

C3DL

webGL made easy!

WebGL Seminar ,28-01-2011

Tampere university of technology

Veerakishore Goduguluri

Department of Software Systems

Introduction

- The Canvas 3D JS Library (C3DL) is an Open source JavaScript library that will make it easier to write 3D applications with in web page.
- It is a library of math, scene, and 3d object classes that makes WebGL more accessible for developers that want to develop 3D content in browser but do not want to have to deal in depth with the 3D math needed to make it work.

Background

- C3DL is acronym for Canvas 3D JavaScript Library.
- C3DL was first developed as part of the CATGames Research network working on providing a middle layer api for Canvas 3D (the precursor of WebGL).
- Most of the C3DL is developed by Seneca College and other developers from various backgrounds and geographical locations.
- Sponsored project of Mozilla and other gaming companies.

Technical overview

- Initially this library is built on the top of Mozilla Firefox Canvas 3D add-on and OpenGL ES 2.0.
- The main goal of C3DL is to provide 3D content in the browser without having to go into the intricacies of WebGL and 3D math.
- COLLADA format is used for models. These files have .dae extension
- C3dapi.js is the file that includes every other function of the library, this file should be included in any .js file where we have to use the library functions.

Features offered by C3DL

- Collada model loading
- Mouse Picking
- Camera system
- Lighting and Directional light system
- Orbiting light system
- Lines and Dots
- Effects system
 - cartoon (with or without outlines)
 - greyscale
 - solid colour
 - sepia
 - gooch
- Material
- Scene
- Particle System
- Html events

Technical overview

- Main object that are required to display are
 1. Model to be displayed
 2. Scene
 3. WebGL renderer
 4. Camera

Technical overview

Steps to display any thing in canvas

1. Imported the model of something to put in the scene.
2. Create a scene
3. Create WebGL renderer
4. Attach renderer to scene
5. Once renderer is ready Create a Collada object that will contain a imported model
6. Set parameters to Collada object depending on requirements
7. Add the Collada object to scene
8. Create a camera
9. Place the camera at particular position
10. Point the camera towards a position
11. set camera to scene
12. Finally start the scene.

Example a simple scene

- The HTML document

- `<html>`

- `<head>`

- `<title>Cavas3D tutorial #3: Callbacks</title>`

- `<script type="application/javascript" src="../../canvas3dapi/c3dapi.js"`

- `></script>`

- `<script type="application/javascript" src="tutorial.js"></script> </head>`

- `<body> <canvas id="tutorial" style="border: 2px solid blue" width="500" height="500"></canvas> </body> </html>`


```
c3dl.addMainCallBack(canvasMain, "tutorial");
c3dl.addModel("duck.dae"); var duck;
canvasMain(canvasName){
c3dl.Scene object scn = new c3dl.Scene();
scn.setCanvasTag(canvasName);
    renderer = new c3dl.WebGL(); renderer.createRenderer(this);
scn.setRenderer(renderer);
scn.init(canvasName);
if(renderer.isReady() ) {
duck = new c3dl.Collada();
duck.init("duck.dae");
duck.setAngularVel(new Array(0.0, 0.001, 0.0));
scn.addObjectToScene(duck);
var cam = new c3dl.FreeCamera();
cam.setPosition(new Array(200.0, 300.0, 500.0));
cam.setLookAtPoint(new Array(0.0, 0.0, 0.0));
scn.setCamera(cam);
scn.startScene(); } }
```

Examples

Orbit camera

- <http://www.c3dl.org/index.php/webgl-demos/cross-browser-orbiter/>

Particle system

- <http://www.c3dl.org/index.php/webgl-demos/rts-prototype/>
- <http://www.c3dl.org/index.php/webgl-demos/particle-systems-demo/>

Asteroid 3D

- <http://www.c3dl.org/index.php/webgl-demos/asteroids-3d/>

Animation

- http://www.c3dl.org/index.php/webgl-demos/mocap_spheres/

Evaluation

- Very easy to use and understand.
- As it is a JavaScript library no special plug-in are required.
- It is open source.
- Good Documentation.
- Browser independent.
- Still improving.
- It has all the feature to develop 3d Applications for web
- Active community.



QUESTIONS?