

基于HTML5和JavaScript轻量型动画框架开发

平淑文,潘珏羽,张学金,杜晓荣

(中山大学 电力电子与控制技术研究所,广东 珠海 519082)

摘要:互联网作为一个新兴的媒介,最突出的优势就是图片和动画展示,利用动画引擎可以很方便地实现网页动画显示和制作网页游戏。首先简要介绍了HTML5和动画引擎的概念,及相较于FLASH等现有技术,使用HTML5制作动画的优势所在。然后详细介绍了基于HTML5和JavaScript开发的轻量型动画引擎,从结构和算法方面介绍了滤镜、物理效果、路径设置等具体方法,提出了表单式编程概念。最后简要介绍了利用该引擎制作的动画实例,分别展示了动画滤镜效果和路径动画效果。

关键词:HTML5;轻量型;动画;引擎

中图分类号:TP311.52

文献标识码:A

文章编号:1673-629X(2013)12-0005-06

doi:10.3969/j.issn.1673-629X.2013.12.002

Development of a Lightweight Animation Engine Based on HTML5 and JavaScript

PING Shu-wen, PAN Jue-yu, ZHANG Xue-jin, DU Xiao-rong

(Institute of Power Electronics & Control Technology, Sun Yat-Sen University, Zhuhai 519082, China)

Abstract: As a new medium, the most important advantage of the Internet is the display of pictures and animations. It is very easy to make an animation or webpage with an animation engine. Firstly, introduce the basic concepts of HTML5 and animation engine, and advantages of making animations with HTML5, compared with the existing technologies like FLASH. Then introduce the lightweight animation engine based on HTML5 and JavaScript in details, describe the structure and algorithms of filter, physical effects, path setting and so on, and propose the concept of form programming. At the end, there are some examples made with this engine, and it shows the effects of animation filter and path animation.

Key words: HTML5; lightweight; animation; engine

0 引言

互联网作为一个新兴的媒介,日渐成为网络游戏、电子商务等大展拳脚的平台。相比传统媒介,它最突出的优势就是图片展示和动画展示。试想,当打开一款优秀的网络游戏时,最先被吸引到的一定是精美的人设和场景;而当打开淘宝网的主页时,一定首先被搜索框下面的大幅滚动图片广告吸引。所以,作为众多商家和企业宣传自身的重要平台,互联网在如今的电子商务和电子产品中,显得非常重要,而铺天盖地的动态图片展示自然也成为网站盈利的重要组成部分,如何快速地制作精美的动画,从而吸引人们的注意,渐渐成为网页设计者们所关注的问题。

1 动画框架

1.1 什么是动画框架

游戏的核心在于动画,没有动画,就只能玩一些猜字和数独等小游戏;而广告的本质也是动画。因此,动画编程在游戏编程和广告设计中是非常重要的。

动画其实只是一系列快速显示的、只有微小差别的图像,由于人眼的敏感性较低,所以会认为这些变化就是移动。人眼能够感觉到的动画是至少每秒12帧,即每秒更换12幅图片,但为了良好的动画效果,一般的动画采用的是24帧每秒。在动画编程的同时,需要考虑的是处理器和系统的负担,因此为了平衡动画的帧数和系统的负担,会根据不同的需要设置合适的更

收稿日期:2013-03-02

修回日期:2013-06-08

网络出版时间:2013-09-29

基金项目:广东省高等学校大学生创新实验项目(201002017)

作者简介:平淑文(1991-),女,硕士研究生,研究方向为计算机图形图像技术、移动互联网开发;杜晓荣,通讯作者,教授,研究方向为软件开发方法及支撑环境、图形图像技术及应用等。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20130929.1523.029.html>

新时间间隔。2D 帧动画是最基本的动画技术,它主要是通过显示一系列预先生成的、静态的帧图像来实现动画。

虽然目前已经有各种各样的技术可以帮助人们制作一个完整,甚至精美的动画,但是还是希望可以找到一个方法,在保证质量的同时,追求效率。动画框架就可以满足这项需求。动画框架一般包括基本的绘制和更新函数,只需要利用框架的函数,就可以很简洁、方便地实现动画功能。

利用动画框架制作动画,可修改性强,可以很方便地通过动画的某些特征检索并修改、删减该动画;可扩展性也很强,利用已有基础动画类型,修改部分参数就可以创建一个新的动画类型;代码复用率高,使得编程变得更高效^[1-3]。

1.2 现有技术

目前实现网页动画的途径主要有 Adobe Flash、第三方插件和 HTML5,三者各有特点。Flash 是较为传统的技术,使用面积最广,目前市面上超过 75% 的网页游戏及动画都基于该技术。2012 年 Flash11 推出后,能良好地实现 3D 画面渲染,较过去有很大改善。使用第三方插件运行的网页动画也占有着一定比例,插件扩展了浏览器的功能,但同时带来了插件安装的麻烦,以及潜在的安全隐患。

HTML5 是实现网页动画的一条新途径。Canvas 元素、WebSocket 技术以及 WebGL 标准的出现,为网页提供了新的编程接口,使网页游戏开发更加便捷。与其他技术相比,HTML5 有两点主要优势,首先 HTML5 标准及其相关技术均免费开放,规范并且易于推广;另外 HTML5 标准免去了安装插件的必要,减少了用户的麻烦,提高了安全性^[4-5]。

2 基于 HTML5 和 JavaScript 的轻量型动画框架

对于一款游戏框架而言,动画框架是其最基本的绘制显示子部分,而对于广告制作,只需要最基本的绘制和滤镜等功能即可,不要诸如碰撞检测、人机交互等功能。所以,从根本上讲,动画框架就是加入了输入输出和碰撞检测等功能的绘制框架。从细节上,基于 HTML5 和 JavaScript 的动画框架可以完成以下工作:

(1) 图形绘制:动态图形需要绘制在画布上,才能展现在观众和玩家面前;

(2) 路径设置:大部分动画不是固定在画布上某一个位置的,比如一条鱼,它往往会在大海中游来游去,不会静止在某一个固定的地方,所以设置移动路径和移动速度也是赋予动画灵魂的关键;

(3) 滤镜效果:有些时候,希望给动画添加一些 PS

效果,以增加一些真实感,比如将鱼的透明度进行设置,以模仿不同深度的鱼;

(4) 物理效果:为了提高真实感,希望动画中的角色遵循正确的物理原理,展现出和现实生活中一样的物理效果,所以物理效果的添加对于一个动画的真实感十分关键^[6]。

2.1 结 构

动画框架分为三部分:舞台类、对象类、工具类。

为了方便控制和操作,将所有的动画实例放在一个舞台上,只需要像个导演一样总的调控、控制舞台即可,这个舞台即由舞台类实例化得到,这里,舞台类仅包括一个类。它可以实现舞台初始化、绘制所有动画等功能。舞台类也是直接面向用户的接口层,用户只需要搞清楚这个类的基本用法就可以绘制最基本的动画。

对象类的实例就是舞台上的一个一个动画,所以,对象类的功能就是创建动画。它包含动画类、鱼实例类、太阳实例类三个类。可以实现图片分割、图片绘制、路径设置、添加滤镜、添加物理效果等基础功能。图片分割可以将一幅多帧图片分割为一帧一帧的图片,并将其存入数组,这个步骤一般在初始化时完成,所以绘制时无需再分割图片,可以大幅度提高绘制速度;路径设置即根据用户设置的参数:动画的初始位置以及速度,计算每帧图片显示的具体位置;滤镜函数可以为图片添加不同的滤镜效果,以增加真实感;添加物理效果可以使动画角色表现出诸如自由落体运动、匀速直线运动、反弹之类的物理效果,使得动画真实感更强;利用绘制函数,将已经准备就绪的图片绘制在画布上。鱼实例类、太阳实例类两个类继承自动画类,只是重载了动画的路径设置功能,用户可根据自身需求,扩展其他的动画实例类。

这里的工具类主要功能是为对象类提供具体的滤镜算法,有高斯模糊、色彩变换、透明度设置等多个类。它们之间不存在继承关系。

2.2 实 现

该动画框架的开发过程中,使用 HTML5 + JavaScript 语言,该框架的结构图如图 1 所示。

下面按照功能分类来介绍一下这个动画框架。

2.2.1 舞台类

舞台类只包含 stage 类,主要包括两个功能,初始化和绘制舞台。

(1) 初始化。

初始化函数—initialization,遍历整个传入舞台的图片数组,根据每一项中的 type 参数,创建相对应的对象实例,并将其放入舞台中。紧接着,完成初始化过程中最重要的一步,调用滤镜函数,分割多帧图片,将

每一帧存入数组中,方便接下来绘图使用。同时,如果该对象需要使用滤镜,也在这里完成,这样数组中保存的图片就是切割好,已经添加过滤镜的,绘图函数可以直接使用。

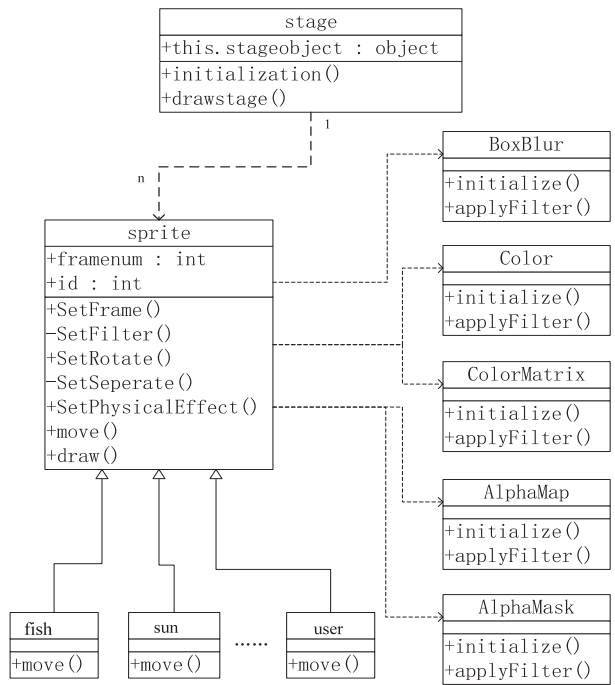


图 1 动画框架结构图

这里传入舞台的图片数据数组保存了整个动画中所有的数据,动画框架会根据这个数组自动创建相应对象实例,添加滤镜、设置关键帧、分割图片、添加物理效果、设置路径,最后完成绘制。用户在编程时,只需要根据数组中对象的名称,填入正确的数据即可,对于不熟悉 JS,甚至完全不懂 JS 的用户都非常容易实现。

(2)绘制舞台。
该函数会由用户在外部,每隔一段时间调用一次,它会清除整个画布,然后遍历整个图片数组,将其中的每一个对象的合适帧绘制在舞台上。究竟绘制第几帧,将由函数的入口参数决定。

2.2.2 对象类

对象类包括 sprite 类、fish 类、sun 类。其中 sprite 是基类,下面重点介绍它。

(1)切割。
为了保证接下来的绘图速度,首先需要切割图片。其实这里的原理很简单,就像是包装牛皮糖。制作好的牛皮糖都是一整张,很多很多个糖,而包装纸是一块糖一张的,师傅们会按照一块糖的尺寸把整张牛皮糖剪成一块一块的,一张糖纸放入一块。然后把包装好的糖放入大的包装袋中,这样就把整块的牛皮糖,变成了一袋独立包装的牛皮糖,当你想吃某一块或者把哪一块送给朋友时,你只需要对这一块进行操作就好。同样的道理,这里把整张的牛皮糖换成一幅多帧图片,

然后把包装纸换成一幅幅 canvas 画布,按照每帧图片的尺寸,将它们剪下,依次贴在画布上,然后把画有单帧图片的画布们放入数组,这样就得到一袋独立包装的“牛皮糖”了。当需要绘制某一帧图片时,找到对应的那块“牛皮糖”,对它进行操作就好了。

(2)滤镜。
很多时候,往往希望舞台上的动画各自呈现不同的滤镜效果,比如两条一样的鱼,希望小一点的那条鱼透明度高一点,而大一点的透明度低一点,这样看起来,那条小一点的就像是离得较远,大的那条离得近一样。所以相比简单地给整个画布添加某一个滤镜,给单个动画添加动画显得更明智一些,但是同样也麻烦一些。

这就相当于有人爱吃包着糖霜的牛皮糖,有人不喜欢。相比直接在牛皮糖制作时放入糖霜,让每一张牛皮糖都变得一模一样,师傅们只需要在卖给顾客时根据他们的要求,在把包装好的糖放入大包装时,将卖给要糖霜的顾客们的牛皮糖裹上糖霜就好。好在每一幅图片在初始化阶段的第一步已经变成了一袋独立包装的“牛皮糖”,所以只需要在将“小鱼牌牛皮糖”放入大包装袋时,让它里面的每一颗糖都变透明一点就好。

滤镜可以实现高斯模糊、色彩变换、透明度变换等功能,sprite 类会根据入口参数中 filterType 来决定添加哪项滤镜效果。

(3)路径设置。
为了使动画动起来,更有生气,更形象,需要为它们设置路径。这里有三种基本的移动方式,sprite 默认的运动方式是匀速直线运动。

根据入口参数的初始位置和终止位置,除以动画存在的时间,计算出两个方向上的平均速度。匀速直线运动可以满足基本的汽车、人物等大多数动画要求。

Fish 类重载了 move 函数,定义了另外一种路径设置方式。如果鱼在大海中直线前进,真实感很弱,所以为了体现鱼沉浮的状态,这里将鱼的路径设置为一个弧度较小的弧线。

以开始点和结束点间的距离作为圆弧半径,将开始点和结束点作为圆弧上两点,绘制一个 60°圆弧作为 fish 的轨迹。将 60°除以对象存在时间平均分为若干份,对象每秒滑过其中一份,据此利用几何原理计算出 fish 的运动速度。

Sun 类同样重载了 move 函数。这里 move 函数模拟太阳一天中东升西落,采用半圆路径。

以开始点和结束点间的距离的一半作为圆弧半径,将开始点和结束点作为圆弧上两点,绘制一个 180°圆弧—即半圆作为 sun 的轨迹。将 180°除以对象存在时间平均分为若干份,对象每秒滑过其中一份,据

此利用几何原理计算出 sun 的运动速度。

用户还可以自行扩展该框架,重载 move 函数,创建不同的路径设置函数。

(4) 物理效果。

现实生活中,物理效果无所不在,十分复杂。但是动画中并不要求把所有的物理效果均展现出来。比如需要制作一个模拟小球自由落体或者平抛运动的动画课件,只需要考虑重力因素就可以了,因为空气和小球的摩擦系数很小,摩擦力完全可以忽略不计。所以,该动画框架中封装了一些最常用的物理效果,并未包罗万象。

该动画框架中包含的物理效果有平抛运动、摩擦力、反弹等常用的基本原理。

①平抛运动是最常用的物理效果,调用该物理效果函数时,用户只需要给定水平速度和距离地面的高度两个参数即可。

根据物体距离底面高度 h , 可以计算出物体落地

$$\text{时间为: } t_{\text{total}} = \sqrt{\frac{2h}{g}}。$$

设时间 t_0 时,物体的位置是 (x_0, y_0) , 经过时间 Δt , 物体的水平位置很好计算,即 $\Delta t \cdot v + x_0$, 而垂直位置为: $h - \frac{g(\Delta t + t_0)^2}{2} = h - \frac{g}{2} \left(\Delta t + \sqrt{\frac{2(h - y_0)}{g}} \right)^2$, 根据舞台设置的刷新时间间隔,设置 Δt , 每次调用路径设置函数时,刷新物体的位置。

自由落体运动是平抛运动的一种特殊情况,当 $v = 0$ 时,平抛运动转换为自由落体运动,所以模拟自由落体效果时,不需要重新编写函数,调用平抛运动函数即可实现。

②虽然中学物理题中,常常会假定摩擦力忽略不计,但是现实生活中摩擦力很多时候并不能忽略不计,需要时时刻刻考虑摩擦力对物理效果的影响。不过好在模拟摩擦力并不麻烦,根据动摩擦力公式 $F = \mu T$, T 为物理作用在接触面上垂直于接触面的压力,而大多数情况物体都是水平放置的,所以 $F = \mu mg$ 。只需要在水平方向将物体的速度 v 添加一个消减因子, $-\mu gt$ (t 是作用时间) 即可。需要注意的是考虑摩擦力的情况下,摩擦力的方向随着运动方向变化,永远同运动方向反向。

③经常会看到类似小钢珠或者网球之类的球体自由落体掉在地面上反弹起来的动画,这类动画似乎因为反弹这一个小小的细节而变得非常生动,所以为了使得制作的动画更逼真、有灵动感,特别添加了反弹特效。反弹的原理很简单,物体打在另一个物体时,会在摩擦力的作用下,消减物体垂直于被作用物体表面方向的速度。如果被打物体是不规则形,那么作用力

方向和大小较难确定,而现实生活中也并不常常遇到,所以只考虑小球垂直于水平面、垂直于竖直面和两小球弹性碰撞三种情况。

调用该物理效果函数时,用户需要指定弹性消除因子的大小,这个因子会随着两个碰撞物体的材质不同而存在非常大的不同。比如把一个钢珠丢到一个木板上,钢珠一般会弹起来,但是如果把一个球状的糯米糍丢在同样的木板上,糯米糍是一定不会反弹起来的。这是因为糯米糍表面的摩擦因子非常大,导致它和木板碰撞时的弹性消除因子也非常大,它竖直方向速度瞬间被消减至 0。

假设某物体垂直与某水平板相撞,末速度为 v , 消减因子为 η , 则该物体被反弹起来的垂直向上的初速度为 $-v(1 - \eta)$ (参考系以竖直向下为正方向)。水平方向原理一样,只需要把 y 轴换做 x 轴即可。如果是两个小球弹性碰撞,则将 y 轴换做小球运动方向即可。如果两小球完全弹性碰撞,那么 $\eta = 0$ ^[7-8]。

(5) 缓动、快进与关键帧。

设置好上面所有显示方法之后,还需要设置动画显示帧。

很多时候,希望每一帧显示的时间不一样长,比如一个开灯的动画中,开灯只是一瞬间的事情,希望灯亮前的黑暗状态和灯亮后的明亮状态长一些。这里利用关键帧来实现,将开灯动作的动画设置只显示一次或少数次,将动作前和动作后的动画设置多次,即可模拟这个动画。

在一个汽车下坡的动画中,汽车在坡顶的速度应该较慢,而在坡底的时间则应该较快,这里可以按照牛顿第二定律,设置汽车在坡顶的动画循环多次,而在坡底的动画循环次数少一些,就可以真实呈现这一物理情景。

制作动画过程中,用户也可以利用函数修改动画的缓动或快进效果,通过将某一帧的显示次数减少来模拟快进,而将某一帧的显示次数增加来模拟缓动,十分容易。

(6) 绘制。

到此,已经做好了一切前期准备工作,接下来需要做的是将图片绘制到画布上。这里利用 HTML5 的 canvas 标签,canvas 就像一个画册一样,每一个 canvas 元素都有一个绘图上下文,相当于这本画册中的一页,可以在 canvas 绘图上下文上绘制任意图形。支持 HTML5 canvas 标签的浏览器支持多个绘图上下文,并且通过提供不同的 API 提供绘制功能。下面简单介绍一下 canvas 的基础用法:

a. 首先需要创建 canvas 元素,规定 canvas 的 id、宽度、高度等属性;

b. 然后通过 id 查找 canvas 元素,创建一个绘图上下文,相当于根据画册的名字找到画册,然后在里面插入一页空白画纸,以便待会儿可以开始画画;这一段需要利用 JavaScript 实现;

c. 接下来就可以在定义好的绘图上下文中绘制任意图形。

canvas 支持一系列函数,可以绘制直线、任意曲线、矩形、圆等,利用这些基本函数可以绘制任意想要的图形,还支持填充、渐变等;也支持绘制多种格式的图片;还可以裁剪图片,以绘制动画。

对象类的绘制利用 drawImage 函数实现,且支持关键帧绘制。用户只需要规定好每次动画循环显示的帧顺序和次数,draw 函数会自动根据设置显示动画,而不是死板地按照多帧图片的顺序来播放动画,具有更高的灵活性^[9-12]。

由上可知,对象类定义了三种动画运动方式,用户还可以根据自己的需求,定义其他的运动方式,这个动画框架具有很高的可扩展性。

2.2.3 工具类

工具类主要封装了滤镜函数具体算法,算法虽然复杂,但是这部分提供了非常简单的外部接口给用户使用。

2.3 特点

该动画框架在普通动画框架的基础上,取其精华,去其糟粕,并适当添加了新的元素,更贴合现在的需求和最新的技术特点,使用更方便。它的特点如下:

(1) 时间轴:用户只需要在时间轴上创建一个动画,规定好每个动画的出现时间、结束时间、起点终点,就可以制作一个最简单的动画;

(2) 预分割:在初始化舞台时,预分割所有多帧图片,将其分割成一帧一帧的,并添加所要求的滤镜效果,然后存入数组,使用时只需读取即可,不必再分割,可以大大提高绘制时间;

(3) 滤镜效果:多个基本滤镜函数保证用户可以随心所欲地 PS 自己的动画,展现不一样的动画效果;

(4) 物理效果:多种物理效果函数可以让用户随心所欲模拟真实场景,增加真实感;

(5) 答卷式编程:对外用户接口是一个包含所有信息的数组,用户只需要按照数组中的对象的对象名填写自己的需求即可,例如初始时间、结束时间、开始位置、结束位置、滤镜效果、物理效果、移动方式等,即便是不熟悉 JS 语言的用户也可以很方便的编程。

3 基于 HTML5 和 JavaScript 的动画框架的实例应用

下面分功能展示几个简单的动画效果。

3.1 运动轨迹展示

运动轨迹效果如图 2 所示,可以看到,太阳和两条小鱼分别按照设定好的移动方式运动。

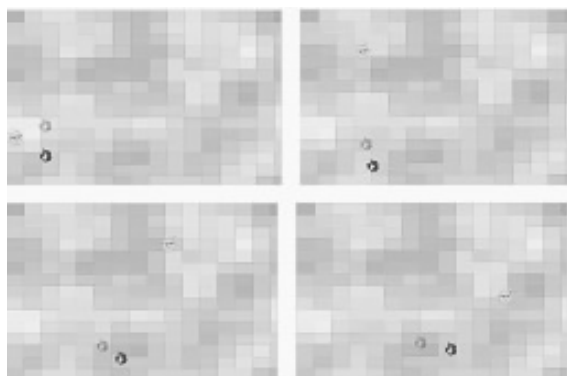


图 2 运动轨迹效果

3.2 滤镜效果展示

图 3 展示了高斯模糊滤镜和透明滤镜效果,可以看到离得较近的那条鱼颜色深一些,而远处那条鱼要浅一些,这是因为它们的透明度不同。而且颜色深的那条鱼,在第一幅图中很清楚,但是在第二幅图中因为高斯模糊滤镜的作用变得模糊了。

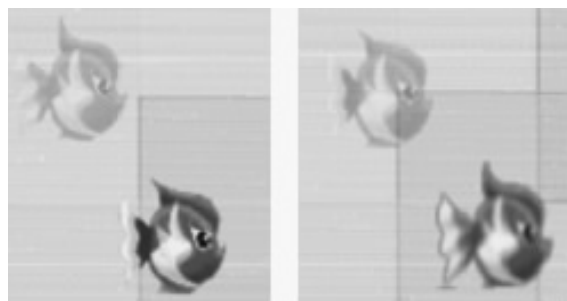


图 3 滤镜效果展示

3.3 动画实例

利用这个框架,编写了一个模拟运输煤炭的动画,接口参数如下所示:

```
img1 = {  
  image : new Image(),  
  filterimage : new Array(),  
  starttime : 1,  
  endtime : 4 000,  
  cutpositionX : 0,  
  cutpositionY : 0,  
  cutsizeX : 200,  
  cutsizeY : 186,  
  startX : 100,  
  startY : 450,  
  endX : 600,  
  endY : 400,  
  imagewidth : 200,  
  imageheight : 2 232,  
  drawpositionX : 100,
```

```
drawpositionY : 450,
showsizewidth : 50,
showsizeheight : 46.5,
frame : [0,0,0,0,0, 1,1,1,1,1,2,2,2,2,2, 3,3,3,3,3],
circle : -1,
direction : "endwise",
filter : [ "boxblurfilter", true , [0 ,0 ,0 ]],
type : "Sprite"
};
img1. image. src="back. jpg";
```

上面的代码创建了一个从 1 开始,持续到 4 000 的动画,模仿上面的代码,还可以创建别的动画。按照上述模式,创建了一个有关物流运输的动画,虽然实际编程时,不会真地创建一个时间轴,但是就像文章的中心线一样,时间轴贯穿整个动画,它将各个分动画串联成一个完整的、有意义的动画,为了更直观地展示物流动画中各分动画的关系,绘制了图 4(实际编程不需要创建时间轴)。

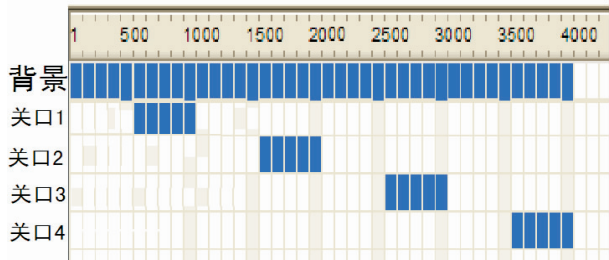


图 4 物流动画在时间轴中的描述

这个动画完整地展示了运煤车从进入卸煤地点到出卸煤地点整个过程中有关进门,检测,称重,卸货,再称重,离开的全部过程,这里利用关键帧设置,使得汽车前进动画持续时间长,而在每一个检测处的停留时间短,动画效果流畅,制作过程简单。

4 结束语

综上完成了一个动画框架该有的基本功能,并使

用该引擎制作了几个简单的小动画。该框架分层构建框架的底层程序,具有较强的重用性和可维护性。它具有高度的可扩展性,当用户所需的运动方式或者滤镜效果没有被定义时,用户只需要给出简单的计算公式,即可添加想要的结果。文中由于篇幅有限,部分程序代码没有给出。

参考文献:

[1] 王建民,郑子彬,麦章灿,等. 一个交互自适应手机游戏引擎的设计与实现[J]. 系统仿真学报,2009(10):3120-3122.

[2] 叶 绿. 一个建立在 JXTA 平台上的对等网络游戏框架的设计[J]. 计算机工程与应用,2004,40(17):144-147.

[3] 赵丽娟,朱全银,张 帅,等. 基于 J2ME 的移动网络游戏设计与实现[J]. 计算机工程与设计,2010,31(12):2720-2725.

[4] 冯科融,王和兴. HTML5 网页游戏分析[J]. 网络与通信,2012(24):71-72.

[5] 刘华星,杨 庚. HTML5-下一代 Web 开发标准研究[J]. 计算机技术与发展,2011,21(8):54-58.

[6] 宋 宇,谭 浩. 游戏引擎开发技术析[J]. 福建电脑,2007(8):31-32.

[7] 成迟慧,石教英,徐迎庆,等. 基于物理模型的窗帘运动实时动画[J]. 软件学报,2000,11(9):1228-1236.

[8] 牛红攀,高 勇,侯忠明. 图形引擎与物理引擎结合的研究与实现[J]. 计算机仿真,2011,28(6):299-303.

[9] Lubbers P, Albers B, Salim F. HTML5 高级程序设计[M]. 北京:人民邮电出版社,2011.

[10] Freeman E, Robson E. Head first HTML5 programming[M]. USA: O'Reilly Media, 2011.

[11] Grady M. Functional programming using JavaScript and the HTML5 canvas element[J]. Journal of computing sciences in colleges, 2010, 26: 97-105.

[12] Fulton S, Fulton J. HTML5 Canvas[M]. USA: O'Reilly Media, 2011.

(上接第 4 页)

[5] Fritz M H, Leinonen R, Cochrane G, et al. Efficient storage of high throughput DNA sequencing data using reference-based compression[J]. Genome research, 2011, 21(5):734-740.

[6] 方小永, 骆志刚. DNA 序列拼接的分布式并行处理[J]. 计算机工程与科学, 2005, 27(2):71-73.

[7] Grumbachand S, Tahi F. A new challenge for compression algorithms: Genetic sequences [J]. Information processing & management, 1994, 30(6):875-886.

[8] Chen X, Li M, Ma B, et al. DNACompress: Fast and effective DNA sequence compression [J]. Bioinformatics, 2002, 18

(12):1696-1698.

[9] Rajarajeswari P, Apparao A. DNABIT compress-genome compression algorithm[J]. Bioinformation, 2011, 5(8):350-360.

[10] Matsumoto T, Sadakane K, Imai H. Biological sequence compression algorithms[J]. Genome informatics series, 2000, 11: 43-52.

[11] 纪 震, 周家锐, 姜 来, 等. DNA 序列数据压缩技术综述[J]. 电子学报, 2010(5):1113-1121.

[12] 郑翠芳. 几种常用无损数据压缩算法研究[J]. 计算机技术与发展, 2011, 21(9):73-76.

基于HTML5和JavaScript轻量型动画框架开发

作者：[平淑文](#)，[潘珏羽](#)，[张学金](#)，[杜晓荣](#)，[PING Shu-wen](#)，[PAN Jue-yu](#)，[ZHANG Xue-jin](#)
，[DU Xiao-rong](#)

作者单位：[中山大学 电力电子与控制技术研究所, 广东 珠海, 519082](#)

刊名：[计算机技术与发展](#)

ISTIC

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2013(12)

本文链接：http://d.g.wanfangdata.com.cn/Periodical_wjfz201312002.aspx