### 用HT for Web实现3D列车行进动画

HT for Web有一套强大的基于WebGL技术的图形引擎，可以用js代码实现3D图像的建模和动画。现在用HT for Web来实现一个3D列车行进的动画来带大家了解一下如何使用这套框架。

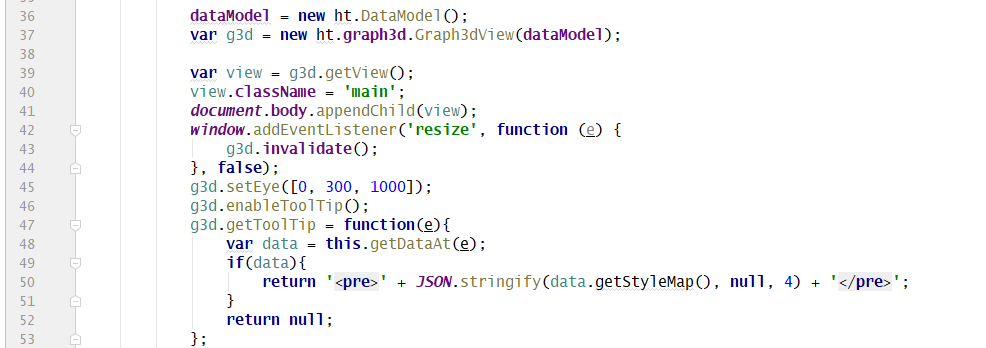


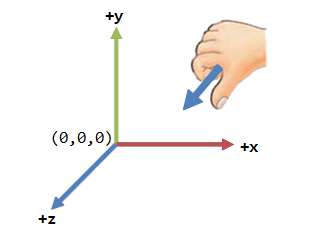
首先需要引入如图的3个js文件。ht.js是基础库，ht-animation.js是动画库,texture.js是图像素材库，是用户自己定义的。



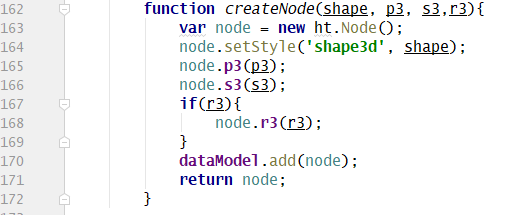
这里定义了两组图片，第一个参数是name，后面引用的时候就是用的它，第二第三分别是宽度高度，第四个参数是图片路径。

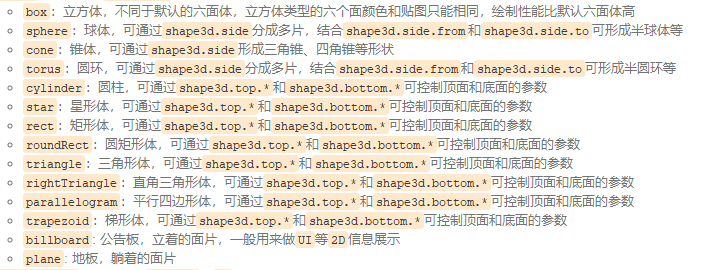
初始化工作：

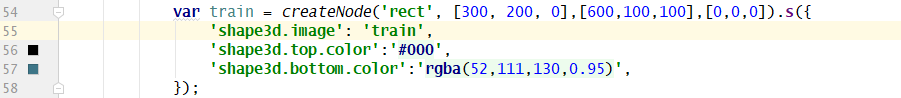


这里是对整个图像的初始化操作以及一些设置。ht就是HT for Web的一个全局参数，DataModel就是数据模型，所有的3D建模都是由它来管理，建出来模型都得通过add方法加入进DataModel才能显示。Graph3DView是3D渲染引擎组件，可视化呈现数据模型的三维环境场景。g3d.getView()是用来获得html的根div的，通过document.body.appendChild来添加到页面中。42~44行是给页面改变大小事件增加一个监听，调用g3d.invalidate()来进行重新渲染。g3d.setEye()就是眼睛所在位置，也就是观察点。[x,y,z]代表了3d坐标轴的坐标。HT for Web采用的是坐标系如图。g3d.enableToolTip()是开启提示工具。47~53行是当鼠标放在画面上时提示工具要显示的内容。

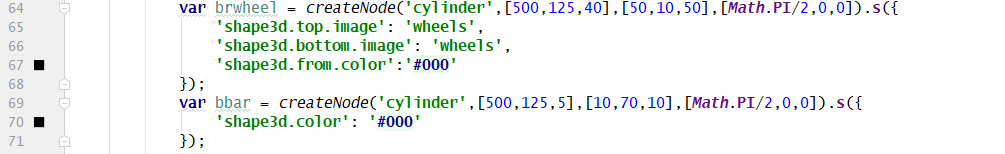
建模：

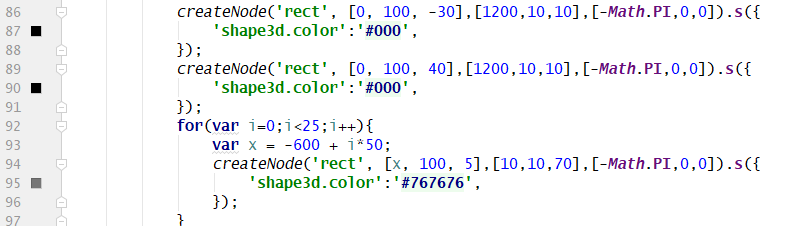
为了减少冗余代码，可以把建模的代码抽出一个公共方法，只要传入对应的参数即可。

new ht.Node()就是新建一个模型。node.setStyle就是定义这个模型的形体，第一个参数固定，代表3D模型，第二个参数有：HT for Web支持一些方法的简写，p3就是setPosition3d和getPosition3d的简写，不传参数就是get，传参就是set，传入[x,y,z]来给模型定位。s3就是setSize3d和getSize3d的简写，用来定义模型大小。r3就是setRotation3d和getRotation3d的简写，用来旋转模型，旋转一圈为2\*Math.PI。dataModel.add()来把模型加入画布。

要建一个车身的模型可以用这段代码。

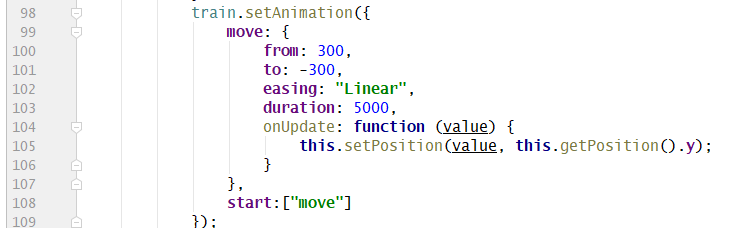
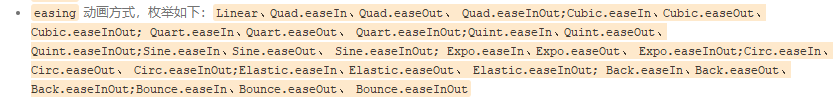
这样就画了一个大小为[600,100,100]，位置在[300,200,0]不旋转的一个矩形体。s就是getStyle和setStyle的缩写，用来定义样式。这里用shape3d.image来定义整体贴图，train就是在texture.js定义好的Image。shape3d.top.color就是顶面的颜色,shape3d.bottom.color就是底面的颜色。用同样的方法可以画出火车的轮子、轮轴、铁轨等。





动画：

接下来就是需要让模型动起来了。对于车身，需要让他从右往左水平移动。

train.setAnimation就是给车身增加动画，这个方法接收一个json对象，除了start是保留name，其他都可以作为动画的名称。from是开始值，to是终值，easing就是动画方式，有如下枚举：

duration就是持续时间，以ms为单位。onUpdate就是在数值从from到to的过程中每变动一次就会调用的方法，在这里对模型进行移动。这里就是不断改变模型的x坐标。start的意思是指定动画从哪个开始。

比较复杂一点的是轮子的动画，不仅要让它平移，还要旋转：

这里的move都一样，区别是多了个rotate的动画。旋转是从0到Math.PI\*2，就是整整一圈，repeat就是是否重复，接收一个bool值，onUpdate里让轮子旋转起来。因为平移和旋转是同时发生的，所以start里指定了两个动画。

最后就是让动画动起来：

就是启动全局动画定时器，参数是刷新间隔，以ms为单位，指每隔多久进行一次重新渲染。

至此，一个列车在轨道上行进的3D动画就完成了。

完整代码：

<!DOCTYPE **html**>  
<**html**>  
<**head**>  
 <**title**>shape3d</**title**>  
 <**meta charset="UTF-8"**>  
 <**style**>  
 **html**, **body** {  
 **padding**: 0**px**;  
 **margin**: 0**px**;  
 }  
 .**main** {  
 **margin**: 0**px**;  
 **padding**: 0**px**;  
 **position**: **absolute**;  
 **top**: 0**px**;  
 **bottom**: 0**px**;  
 **left**: 0**px**;  
 **right**: 0**px**;  
 }  
 </**style**>  
 <**script**>  
 **htconfig** = {  
 **Default**:{  
 **toolTipDelay**: 100,  
 **toolTipContinual**: **true** }  
 };  
 </**script**>  
 <**script src="js/ht.js"**></**script**>  
 <**script src="js/ht-animation.js"**></**script**>  
 <**script src="js/texture.js"**></**script**>  
  
 <**script**>  
 **function** *init*(){  
  
 **dataModel** = **new** ht.DataModel();  
 **var** g3d = **new** ht.**graph3d**.Graph3dView(**dataModel**);  
  
 **var** view = g3d.getView();  
 view.**className** = **'main'**;  
 ***document***.**body**.appendChild(view);  
 ***window***.addEventListener(**'resize'**, **function** (e) {  
 g3d.**invalidate**();  
 }, **false**);  
 g3d.setEye([0, 300, 1000]);  
 g3d.enableToolTip();  
 g3d.getToolTip = **function**(e){  
 **var** data = **this**.getDataAt(e);  
 **if**(data){  
 **return '<pre>'** + ***JSON***.stringify(data.getStyleMap(), **null**, 4) + **'</pre>'**;  
 }  
 **return null**;  
 };  
 **var** train = *createNode*(**'rect'**, [300, 200, 0],[600,100,100],[0,0,0]).**s**({  
 **'shape3d.image'**: **'train'**,  
 **'shape3d.top.color'**:**'#000'**,  
 **'shape3d.bottom.color'**:**'rgba(52,111,130,0.95)'**,  
 });  
 **var** blwheel = *createNode*(**'cylinder'**,[500,125,-30],[50,10,50],[***Math***.**PI**/2,0,0]).**s**({  
 **'shape3d.top.image'**: **'wheels'**,  
 **'shape3d.bottom.image'**: **'wheels'**,  
 **'shape3d.from.color'**:**'#000'** });  
 **var** brwheel = *createNode*(**'cylinder'**,[500,125,40],[50,10,50],[***Math***.**PI**/2,0,0]).**s**({  
 **'shape3d.top.image'**: **'wheels'**,  
 **'shape3d.bottom.image'**: **'wheels'**,  
 **'shape3d.from.color'**:**'#000'** });  
 **var** bbar = *createNode*(**'cylinder'**,[500,125,5],[10,70,10],[***Math***.**PI**/2,0,0]).**s**({  
 **'shape3d.color'**: **'#000'** });  
  
 **var** flwheel = *createNode*(**'cylinder'**,[100,125,-30],[50,10,50],[***Math***.**PI**/2,0,0]).**s**({  
 **'shape3d.top.image'**: **'wheels'**,  
 **'shape3d.bottom.image'**: **'wheels'**,  
 **'shape3d.from.color'**:**'#000'** });  
 **var** frwheel = *createNode*(**'cylinder'**,[100,125,40],[50,10,50],[***Math***.**PI**/2,0,0]).**s**({  
 **'shape3d.top.image'**: **'wheels'**,  
 **'shape3d.bottom.image'**: **'wheels'**,  
 **'shape3d.from.color'**:**'#000'** });  
 **var** fbar = *createNode*(**'cylinder'**,[100,125,5],[10,70,10],[***Math***.**PI**/2,0,0]).**s**({  
 **'shape3d.color'**: **'#000'** });  
 *createNode*(**'rect'**, [0, 100, -30],[1200,10,10],[-***Math***.**PI**,0,0]).**s**({  
 **'shape3d.color'**:**'#000'**,  
 });  
 *createNode*(**'rect'**, [0, 100, 40],[1200,10,10],[-***Math***.**PI**,0,0]).**s**({  
 **'shape3d.color'**:**'#000'**,  
 });  
 **for**(**var** i=0;i<25;i++){  
 **var** x = -600 + i\*50;  
 *createNode*(**'rect'**, [x, 100, 5],[10,10,70],[-***Math***.**PI**,0,0]).**s**({  
 **'shape3d.color'**:**'#767676'**,  
 });  
 }  
 train.setAnimation({  
 **move**: {  
 **from**: 300,  
 **to**: -300,  
 **easing**: **"Linear"**,  
 **duration**: 5000,  
 onUpdate: **function** (value) {  
 **this**.setPosition(value, **this**.getPosition().**y**);  
 }  
 },  
 **start**:[**"move"**]  
 });  
 [blwheel,brwheel,bbar].forEach(**function**(obj){  
 obj.setAnimation({  
 **move**:{  
 **from**: 500,  
 **to**: -100,  
 **easing**:**"Linear"**,  
 **duration**:5000,  
 onUpdate:**function**(value){  
 **this**.setPosition(value, **this**.getPosition().**y**);  
 }  
 },  
 **rotate**:{  
 **from**:0,  
 **to**:***Math***.**PI**\*2,  
 **easing**:**"Linear"**,  
 **duration**:1000,  
 **repeat**:**true**,  
 onUpdate:**function**(value){  
 **this**.**r3**([**this**.getRotationX(),**this**.getRotationY(),value]);  
 }  
 },  
 **start**:[**"move"**,**"rotate"**]  
  
 });  
 });  
 [flwheel,frwheel,fbar].forEach(**function**(obj){  
 obj.setAnimation({  
 **move**:{  
 **from**: 100,  
 **to**: -500,  
 **easing**:**"Linear"**,  
 **duration**:5000,  
 onUpdate:**function**(value){  
 **this**.setPosition(value, **this**.getPosition().**y**);  
 }  
 },  
 **rotate**:{  
 **from**:0,  
 **to**:***Math***.**PI**\*2,  
 **easing**:**"Linear"**,  
 **duration**:1000,  
 **repeat**:**true**,  
 onUpdate:**function**(value){  
 **this**.**r3**([**this**.getRotationX(),**this**.getRotationY(),value]);  
 }  
 },  
 **start**:[**"move"**,**"rotate"**]  
  
 });  
 });  
 **dataModel**.enableAnimation(50);  
 }  
 **function** *createNode*(shape, p3, s3,r3){  
 **var** node = **new** ht.Node();  
 node.setStyle(**'shape3d'**, shape);  
 node.**p3**(p3);  
 node.**s3**(s3);  
 **if**(r3){  
 node.**r3**(r3);  
 }  
 **dataModel**.add(node);  
 **return** node;  
 }  
  
  
  
 </**script**>  
</**head**>  
<**body onload="***init*();**"**>  
</**body**>  
</**html**>