

基于 HTML5 的算法动画可视化平台

戴松, 许冉, 周忠

(北京航空航天大学虚拟现实国家重点实验室, 北京 100191)



摘要: 基于新一代的 HTML5 标准, 设计实现了一个算法动画可视化平台, 并实际用于教学实践。平台包括算法动画开发库、可视化编辑器和辅助教学网站。算法动画开发库中绘图模块基于 HTML5 Canvas 和 JavaScript 提供主要动画元素和效果的类库, 动画生成模块进一步把这些类封装成绘制框架和函数, 在动画创建过程中存储和定时更新画布。设计了一种基于时间线的脚本序列语法定义、存储与动画缩略图生成方法, 在此基础上设计了支持离线工作的可视化编辑器。在这些工作基础上, 设计实现了辅助教学的“新手画呀”网站, 支持在线编辑动画、评论和分享等功能, 丰富了教学手段。在初步的教学实践中, 平台表现出良好的应用成效。

关键词: HTML5; 算法动画; 可视化编辑器; 脚本序列; 辅助教学

中图分类号: TP312 文献标识码: A 文章编号: 1004-731X (2013) 10-2436-08

HTML5-Based Algorithm Animation Online Platform

DAI Song, XU Ran, ZHOU Zhong

(State Key Laboratory of Virtual Reality Technology, Beihang University, Beijing 100191, China)

Abstract: A novel visualized algorithm animation online platform was developed on HTML 5 which included development library, online editor with website. The development library contained animation drawing and generation modules. The drawing module offered a library for animation elements and effects based on HTML5 Canvas and JavaScript, and the generation one further packaged it into framework and functions with the animation storage and update. A timeline-based scheme for script sequence definition, storage and snapshot generation was proposed. On the basis of work above, a teaching website was implemented, which provided an online algorithm-animation editor, comments and sharing. This platform has been applied in several course teaching.

Key words: HTML5; Algorithm Animation; Visualized Editor; Script Sequence; teaching assist

引言

远程教学可以将教学资源传播到教育资源不发达的国家或地区, 是当前计算机技术与应用的热点之一。算法可视化是从 80 年代开始发展的一种算法辅助教学技术, 它可以用可视化的形式动态展示算法的执行过程, 使学生更容易理解算法思想, 其关键在于如何将图形的动态变化

与算法的动态执行过程有效关联。算法可视化辅助教学引起了很多研究者的关注, 特别是随着互联网的飞速发展, 很多工作开始研究算法可视化的远程教学系统, 并取得了进展。

本文基于最新的 HTML5 标准, 设计实现了算法动画开发库以及可视化编辑器, 设计了一种基于时间线的脚本序列语法定义、存储与动画缩略图生成方法, 在此基础上, 研发了“新手画呀”算法教学辅助网站, 介绍了教学实践情况, 最后进行了总结。

1 研究现状

在算法教学中, 目前常用的算法教学方法是“幻灯片+

收稿日期: 2013-05-26 收稿日期: 2013-07-22
基金项目: 国家“863”高技术研究发展计划(2012AA011801); 北京航空航天大学研究生教育与发展研究基金
作者简介: 戴松(1991-), 男, 湖北, 硕士生, 研究方向为计算机视觉与可视化教学; 许冉(1990-), 女, 安徽, 硕士生, 研究方向为计算机视觉与可视化教学; 周忠(1978-), 男, 江苏, 硕导, 研究方向为虚拟现实、分布式系统等。

教材”的形式讲解算法的原理和实现过程,并在课后安排相应的算法编程实验。学生看到的多是算法的静态信息,在理解抽象概念和算法思想方面仍然存在困难。也有一些教师在幻灯片中加入 Flash 动画演示并配合一定的文字说明来讲解算法的动态执行过程,在表现力上有所提高,但在编辑已有动画方面存在一定的技术门槛,且不利于过程的探讨交流。另外,算法的编程实验还存在对编程语言和工具的要求,只能降低算法练习量,影响了学生对算法本身的实验效果。

算法动画(Algorithm Animation)为算法学习和教学带来了新的思路,它利用连续的动态图形形式来解释算法的执行过程,有助于学生理解和掌握算法思想^[1]。最早 1981 年就出现了一个可视化的排序算法教学短片“Sorting Out Sorting(SOS)”^[2],如图 1 所示。

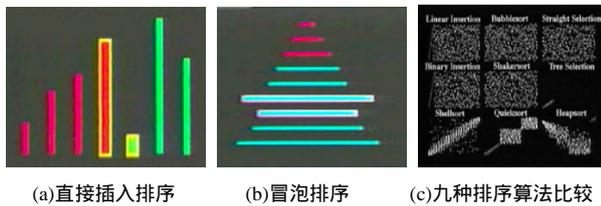


图 1 SOS 的排序算法动画过程

在 80 年代出现了真正意义上的算法动画软件,比较著名的是 Brown M.H.和 Sedgewick R.开发的 3 个算法动画系统: BALS A^[3,4]、BALS A-II^[5]和 Zeus^[6],并在布朗大学实际使用这些系统辅助计算机课程的教学, BALS A^[3]提出在算法状态改变时增加感兴趣事件进行标识, BALS A-II^[5]在 BALS A 基础上增加了算法动画演示的分步播放控制, Zeus^[6]在前两个系统的基础上增加了多视图编辑的概念。Brown 等后来又在 BALS A 系列的基础上开发了在电子书内嵌入算法动画的 CAT^[7]和用 Java 编写的 JCAT^[8],使用 JCAT 的算法教学如图 2 所示。

在 90 年代, Stasko J.T.等设计实现了 Tango^[9]、Polka

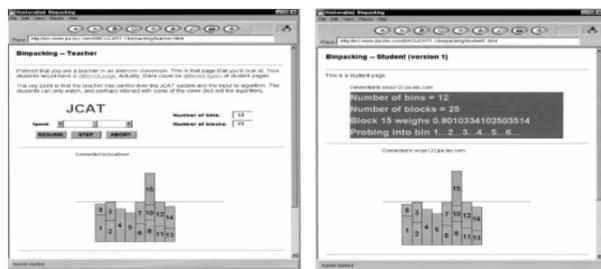


图 2 JCAT 系统算法教学界面

等系统,实现了可沿路径平滑运动的动画效果, Polka^[10,11]在 Tango 的基础上增加了一些动画命令,使开发者更容易创建动画。Polka 有一个用于前端交互的动画命令解释器 Samba^[12],包含了一些参数化的动画命令,这些命令可以生成不同的动画效果,如图 3 所示。

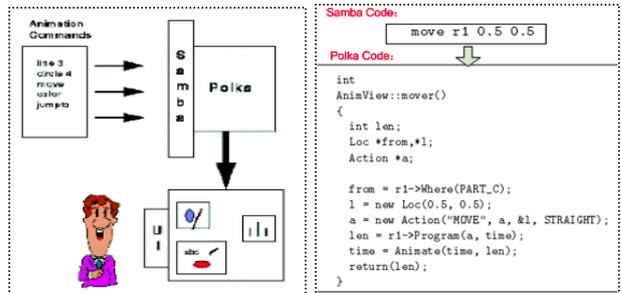


图 3 Polka 系统算法教学界面

随着互联网的快速发展,越来越多的算法动画软件基于 Web 进行开发,例如 Jeliot、JAWAA、ANIMAL、TRAKLA2 等。Jeliot 使用自动生成的动画技术使用户便于创建算法动画,通过使用 Polka 包实现可视化绘图。JAWAA 和 ANIMAL 系统通过让学生实践加深对算法思想的理解,学生可以使用算法动画系统创建算法动画,分享学习成果。JAWAA 支持学生创建自己的算法动画,并在浏览器中形成以动画命令序列为顺序的分步动画演示。ANIMAL 提供给学生创建动画的 ANIMAL API,来实现创建动画的效果。TRAKLA2 比较注重学生的实践能力,其教学网站提供了很多学习资料和算法编程练习题,如图 4 所示。

从国内外相关工作可以看出算法动画发展的两个趋势:一是算法动画应用向 Web 方向发展;二是算法动画系统由传统的只能观看动画来理解算法原理向可交互编辑进

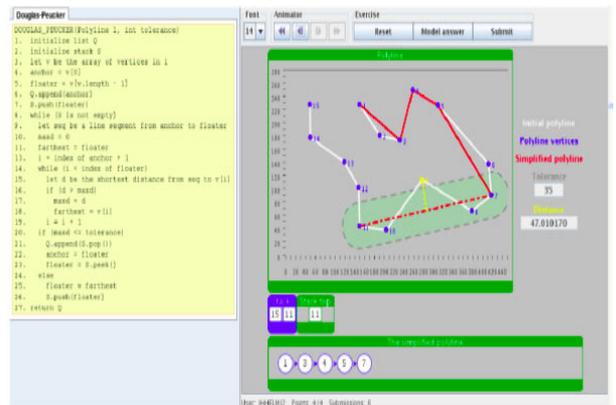


图 4 TRAKLA2 系统算法练习网

行实验方向发展。现有的算法动画系统还存在着一些问题：1. 需要安装插件或程序使用，不利于普及；2. 制作基础要求高，实现过程相对复杂，需要花费较多时间；3. 可视化编辑功能较弱，特别是已完成的算法动画不容易传播和再次编辑修改，降低了算法动画的可重用性。

HTML5 是新一代互联网技术标准，兼容 HTML 5 的浏览器不需要插件即可支持图形、动画、视频等功能。论文基于 HTML5 的新特性并结合 JavaScript 设计实现了算法动画开发库及可视化编辑器，可以更简单、快速的创建算法动画，通过教学网站形式进一步增强了使用算法动画进行算法教学的可用性。

2 系统结构

我们设计的算法动画可视化平台主要包括算法动画开发库、可视化编辑器和教学辅助网站这三个部分。算法动画开发库封装了常用元素及其效果的绘制方法，设计了所有动画元素的存储管理，提供常用元素的接口。我们设计了一种基于时间线脚本序列的动画编辑方法，并在此基础上设计实现了可视化编辑器。最后设计实现了算法动画教学辅助网站。系统结构设计如图 5 所示。

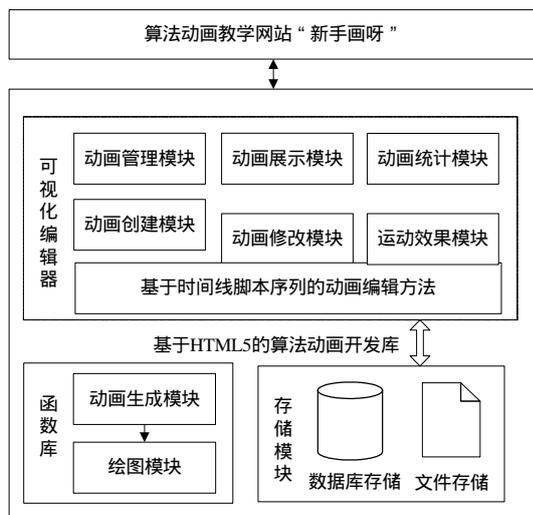


图 5 系统结构设计图

3 算法动画开发库的设计与实现

算法动画开发库是一组复用性高的函数（类）集合，包含函数库和存储模块，为可视化编辑器提供底层的绘制和存储方式，其中函数库包括绘图模块和动画生成模块。

3.1 绘图模块

算法动画由多个动画元素和动画效果组成，动画效果

是作用在动画元素上的特定效果，通过改变动画元素的属性实现，其实现方法是每个动画元素对应相对独立可复用的绘图函数（类），函数包含单个动画元素的绘制，调用绘图函数将其绘制在画布上，通过绘图函数接口获得动画元素的属性信息，改变动画元素的属性来实现动画效果，如图 6 所示。

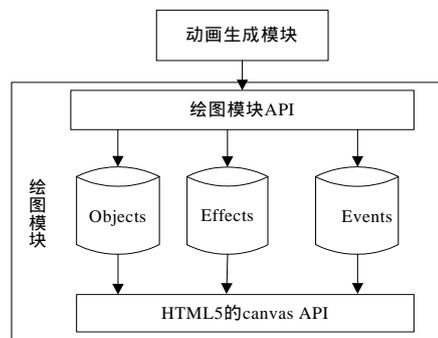


图 6 绘图模块的设计与封装

绘图模块把算法动画中常用元素的动画效果封装成有意义的 API 接口。在图 6 中，动画元素“Objects”包含矩形、圆形、圆环、文本、链表结点等常见动画元素，此外还加入了图片、声音多媒体元素以丰富算法动画的创建。动画效果“Effects”包含选择、移动、运动、高亮、连接、删除等常用动画效果，绘图模块 API 包含封装的主要绘图类的使用方法。“Events”负责处理动画元素和效果的监听事件。

绘图模块基于 HTML5 canvas 元素的绘图 API 实现动画效果。由于 canvas 元素的编程接口是面向底层的图形绘制，并没有封装好可以使用的图形绘图类，需要对每个设计好的图形编写绘图类进行封装。我们在完成绘图类的设计后，提取出模板文件，以方便未来绘图类的扩展。绘图模块主要由 AnimatedObject、rectangle、circle 等等实现，如图 7 所示。

3.2 动画生成模块

动画生成模块的主要功能是利用动画元素属性的变化调用绘图模块 API 实现绘图效果，并通过定时重绘生成动画。动画生成模块完成绘图代码的解释部分，具体的执行绘图部分由绘图模块完成。

3.2.1 绘图代码的解释执行

绘图代码由绘图函数及参数组成，在其执行过程中，需把绘图代码转化成相应动画命令，才能通过这些命令调用绘图模块的函数，完成动画创建，如图 8 所示。

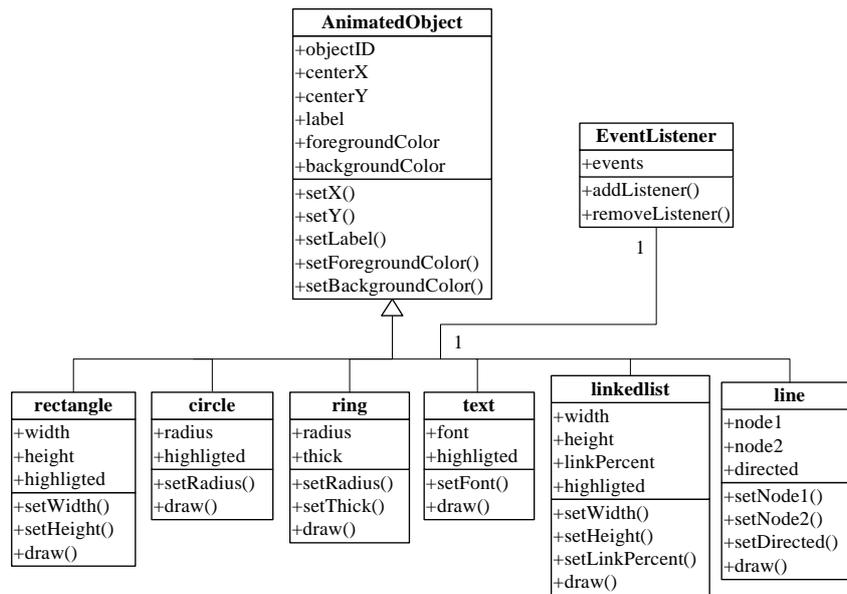


图 7 绘图模块的主要类图

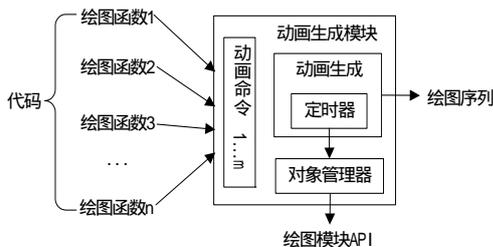


图 8 绘图代码的解释执行

3.2.2 动画元素的绘制

用户编写绘图函数语句时, 根据函数的参数执行函数体部分, 由动画生成模块解释调用的动画命令为具体的绘图模块中的动画效果, 这些动画效果序列在画布中按照顺序完成绘制。绘图序列是绘图模块封装的动画元素及其操作的具体带参数的实例绘制, 所有在网页画布上的动画效果均由绘图序列完成。动画生成模块把创建动画过程中所有的动画命令按照顺序存储在一个固定的数组中, 通过执行数组中的动画命令, 调用相应的绘图模块 API, 完成动画元素的绘制, 动画元素绘制流程如图 9 所示。

动画生成模块使程序无需关心动画元素对象的重绘, 只需要通过绘图调整动画元素对象的属性即可, 对象管理器负责管理所有的动画元素对象的绘制, 动画生成模块主要包括 objManager 类和 animManager 类, 如图 10 所示。

动画生成模块可以实现解释型绘图函数代码的执行, 同步创建算法动画, 如图 11 所示。

4 可视化编辑器的设计与实现

可视化编辑器调用函数库和存储模块完成底层的绘制和存储。为了实现可视化创建算法动画, 设计了一种基于时间脚本序列的动画编辑方法, 在此基础上实现了动画创建、动画修改、运动效果、动画管理、动画展示及动画统计等功能模块。编辑器主要包括两部分内容: 编辑算法

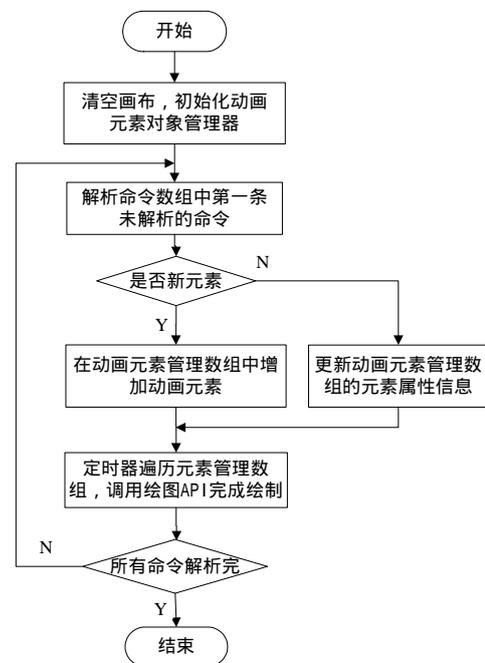


图 9 动画元素绘制流程图

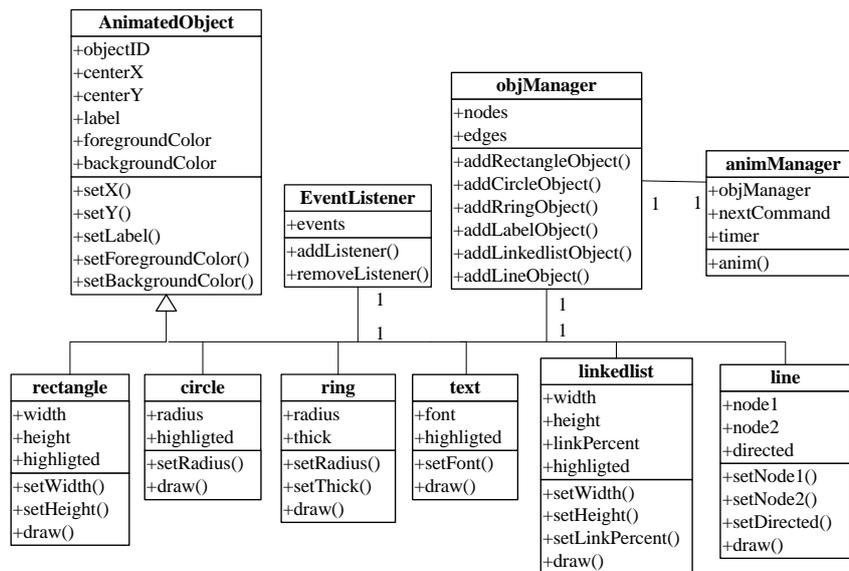


图 10 动画生成模块的主要类图

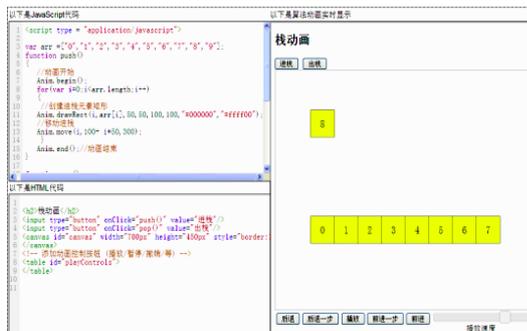


图 11 绘图函数同步生成动画

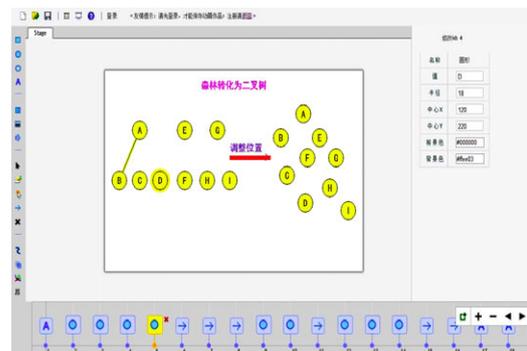


图 13 可视化编辑器界面

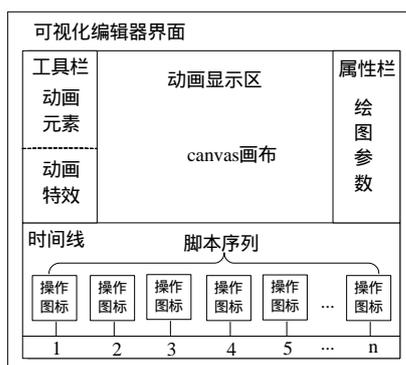


图 12 可视化编辑器界面区域组成

动画和记录算法动画生成过程,其界面区域设计如图 12 所示,最终的编辑器界面如图 13 所示。

4.1 时间线脚本序列的编辑方法

基于开源的时间线库 Timeline 的 API 接口可以把绘图

参数等内容以图形化的方式显示在时间线上,以时间线脚本序列的方式生成动画。

4.1.1 脚本序列的语法定义

动画命令的脚本命令由绘图函数和参数按照一定的语法规则生成,其式如下:

脚本命令 = 绘图函数名称+<;>+绘图元素 id+<;>+大小参数+<;>+位置坐标+<;>+前后颜色参数。

如绘制一个矩形, id 号为 0,显示值是“矩形”,长宽均是 50,中心坐标是 (100,100),前景色是黑色,背景色是蓝色,那么其脚本序列为:“CreateRectangle<;>0<;>矩形<;>50<;>50<;>100<;>100<;>#000000<;>#90c8ff”。

4.1.2 脚本序列的可视化方法

时间线上的可视化事件内容由两部分组成,第一部分是可视化事件的开始和结束时间,第二部分是可视化事件

的内容, 即文中的脚本序列和图标显示信息。脚本序列可视化的具体实现方法是在可视化事件生成时封装好每步的动画步内容, 包括脚本内容和图标标识信息, 在执行脚本序列生成动画的相关操作中都是需要只提取出脚本内容的, 可视化部分的具体实现类如图 14 所示。

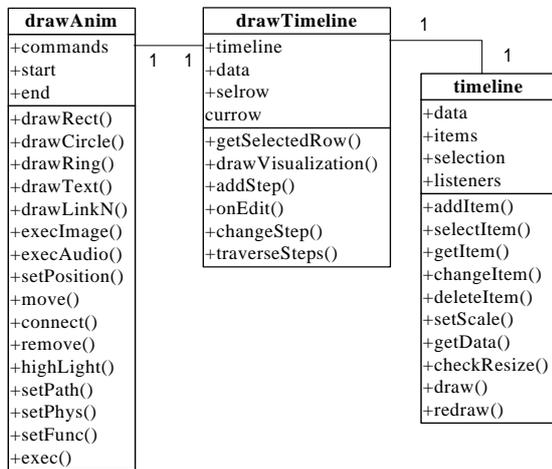


图 14 脚本序列的可视化类图

drawTimeline 类中的 addStep 函数功能是实现添加脚本序列内容及其图标标识信息, 作为一个事件项的内容可视化显示, 此部分内容的获取是通过 drawAnim 类中的命令数组 commands, 标识每步动画的脚本命令和图标信息, 具体对应关系如表 1 所示。

表 1 动画操作与可视化显示的关系表

操作类别	动画操作	动画脚本命令	可视化显示
动画元素类	创建矩形	CreateRectangle<>0<><>50<>50<>100<>100<>#000000<>>#90c8ff	
动画效果类	移动元素	Move<>0<>100<>100<>200<>200	

4.2 动画创建模块

动画创建模块的功能主要有实时绘制与脚本添加, 其中实时绘制是调用了开发库的绘制接口, 脚本添加是根据操作类型及其参数生成动画操作对应的脚本命令, 按顺序添加到动画脚本序列中, 并在时间线上可视化显示。例如, 当点击绘制一个矩形时, 会调用封装好的 CreateRectangle() 函数并获取其长、宽、背景色等属性参数, 并生产对应的脚本序列, 最后以图标的方式将其添加至时间线。

4.3 运动效果模块

动画的运动效果通过改变动画元素的位置实现, 该模

块包括编辑路径运动、物理特性运动和函数运动。

编辑路径运动是指在选择移动操作后, 把动画元素从一个位置移动到另一个位置的操作。默认路径是一条直线, 也可让动画元素沿任意路径运动, 则需通过编辑原来路径, 通过拖拽路径上的控制节点, 改变路径的形状, 动画元素沿着编辑好的路径运动, 如图 15 所示。



图 15 运动效果的路径编辑

在常见的算法动画中并没有动画元素的包含物理特性的运动, 我们给动画元素增加了一些物理特性, 可以创建出更符合算法的生动动画效果, 包括速度、重力、弹力和碰撞检测等。实现效果如图 16 所示。具体实现方法如下:

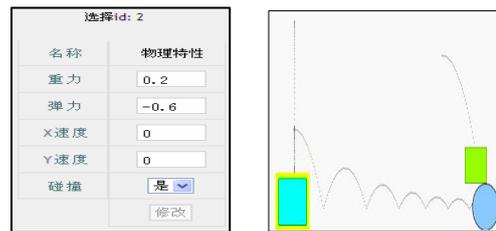


图 16 物理特性运动效果



图 17 运动效果的函数编辑

编辑路径运动是提供动画元素沿用户设置路径运动的效果, 但有时需要让动画元素沿着标准的函数运动, 添加的函数运动的动画效果使用 JavaScript 脚本语言中的数学函数 (Math 对象), 根据动画元素的 x 坐标, 计算得到元素的 y 坐标, 然后使用定时器的定时重画功能及时更新动画元素的位置。结合实际的运行效果, 使用的 Math 对象。

在函数运动中也增加了碰撞检测效果，具体实现与物理特性的碰撞检测类似，发生碰撞时加入了声音效果，界面如图 17 所示。

4.4 动画储存模块

动画内容存储在数据表中和服务端文件夹下，用户对动画内容的管理操作使用管理界面实现，用户对动画内容的管理包括登录、新建保存动画、打开修改动画、删除动画等，该模块的实现类如图 18 所示。

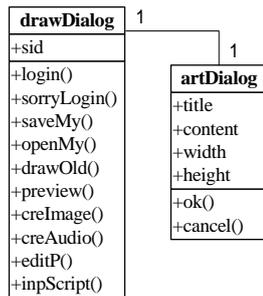


图 18 管理模块的主要类图

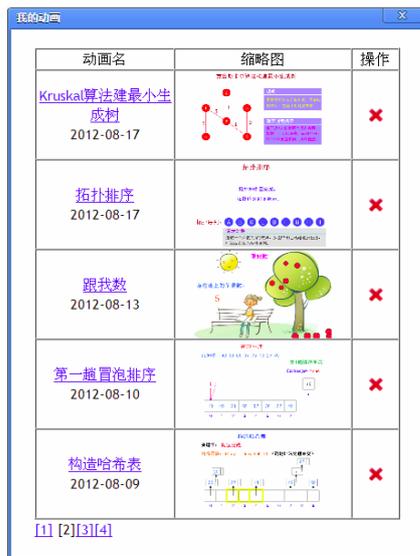


图 19 动画作品储存界面

动画储存界面如图 19 所示。

4.5 动画缩略图的生成

在储存算法动画序列的同时，动画缩略图也被保存至服务器。缩略图通过 canvas 支持的 toDataURL 函数导出 base64 编码图片保存为文件，在显示缩略图时将图像输出为 base64 压缩的字符串，并通过 php 读写文件将其保存为 jpg 格式图片。其中缩略图的编码过程如下：

```

//生成 image data
var canvas = document.getElementById("canvasPanel");
if (canvas.getContext) {
    var ctx = canvas.getContext("2d");
    //将图像输出为 base64 压缩的字符串 默认为
    image/png var data = canvas.toDataURL("image/png");
    document.getElementById("thumbnail").value = data;
}
    
```

动画缩略图的 GIF 生成是提取多帧图片并进行合成，首先是通过 javascript 将动画的脚本序列转化成多帧静态图片，具体实现方法是将获取的每帧动画的脚本序列投射至 canvas 画布中进行绘制，绘制完毕后用上文介绍的方法将此帧的图片保存，从而获取一个动画的每帧图片。然后调用开源的 PHP 函数包 GIFEncoder 将多帧静态图片合成一个 GIF 图源，最后通过读写文件来创建一个.gif 后缀的 Gif 图片。

4.6 编辑器离线操作的设计与实现

可视化编辑器的离线操作包括在离线状态下创建动画和储存动画，是基于 HTML5 的 manifest 属性和 webstorage 属性实现，其流程如下：

if(在线)

根据 manifest 文件下载离线缓存至本地

else { //离线

直接访问本地的离线缓存

用离线编辑页面 offline.html 替代在线编辑页面 new.php }

//本地储存

创建本地数据库 webstorage

while(制作动画) {

获取所创建的动画数据并提交至本地数据库

//在线提交

if(在线状态) {

查看本地数据库并获取离线动画数据

在线提交至服务器数据库 }

}

END

5 平台的教学实践分析

我们在以上工作的基础上，设计实现了一个算法动画的辅助教学网站 (http://aa.vrlab.buaa.edu.cn)，包括动画管理、动画展示和用户管理三个板块，具体功能如图 20 所示。教学网站首页根据访问量、评价等对作品进行排序，并

提供优秀作品或活跃用户的缩略图动画表示, 如图 21 所示。

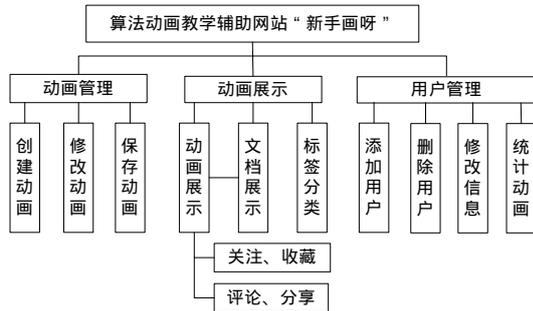


图 20 教学辅助网站结构设计

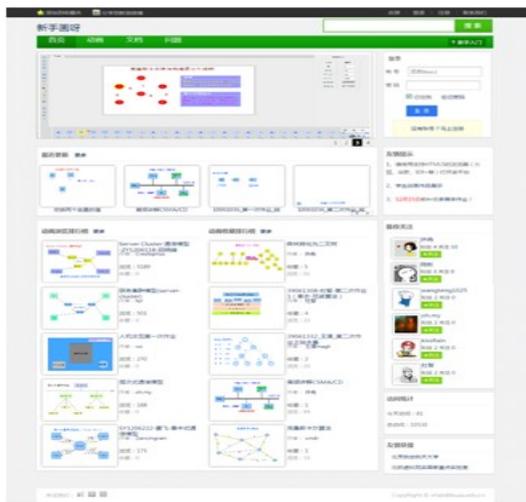


图 21 算法动画教学网站首页

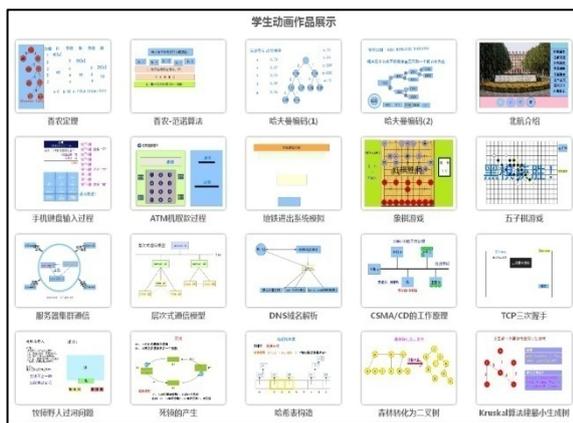


图 22 学生优秀作品展示

有注册用户 134 人, 提交动画作品 317 份, 文档作品 37 份, 浏览量 10530 次。学生可以通过创建算法动画的方式表达自己对于算法的理解, 也可以通过观看别人的动画作品进一步加深对算法的理解, 还可以与同学进行交流、

分享等互动。部分算法动画作品如图 22 所示。

为了改进“新手画呀”教学应用的实际教学效果, 在课程结束前对使用教学应用的学生进行了在线问卷调查统计, 共有 58 名同学完成了该调查问卷, 其中男生 50 人, 女生 8 人。在学生对动画创建平台在其学习课程的帮助统计中, 45% 的同学认为有些帮助, 36% 同学认为帮助较大, 12% 的同学认为帮助很大。学生反馈情况统计如图 23 所示。



(a) 学生访问频率 (b) 学生停留时间 (c) 对学习帮助情况

图 23 调查问卷反馈

5 结论与展望

本文使用 HTML5 的新特性并结合 JavaScript 技术研发算法动画开发库、动画可视化编辑器和教学辅助网站, 增强了使用算法动画进行算法教学的切实可用性。与传统的算法动画软件相比, 如表 2 所示, 本文设计的在线的算法动画可视化编辑器具有无需安装插件、可视化效果好、编辑要求低等优点, 为算法教学及学习提供了一种新的途径。

表 2 “新手画呀”与同类系统的功能比较

	是否安装插件	可视化效果	开发基础	开发过程难度
“新手画呀”	否	强	低	易
Jeliot	是	弱	高	易
Flash	是	强	高	难
JCAT	是	弱	高	易
TRAKLA	是	强	高	易

但该算法动画平台目前还存在一些问题, 根据调查统计, 学生的意见主要集中在以下三个方面: 一是完善动画创建平台, 丰富 UI 界面、增加创建元素、音频编辑等功能; 二是完善动画展示页面, 改进对动画作品的打分、评价、分享方式等; 三是改善用户体验, 如在删除作品时增加提示框, 提交作品后给予及时反馈等。我们进一步将完善平台, 并结合心理学进行交互设计和学习效果分析和研究。

参考文献:

- [1] 刘亚琴. 基于三层结构模型的算法可视化教学研究与实践[D]. 长沙: 湖南师范大学, 2007.
- [2] Baecker R. Sorting out sorting: A case study of software visualization for teaching computer science [C]// Software Visualization: Programming as A Multimedia Experience, 1998: 369-381.

(下转第 2448 页)

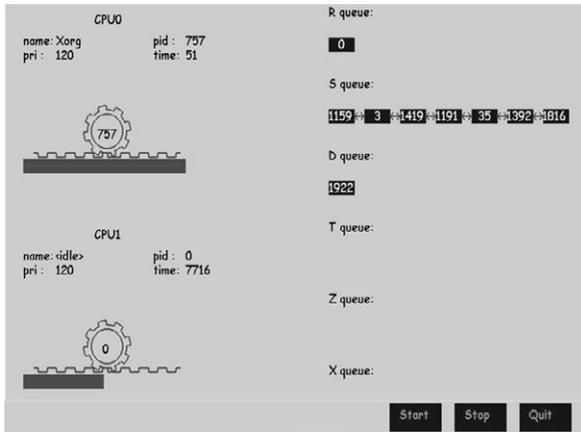


图 8 进程运行过程可视化主界面

4 结论

通过可视化技术演示操作系统中进程的调度及状态转换,是用图学的技术解释程序的运行过程。这种方式有如下特征:(1)直观,没有复杂的逻辑,没有分支,没有循环,所有的操作都是自然顺序地执行,体现了系统处理的流程。(2)生动形象,用图形表示系统中的对象,动画表示系统中的过程。(3)可以从不同粒度上展示系统的处理过程,可以选择采集关键事件,忽略繁杂的不重要的事件,演示系统处理的主要流程。当然,为演示系统的某个处理过程,事

(上接第 2443 页)

[3] Brown M H, Sedgewick R. A system for algorithm animation [J]. ACM SIGGRAPH Computer Graphics (S0097-8930), 1984, 18(3): 177-186.

[4] Brown M H, Sedgewick R. Techniques for algorithm animation [J]. IEEE Software, 1985, 2(1): 28-39.

[5] Brown M H. Exploring algorithms using Balsa-II [J]. Computer (S0018-9162), 1988, 21(5): 14-36.

[6] Brown M H. Zeus: A System for Algorithm Animation and Multi-view Editing [C]// Visual Languages, 1991: 4-9.

[7] Brown M H, Najork M A. Collaborative active textbooks: A web-based algorithm animation system for an electronic classroom [C]// Proceedings of the 1996 IEEE International Symposium on Visual Languages. USA: IEEE, 1996: 266-275.

[8] Brown M H, Raisamo R. JCAT: Collaborative active textbooks

先要做充分的准备。

参考文献:

[1] Torvalds Linus, David Diamond. Just for Fun: The Story of an Accidental Revolutionary [R]. Mason, OH, USA: Harper-Collins Business, 2001.

[2] Jaidev P Patwardhan, Alvin R Lebeck, Daniel J Sorin, Communication breakdown: Analyzing cpu usage in commercial web workloads [C]// Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software, Austin, Texas, USA, March 10-12, 2004. USA: IEEE, 2004: 12-19.

[3] Daniel P Bovet, Marco Cesati. 深入理解 Linux 内核. [M]. 第 2 版. 陈莉君 等译. 北京: 中国电力出版社, 2011: 51-70.

[4] EfficiOS Inc, Linux Trace Toolkit Open-source Project. [EB/OL]. (2004-12-14) [2013-6-20]. <http://www.opensys.com/LTT/>

[5] 陈丽波, 吴庆波. Linux 内核跟踪机制 LTT 的研究[J]. 计算机工程, 2005, 31(12): 61-63.

[6] Danlel A Keim. Information Visualization and Visual Data Mining [J]. IEEE Transaction, Visualization and Computer Graphics, 2002, 7(1): 1-8.

[7] K A Hawick. Interactive Graph Algorithm. Visualization and the GraViz Prototype [R]// Technical report, Jansz K, et al. USA: AT & T Bell Laboratories, 2010.

[8] 王生辉. 使用 ftrace 调试 Linux 内核[R/OL]. (2010-6-10) [2013-6-20]. <http://www.ibm.com/developerworks/cn/linux/l-cn-ftrace1/>

using Java [J]. Computer Networks and ISDN Systems (S0169-7552), 1997, 29(14): 1577-1586.

[9] Stasko J T. Tango: A framework and system for algorithm animation [J]. IEEE Computer Society (S1063-6919), 1990, 23(9): 27-39.

[10] Stasko J T, Kraemer E T. A methodology for building application-specific visualizations of parallel programs [J]. Journal of Parallel and Distributed Computing (S0743-7315), 1993, 18(2): 258-264.

[11] Stasko J T. Using student-built algorithm animations as learning aids [C]// SIGCSE '97: Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education. 1997, 29(1): 25-29

[12] Stasko J T. SAMBA Animation Designer's Package [C]// Graphics Visualization and Usability Center Georgia Institute of Technology, Atlanta, USA, 1995, 27(6): 1065-1073.