于数据属性快速定义动画效果，极大简化了数据驱动动画的开发流程。本实验基于这一工具，旨在实践从图表加载到动画定制的完整流程，探索不同动画参数组合的效果差异。		
2、数据集来源： 实验采用 Canis 官方提供的水果销售数据集（Fruit Sale），该数据集包含 2001 年和 2002 年苹果（Apple）、橙子（Orange）、香蕉（Banana）三种水果的销售数据，以分组柱状图（groupedBar.dsvg）格式存储，图表中每个矩形元素均包含数据属性标注，支持按数据维度进行动画分组控制。		
3、描述思路： 首先通过 Canis 在线编辑器（ https://canisjs.github.io/canis-editor/ ）加载官方提供的 groupedBar.dsvg 图表文件；随后基于 Canis 语法规规范，设计三组不同的动画方案，核心差异在于动画效果类型、持续时间及缓动函数的配置；每组方案均采用嵌套分组规则，先按元素位置（position）分组，后按元素 ID（id）分组，确保动画按顺序依次触发。 第一步，访问 Canis 在线编辑器，通过“加载图表”功能导入 groupedBar.dsvg 文件，确认图表成功渲染，显示 2001–2002 年三种水果的销售柱状图；第二步，编写 Canis 动画规范，定义选择器为“.rectangle”（选中所有柱状图元素），分组规则统一设置为“groupBy: position”（外层按位置分组）和“groupBy: id”（内层按 ID 分组），外层延迟 200ms，内层延迟 100ms，确保动画有序播放；第三步，分别配置三组动画效果：第一组采用“fade”（淡入）效果，持续时间 400ms，缓动函数为“easeInOutQuad”；第二组采用“wipe left”（左滑显现）与“fade”组合效果，每组效果持续时间为 300ms；第三组采用“wipe bottom”（下滑显现）效果，持续时间 300ms，缓动函数为“easeOutBounce”（弹跳缓动）；第四步，点击编辑器“渲染”按钮，生成动画并观察效果，记录控制台输出的动画总时长日志；第五步，对比三组动画的视觉效果和性能数据，分析不同动画类型、缓动函数对用户体验的影响。		

结果图片：

第一张结果图片对应第一组动画方案，采用“fade”淡入效果，动画总时长 1900ms。2001 年和 2002 年的柱状图元素按位置和 ID 顺序依次淡入，元素从透明逐渐变为不透明，缓动函数“easeInOutQuad”使淡入过程先慢后快再慢，过渡平滑。



第二张结果图片对应第二组动画方案，采用“wipe left + fade”组合效果，动画总时长1600ms。动画过程中，柱状图元素先从左侧逐渐展开(wipe left)，同时伴随淡入效果(fade)，两种效果叠加使元素显现过程更具层次感。



第三张结果图片对应第三组动画方案，采用“wipe bottom”弹跳效果，动画总时长1600ms。柱状图元素从底部向上展开，缓动函数“easeOutBounce”使元素展开至末端时产生轻微弹跳效果。

