**<canvas> - Anwendungen / Beispiele**

Dieser Bereich soll dazu dienen, Beispiele und Anwendungen vorzustellen, in denen das *<canvas>*-Element zum Einsatz kommt.

Weitere Beispiele/Anwendungen, die ich Rahmen meiner Tätigkeiten auf [tutorials.de](http://www.tutorials.de/) zur Verfügung gestellt habe, finden sich [hier](http://playground.quaese.de/tutorials/canvas_test/).

**Reflexion (Klasse *qpReflexion*)**

Um Grafiken und Bilder aufzuwerten, werden sie häufig mit einer Reflexion versehen. Das Orginalbild wird dazu vertikal gespiegelt und nach unten ausgefadet. Um derartiges Verhalten zu realisieren, kann das *<canvas>*-Element verwendet werden. Um dem Internet Explorer gerecht zu werden, kann hier das gleiche durch die Verwendung der **filter**-Eigenschaft umgesetzt werden.

Mit [qpReflexion](http://canvas.quaese.de/index.php?doc_id=67&nav=13,69) wird eine Klasse vorgestellt, die oben beschriebenes Verhalten mit JavaScript umsetzt.

**Analog-Uhr**

Ein Standardbeispiel für Anwendungen mit dem *<canvas>*-Element ist sicherlich die [Analog-Uhr](http://canvas.quaese.de/index.php?doc_id=68&nav=13,70), da hier das Auflösen der eckigen und rechtwickligen Strukturen deutlich wird. Das Beispiel zeigt aber auch, wie Animationen mit dem neuen Element möglich werden.

**Grafik speichern**

Sicherlich gibt es Gelegenheiten, zu denen das gerade gerenderte Bild gespeichert werden muss. Folgender [Workaround](http://canvas.quaese.de/index.php?doc_id=70&nav=13,72) kann dabei helfen.

**Grafik speichern 2 (Ajax, toDataURL)**

Diese [Anwendung](http://canvas.quaese.de/index.php?doc_id=73&nav=13,73) wird nur theoretisch abgehandelt. Sie beschäftigt sich mit der Vorgehensweise, den data:-URL eines <canvas>-Elements mittels Ajax an den Server zu übermitteln, daraus ein Bild zu erzeugen und zu speichern.

**7-Segment-Anzeige**

Diese [Anwendung](http://canvas.quaese.de/index.php?doc_id=71&nav=13,75) zeigt die Ansteuerung einer 7-Segment-Anzeige. Werden mehr Anzeigen kombiniert, können damit Counter, Uhren oder sonstige Elemente erstellt werden, die zur Anzeige von Ziffern und Zahlen benötigt werden.

**Fortschrittsanzeige**

Mit Hilfe von JavaScript (JS) und dem HTML 5 - Element <canvas> wurde eine kreisrunde Fortschrittsanzeige umgesetzt. Die Funktionsweise und Konfigurationsmöglichkeiten werden [hier](http://canvas.quaese.de/index.php?doc_id=74&nav=13,76) vorgestellt.

**Farbfilter**

Die [Anwendung](http://canvas.quaese.de/index.php?doc_id=75&nav=13,77) zeigt, wie mit Hilfe des canvas-Elements und der Methoden **getImageData** und **putImageData** verschiedene Filter auf ein Bild angewendet werden können. Berücksichtigt werden unterschiedliche Sepia-Filter, Graufilter sowie ein Posterize-Filter. Ausserdem ist es möglich, die Sättigung eines Bildes über einen Regler einzustellen.

**Regelmäßíge Vielecke und Polygramme**

Inspiriert durch einen Artikel vom 03. Juni 2009 auf [ajaxschmiede.de](http://www.ajaxschmiede.de/canvas/regelmaessige-vielecke-mit-canvas/) wurde ein [Script](http://canvas.quaese.de/index.php?doc_id=76&nav=13,78) entwickelt, mit dessen Hilfe regelmäßige Vielecke und Polygramme in ein <canvas>-Element gezeichnet werden können.

**Tangram**

Tangram ist ein altes chinesisches Legespiel, das Anfang des 19-ten Jahrhunderts auch in Europa und Amerika Verbreitung fand. Ziel des Spiels ist es, aus den sieben Einzelfiguren ein Quadrat zu legen.

Die Idee, das Spiel in einer [Online-Variante](http://canvas.quaese.de/index.php?doc_id=77&nav=13,79) zur Verfügung zu stellen, wurde aufgrund einer Nachfrage geboren, die mir im Rahmen dieses <canvas>-Projekts gestellt wurde.

**Blur für <canvas>**

Das HTML5-Element <canvas> enthält keine Methode zum Blurren. Soll die Funktionalität hinzugefügt werden, muss hierfür eine eigene Routine implementiert werden.

[Hier](http://canvas.quaese.de/index.php?doc_id=78&nav=13,80) wird eine mögliche Umsetzung vorgestellt.

**Runde Ecken**

Das [hier](http://canvas.quaese.de/index.php?doc_id=79&nav=13,81) vorgestellte Script ermöglicht das Zeichnen von Rechtecken mit abgerundeten Ecken. Es stehen Routinen zum Erstellen gefüllter Elemente sowie zum Zeichnen von Umrisslinien zur Verfügung. Die Benutzung ist an die Methoden [fillRect](http://canvas.quaese.de/index.php?nav=5,27&doc_id=15) und [strokeRect](http://canvas.quaese.de/index.php?nav=5,28&doc_id=16) angelehnt.

**Multiline für <canvas> (jQuery PlugIn)**

Das <canvas>-Element stellt gemäß [HTML5-Spezifikation](http://dev.w3.org/html5/2dcontext/#text) ein **Text**-Modul zur Verfügung. Aktuelle Browser unterstützen das mittlerweile nativ. Für Internet Explorer vor Version 9 und sonstige ältere Browser steht das Script [canvas.text](http://code.google.com/p/canvas-text/) zur Verfügung, das die Funktionalität nachbildet.

Bedauerlicher Weise stellt das Modul lediglich die Funktionalität für **einzeilige** Texte zur Verfügung. **Mehrzeilige** Inhalte bleiben außen vor und müssen selbst implementiert werden.

Mit dem [hier](http://canvas.quaese.de/index.php?doc_id=80&nav=13,82) vorgestellten [jQuery](http://www.jquery.com)-PlugIn dieses Manko behoben. Neben der **Multiline-Funktionalität** werden einige zusätzliche Features wie **Textausrichtung** oder **Drehung** um unterschiedliche Rotationspunkte integriert.