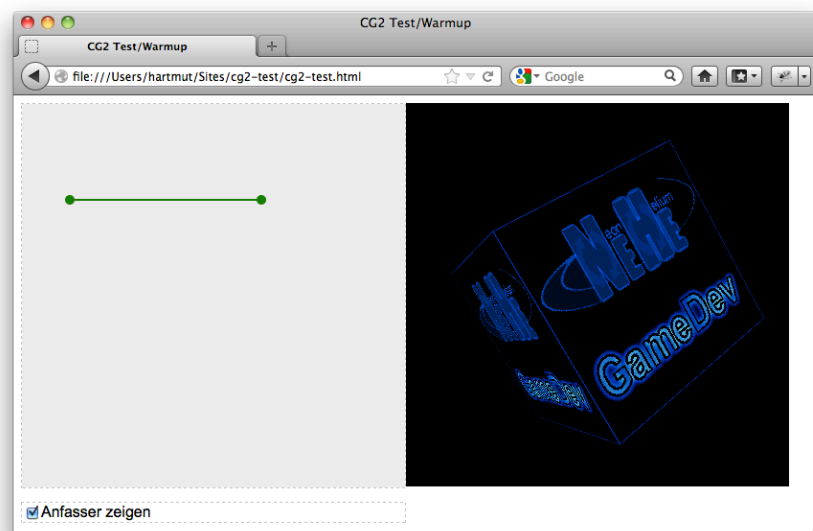


## Computergrafik 2

# Einrichten und Test der Entwicklungsumgebung für JavaScript+WebGL



Als Grundlage für die folgenden Übungsaufgaben setzen Sie einen eigenen Webserver auf, installieren einen WebGL-fähigen Browser und bringen ein Testprogramm zum laufen. Dies ermöglicht Ihnen erste Schritte in Ihrer Entwicklungsumgebung für JavaScript und WebGL im Browser.

## Installation von Webserver, Browser, und Übungsframework

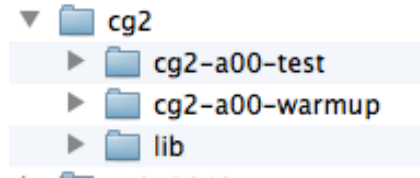
Als Browser für WebGL-Inhalte empfehle ich Chrome oder Firefox. Der Microsoft Internet Explorer ist mangels WebGL-Unterstützung zur Zeit nicht für die CG2-Übungen geeignet. Bitte installieren Sie eine relativ aktuelle Version eines der beiden Webbrowser.

Für fortgeschrittene Übungen mit JavaScript und WebGL ist es sinnvoll, einen eigenen lokalen Webserver zu betreiben; dies vermeidet diverse Probleme wie z.B. mit dem Nachladen weiterer Ressourcen aus dem Dateisystem. Daher installieren / starten Sie bitte zunächst einen geeigneten Webserver auf Ihrem System (auf den Labor-Rechnern ist XAMPP bereits installiert). Falls Ihr System noch keinen Webserver "mitbringt", können Sie z.B. sehr einfach die XAMPP-Distribution herunterladen und zum laufen bringen (Windows, Mac OS, Linux):

- <http://www.apachefriends.org/de/xampp.html>

Als nächstes laden Sie das Test-Framework **cg2.zip** für diese Übung von Moodle herunter; es enthält Elemente dessen, was auch für die weiteren Übungen in diesem Modul verwendet werden wird. Richten Sie sich innerhalb des Dokumenten-Baums

Ihres Webbrowsers ein Unterverzeichnis für die CG2-Aufgaben ein, und entpacken Sie das Framework in dieses Unterverzeichnis. Die Unterverzeichnisse `lib/`, `cg2-a00-warmup/` und `cg2-a00-test/` sollten nebeneinander im Verzeichnisbaum liegen, also z.B.:



### Wo befindet sich der Dokumenten-Baum des Webservers?

Falls Sie XAMPP installiert haben, können Sie die Dokumente wie folgt ablegen und aufrufen:

- Unter Windows legen Sie die Dateien z.B. im Unterverzeichnis `htdocs/cg2/` des XAMPP-Installationsverzeichnisses ab; dann sind sie unter der URL `http://localhost/cg2/` verfügbar
- Unter Mac OS ist es am einfachsten, die Dateien im Unterverzeichnis `Sites/cg2/` Ihres Benutzerkontos (z.B. `/Users/username`) zu platzieren. Sie können Sie dann unter der URL `http://localhost/~username/cg2` erreichen.

Versuchen Sie nun, die Datei `cg2-a00-test/index.html` über den installierten Webserver zu betrachten; achten Sie darauf, die Seite nicht als lokale Datei (`file://`), sondern tatsächlich über den Server mittels `http://localhost/...` anzuzeigen:

falsch (über lokales Dateisystem)

`file://localhost/Users/hartmut/Sites/cg2/cg2-a00-warmup/index.html`

richtig (über `http://`, das zeigt der Browser allerdings oft nicht an):

`localhost/~hartmut/cg2/cg2-a00-warmup/index.html`

### Achtung Skype-User!

XAMPP streitet sich in manchen Fällen mit Programmen wie *skype* um den Port 80. Falls Sie ein Problem haben, beenden Sie *skype*, starten Sie XAMPP, und starten Sie erst dann wieder *skype* (*skype* ist flexibel und sucht sich dann irgendeinen Weg...).

## Browser-Konfiguration für Firefox:

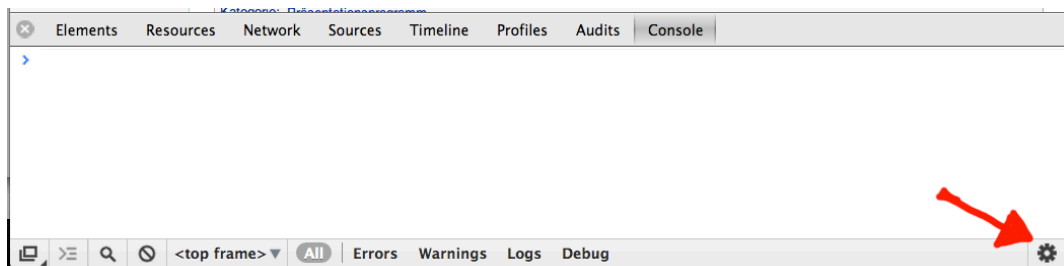
Mit Firefox gab es in letzter Zeit einige Probleme auf manchen Laptop-Rechnern, vor allem unter Windows 7 - wahrscheinlich Fehler in der WebGL-Implementierung des Browsers. Daher empfehle ich folgende Einstellungen für Firefox:

- Geben Sie die URL `about:config` ein (-> Firefox Konfigurationsseiten)
- Geben Sie im Filter-Feld `webgl` ein, um nur die WebGL-Einstellungen zu sehen
- Schalten Sie "Force WebGL" EIN
- Schalten Sie "Prefer Native GL" EIN

*Bitte posten Sie eventuelle Probleme und Lösungen im Moodle-Forum des Kurses!*

## Browser-Konfiguration für Chrome:

Beim Entwickeln mit JavaScript in Chrome gibt es eine kleine Gemeinheit: nach dem Editieren einer Seite wird manchmal nicht der aktuelle Inhalt angezeigt, sondern der Inhalt vor der letzten Änderung. Um dieses sehr ärgerliche Problem beim Entwickeln zu vermeiden, kann der Cache deaktiviert werden. Wenn Sie die JavaScript-Konsole von Chrome öffnen, finden Sie unten rechts den "Zahnrad"-Button



Durch Klick auf diesen Button erscheinen Einstellungen, u.a. auch "disable cache".

## Ausführen der Testseite

Die Testseite `cg2-a00-test/index.html` sollte wie am Anfang dieses Übungsblattes angedeutet zwei Zeichenflächen zeigen; in der linken Fläche befindet sich ein Liniensegment mit zwei kreisförmigen "Anfassern"; in der rechten sollte ein texturierter Würfel rotieren. Die Anfasser des Liniensegments links sollte man mittels Maus-Dragging verschieben können; durch Klicken auf die Checkbox "Anfasser zeigen" sollten die Anfasser ein- und ausblendbar sein.

Falls Sie auf der rechten Seite keinen Würfel sehen, dann öffnen Sie in Ihrem Browser die JavaScript-Konsole bzw. die Konsole mit den Fehlerrausgaben (siehe Screenshots).

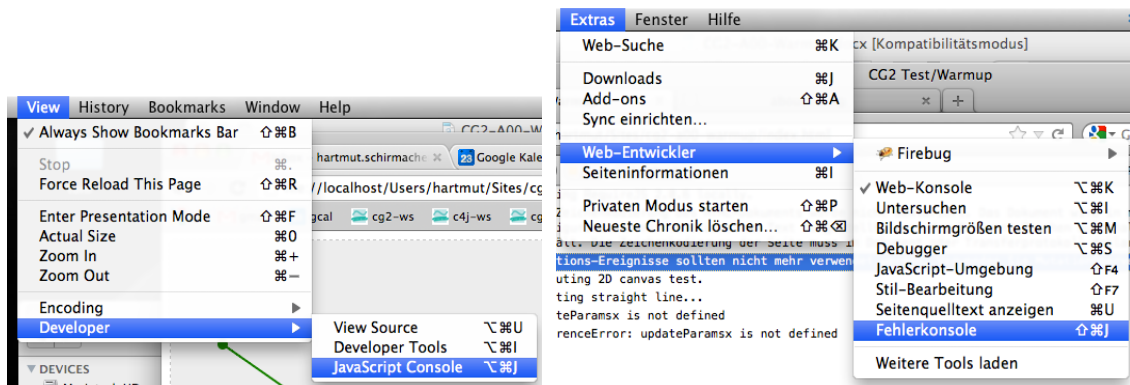


Abb.: Konsole in Chrome bzw. Firefox öffnen

Falls Sie z.B. in Chrome nicht über `http://`, sondern über `file://` die Seite anzeigen, entsteht in der Konsole dieser Fehler:

```
loading RequireJS 2.0.6 locally.
Executing 2D canvas test.
creating straight line...
The File-related APIs seems to exist.
Executing WebGL 3D / texture test.
Uncaught Error: SECURITY_ERR: DOM Exception 18
RENDER WARNING: texture bound to texture unit 0 is not renderable. It maybe non-power-of-2
filtering or is not 'texture complete'
WebGL: too many errors, no more errors will be reported to the console for this context.
```

Die **DOM Exception 18** besagt, dass die Seite versucht hat, auf Ressourcen zuzugreifen, die nicht aus der gleichen Domain kommen (ein sogenannter Cross-Origin-Zugriff). Dieser Fehler kann je nach Betriebssystem und Browser auftreten, wenn man z.B. Texturen lädt und verwendet.

## Entwicklungsumgebung und Editor für JavaScript / WebGL

Als Entwicklungsumgebung genügt der Browser mit seinen eingebauten Logging- und Debugging-Funktionen, sowie ein Texteditor, der in der Lage ist, Syntax-Highlighting für JavaScript durchzuführen. Erwarten Sie nicht zu viel bzgl. automatischer Vervollständigung und ähnlichen aus der Java-Welt bekannten Features, da JavaScript eine sehr dynamische Skriptsprache ist, in der durch sehr "lockere" Typisierung und Namensräume diese Aufgaben etwas erschwert werden. Als kostenlose einfache Editoren gibt es z.B. *Dashcode* (teil von *XCode*) auf dem Mac, *Notepad++* unter Windows, und *gedit* oder *sublime text 2* für Win+Max+Linux. Falls Sie lieber eine komplexere Umgebung verwenden, können Sie z.B. *Aptana Studio*, *Eclipse* oder *Netbeans* verwenden; dies erfordert z.T. jedoch zusätzliche Konfiguration oder Arbeitsschritte - und die Autovervollständigung funktioniert wie bereits erwähnt nicht so gut wie bei Java. Fehler- und Logging-Konsole sind in Firefox getrennt; das Plugin **Firebug** bietet eine deutlich bessere Gesamtdarstellung.

Machen Sie sich mit dem Editor vertraut; bauen Sie z.B. einen Fehler in `main.js` ein, indem Sie den Namen einer Variablen verändern, und laden Sie die Webseite neu; identifizieren und verfolgen Sie den Fehler in der Browser-Konsole und beheben Sie ihn wieder.