



基于 Web3D 的虚拟实验教学平台

苏 雪

(武汉铁路职业技术学院,湖北 武汉 430074)

摘 要:以 SpiderGL 项目为基础,构建了一个无需插件安装的、具有 Web3D 交互能力的三维虚拟实验教学平台。该平台能给用户提供强烈的沉浸感,并具有良好的双向交互性、可扩展性以及安全可靠。

关键词: Web3D;虚拟现实;实验教学平台

中图分类号: TP311

文献标识码: A

文章编号: 1671-931X (2011) 05-0070-04

一、概述

网络虚拟现实技术的出现,给网络教学带来了新的教学手段,可以有效地提高网络教学的质量。网络虚拟现实的技术优势对提高网络教学质量有着重要意义,只是由于网络带宽及个人 PC 计算能力还不能满足高质量的 3D 实验教学内容的展示,因此目前在网络实验教学中的应用较少。

最近几年,有关 3D 内容交付的软硬件技术不断改善。首先,强大的并行 GPU 减轻了 CPU 的处理负荷。其次是新一代的 Web 浏览器中采用的 TraceMonkey、V8 及 SquirrelFish 编译器显著改善了 JavaScript 的执行性能。尤其是由 Khronos 新近提出的 WebGL 标准,将 Web 浏览器和 GPU 紧密地关联在一起了。这一切使得在 Web 上呈现复杂 3D 图形环境成为了可能。也使得基于 Web 的虚拟 3D 实验教学平台的设计有了新的选择。

基于 WebGL 的网络虚拟现实技术利用它在技术上的优势,能够充分展示知识内容、激发学生的创造性、开展虚拟实验和进行廉价的技术训练,使得采用网络虚拟现实技术的网络教学可以取得很好的教学效果。

本文结合混合现实教学环境的思想,提出了一种基于 WebGL 标准的虚拟实验教学平台的设计与实现方案。这在国内还是一个新的尝试。

二、相关工作

(一) Web3D

Web3D 技术是虚拟现实技术中的一种,通过使用这些技术可以开发出应用在网络页面中,达到照片级渲染效果的虚拟场景,实现相应的动画效果,并可根据浏览者输入的指令及时反馈信息,达到人机交互的目的。Web3D 还可以简单地被看成是 Web 技术和 3D 技术相结合的产物,网络性、三维性和交互性是其显著的本质特征。通过应用 Web3D 技术,用户可以在网上浏览以三维形式表现的物体,并对其进行交互性操作以体验身临其境的奇妙感受。

1995 年出现的 VRML(后来被 X3D 所取代)以文本格式描述 3D 场景的几何和材料属性,为了在 web 浏览器中渲染,需要安装平台制定的 plug-in。

Java Applets 在增加用户软件组件方面(不仅仅是 3D)可能是最实用的方法。Java Applets 的思路是 Applet 的 URL 及它的数据放置在 HTML 页面中,并且有第三方的组件 JVM 执行。各种类型的操作系统上 JVM 的实现是的 Java Applets 无处不在。同时还出现了将 OpenGL 绑定的工具如 JOGL,以增强对 3D 图形硬件的控制能力。类似的思路还有微软的 ActiveX 技术。不像 Java Applets,ActiveX 控制器不是字节编码,而是动态

收稿日期:2011-08-12

基金项目:湖北省教育厅“十一五”规划课题“高速铁路通信信号虚拟实训教学环境研究”(项目编号:2008B219)。

作者简介:苏雪(1972-),女,武汉铁路职业技术学院讲师,研究方向:计算机应用。

链接 Windows 库,与调用进程(如浏览器)共享内存空间,因而运行的速度更快。

这些技术允许将 3D 图形集成到 Web 页面中,但是都是通过第三方组件处理页面中的专有成员。

最近,Google 开始开发一个 3D 图形引擎 O3D。O3D 也是作为 Plug-in 部署在浏览器中,只是它不再是一个黑盒子而不可编程控制,他集成到浏览器中,扩展了它的 JavaScript 的 3D 图形能力,在 OpenGL 及 DirectX 基础上。O3D 是基于场景图形,而且提供工具以装载几种常用格式的 3D 场景。

WebGL 是 Khronos 组织提出的,如名称所暗示的,定义 JavaScript 类似针对 C++ 的 OpenGL API。WebGL 紧紧结合 OpenGL/ES2.0,并且使用 GLSL 作为着色器语言,意味着现有的应用的着色器核心可以在 JavaScript/WebGL 中重用。由于 WebGL 是一个规范,它由 Web 浏览器开发者实现它。目前,在最新的常用浏览器(Firefox、Chrome、Safari)都实现了现有 WebGL 版本。而且已有大量的 JavaScript 库正在开发以提供高级功能以创建 3D 图形应用。如 WebGLU,它是 WebGL 与 GLU 的结合,提供封装,在场景中放置 Camera 或者提供简单的几何图形元素,其他的库如 GLGE 或 SceneJS 使用 WebGL 实现场景图形基于渲染及动画引擎。

(二) SpiderGL

目前 WebGL 的问题是针对于 CG 算法的 C++ 模块和 JavaScript 模块的分离,这导致了应用执行速度,同时还有一下空白需要填补:如异步内容加载、模型数据从文件加载、封装等。SpiderGL 是一个 JavaScript 库为填充这些空白而设计。它扩展了 JavaScript 包含一个几何数据结构及算法,并封装他们的实现,进而超越 WebGL,其结构如图 1 所示。

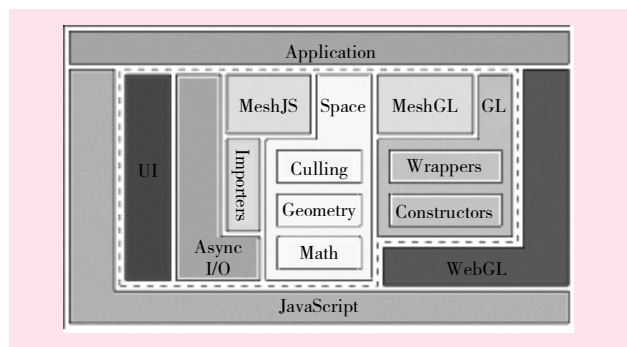


图 1 SpiderGL 结构示意图

(三) MiRTLE

MiRTLE (Mixed Reality Teaching and Learning Environment) 通过对 SpiderGL 平台及其功能模块的开发和应用,为本地和远程用户提供虚拟与现实相结合的双向协作环境。

三、整体设计

虚拟实验平台采用 B/S 架构设计,由具有 WebGL 解析能力的浏览器、Web 服务器组成。

(一) 系统体系架构

虚拟实验平台采用流行的 JSF 架构实现,如图 2

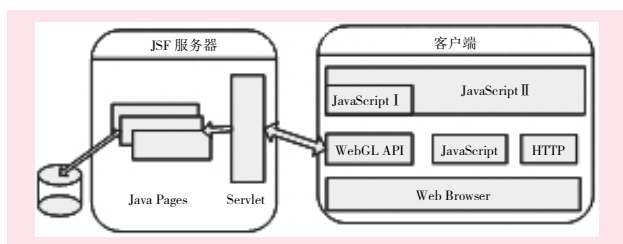


图 2 虚拟实验平台框架结构示意图

所示。客户端使用支持 WebGL 的浏览器,如 Chrome、Firefox 或 Safari。用户交互通过浏览器的 WebGL 接口交与服务端端的 Faceservlet 控制器处理。组成虚拟实验平台的 3D 构件元素,以 Collada 格式存储,通过 JavaBean 实现检索、提取、更新等操作。

(二) 系统特点

JSF 的主要优势之一在于它既是 Java Web 应用程序的用户界面标准又是严格遵循模型-视图-控制器(MVC)设计模式的框架。用户界面代码(视图)与应用程序数据和逻辑(模型)的清晰分离是 JSF 应用程序更易于管理。为了提供页面对应用程序数据访问的 JSF 上下文和防止对页面未授权或不正确的访问,所有与应用程序的用户交互均由一个前端 FacesServlet(控制器)来处理。

四、关键技术实现

(一) 建模

为了制作复杂的 Web3D 图形的场景模型通常借助传统的专业三维建模软件,如 3dsMax、Maya 等。

COLLADA 是一个中间语言的交互式的 3D 应用程序。它本质上是 XML3D 的资产,使用的 url 连结的内容。COLLADA 使物流内容创作工具,交互性的应用程序;它是一种无损的、可扩展的内容串行化和检索。内容可用标准工具处理在其原生型 XML 编码以适应目标的应用。COLLADA 也被用作作为一个交换格式,提供之间的一座桥梁创作中的应用。

为了将 WebGL 与 Collada 关联,需要由开发者在 Web 应用中实现一个软件接口。当前,常见的是采用紧耦合的数据包提供形式,由 Collada 离线处理 3D 内容,将内容封装成应用指定的数据包格式。如 O3D 将 Collada XML 转换成 JSON 格式。随着 HTML5 及相关标准的发展,越来越多的 3D Web 应用可以直接使用 Collada 格式内容。本文设计平台中采用的 Spider 就可以直接异步解析 Collada 格式数据,示意图如图 3 所示。这种结构的优势是利用服务器端 Collada 服务可实现内容的动态获取和创建。

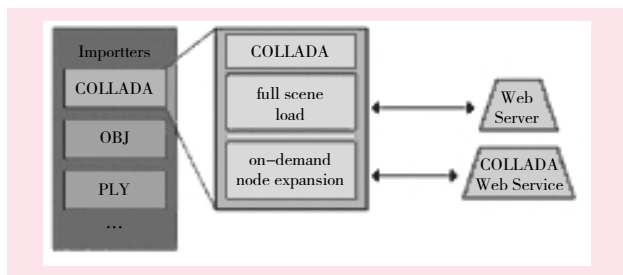


图 3 SpiderGL 与 Collada 结合示意图

(二)3D 渲染呈现

应用 SpiderGL 实现的虚拟实验教学平台中的虚拟实验场景的建立和渲染管线工作流与标准的 3D 虚拟现实应用的流程类似,如图 4 所示。

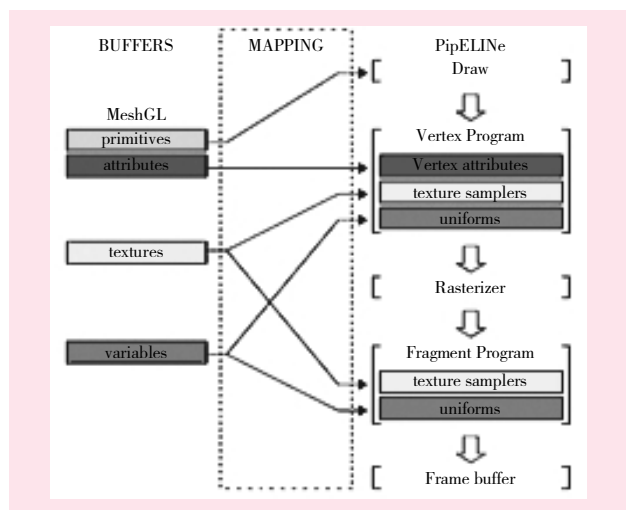


图 4 场景创建和渲染的流程示意图

嵌入动态 SpiderGL 内容的 Web 页面中,所有的页面逻辑都在 HEAD 部分定义,在 BODY 部分有页面结构及 UI 元件。在这些元件中,最重要的当然是 canvas 对象,也就是 WebGL 实现屏幕渲染的地方。

```
<canvas id="SGL_CANVAS" style="border: 1px solid gray" -
```

```
width="900" height="600"></canvas>
```

Canvas 在脚本的末尾输出的区域 register, 对于一个脚本对象来说,它是一个连接各种 canvas 事件的特殊函数。

```
var glMolViewer = new SpiderGLMolViewer();
sglRegisterCanvas("SGL_CANVAS", glMolViewer, 30.0);
```

对我们的虚拟实验场景建立来说,glMolViewer 对象是主要成员,这个对象的结构包含 SpiderGL 提供的事件处理子系统。Canvas 事件触发一个专有函数,SpiderGL 在 JavaScript 语言的基础上开发了一个动态加载和删除的 listener 和 redirect 时间。下面的这个例子是一个 listener 示例:

```
SpiderGLMolViewer.prototype = {
  load : function (gl) {...},
  unload : function (gl) {...},
  update : function (gl, dt) {...},
  keyDown : function (gl, keyCode, keyString) {...},
  keyUp : function (gl, keyCode, keyString) {...},
  keyPress : function (gl, keyCode, keyString) {...},
  mouseDown : function (gl, button, x, y) {...},
  mouseUp : function (gl, button, x, y) {...},
  mouseMove : function (gl, x, y) {...},
  mouseWheel : function (gl, wheelDelta, x, y) {...},
  click : function (gl, button, x, y) {...},
  dblClick : function (gl, button, x, y) {...},
```

```
resize : function (gl, width, height) {...},
draw : function (gl) {...},
};
```

初始化和数据加载由 load 函数完成,他是渲染器的主要特性,包含输入数据的加载和着色器的编译。

```
load : function (gl) {...}
this.xform = new SglTransformStack();
this.camera = new SglFirstPersonCamera();
this.camera.lookAt(0.0, 0.0, 1.5, 0.0, 0.0, 0.0,
-sglDegToRad(0.0));
this.viewMatrix = this.camera.matrix;
this.trackball = new SglTrackball();
[...]
this.prog = new SglProgram(gl, [sglNodeText("-MY_VERTEX_SHADER"),
sglNodeText("-MY_FRAGMENT_SHADER")]);
[...]
```

```
var TextureOptions = { generateMipmap : true, minFilter: gl.LINEAR_MIPMAP_LINEAR,
onload : this.ui.requestDraw
```

```
};
```

```
var ColorTexture = new SglTexture2D(gl, "molecule_color.png",
textureOptions);
```

```
[...]
```

```
this.meshJS = new SglMeshJS();
```

```
this.meshJS.importOBJ("molecule.obj", true, function (-m, url) {...})
```

```
this.meshGL_MOL = that.meshJS.toPackedMeshGL(gl, "-triangles", 65000);
```

```
[...] );
```

```
[...]
```

```
var pdtxt = sglLoadFile("mol.pdb");
```

```
this.atomslist = [];
```

```
var res = AtomListFromPDB(this.atomslist, pdtxt);
```

```
[...]
```

```
this.timeOffset = 0.0;
```

```
this.stereoEnabled = false;
```

```
this.particlesEnabled = true;
```

```
},
```

这种整体的数据管理方式对一个单一的、紧凑的网页数据库来说是很好的方式。更多的时候,还需要利用异步加载机制,使得大型的流/离散数据库的加载或者用户动态加载远程文件更加有效。绘制函数中包含了实际的渲染代码:

```
draw : function (gl) {
  gl.clearColor(0.0, 0.0, 0.0, 1.0);
  gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT | gl.STENCIL_BUFFER_BIT);
  gl.viewport(0, 0, w, h);
  this.xform.projection.loadIdentity();
  this.xform.projection.perspective(sglDegToRad(45.0), -w/h, 0.1, 10.0);
  [...]
```

```
var uniforms = {  
    u_mvp : this.xform.modelViewProjectionMatrix,u_normal_mat : this.xform.viewSpaceNormalMatrix,u_dotrasp : this.atomsEnabled,u_mousepos :[this.ui.mousePos.x, this.ui.mousePos.-y]  
};  
var samplers = { s_texture_c:this.ColorTexture,s_texture_b : this.BumpTexture};  
[...]  
gl.enable(gl.DEPTH_TEST);  
gl.enable(gl.CULL_FACE);  
sglRenderMeshGLPrimitives(this.meshGL_MOL, "triangles -", this.prog, null, uniforms, samplers);  
[...]  
},
```

这个函数可以持续或按需调用，当用 sglRegisterCanvas 函数注册一个 canvas 后，如果参数是 0，那么 canvas 只是重画，否则该参数表示画面呈现理想的帧率。在每一个“tick”，SpiderGL 将调用更新函数，然后重画。在这两种情况中，只要检测到一个 WebGL 帧缓冲区变化，Html 渲染引擎将启动一个页面制作操作。

GLSL 着色器包含在 Web 页面的 HEAD 部分的脚本实体中。

```
<script id="MY_VERTEXSHADER" type="x-shader/x-vertex"> [...] </script>
```

一个虚拟实验的实现代码可以按上述示例的方式逐一设计和组织。这个简单的示例是一个优化的起点，对于有经验的图形编程人员来说非常的简单。

五、总结与展望

本文提出了基于 WebGL 标准的 Web 虚拟实验平台的实现技术。通过使用 WebGL 增强的 SpiderGL 项目库，可以创建基于 Web 的虚拟实验平台原型，它不仅可用户定制，而已可以使用先进的着色和渲染技术。本文讨论了这个技术实现的可能性，描述了可获得的组件以及如何用这些组件创建一个交互式虚拟化方案。因为 WebGL 标准还未最终定稿，且 SpiderGL 项目还是进展当中，这个技术方案还需要进一步的完善。

参考文献：

- [1] Callaghan V, Gardner M. A Mixed Reality Teaching and Learning Environment[A].Proc. of the 1st International Conference on Hybrid Learning[C].Hong Kong, 2008: 60-61.
- [2] 张峰,李慧丽. JavaWeb2.0 架构开发与项目实战[M].北京:清华大学出版社, 2010.
- [3] DI BENEDETTO, [M].2010. SpiderGL: 3D Graphics for Next-Generation WWW. <http://spidergl.org/>.
- [4] GROUP, T. K.. WebGL - OpenGL ES 2.0 for the Web [EB/OL]. <http://www.khronos.org/webgl/>, 2011-06-30.

[责任编辑：刘 骋]

Virtual Experimental Learning Platform Based on Web3D

SU Xue

(Wuhan Railway Vocational College of Technology, Wuhan 430074, China)

Abstract:With the concept of mixing real teaching and virtual learning environment, a powerful virtual experimental learning platform with Web3D interactivity is proposed. It is based on project SpiderGL and free from plug-in installation. The platform will render users a strong sense of immersion, and has a good two-way interactivity, scalability, and secure reliability.

Keywords:Web3D; virtual reality; experimental learning platform