



Universität Regensburg

**Philosophische Fakultät III
Sprach- , Literatur- und Kulturwissenschaften
Institut für Information und Medien, Sprache und Kultur
(I:IMSK)
Lehrstuhl für Informationswissenschaft**

**Seminar URTalking
Modul: INF-Mo7.2
SS 2015
Leitung: Dr. Reischer**

DesignURChar

**Felix Sellmair
1665050
Medieninformatik, Informationswissenschaft
4. Semester B.A.**

**Christian-Jorhan-Str. 24
85465 Langenpreising**

**Email: Felix.Sellmair@stud.uni-regensburg.de
Abgegeben am 09.10.2015**

Inhalt

1	Einleitung	4
2	Die Anwendung ‚DesignURChar‘	5
2.1	Canvas-Controller	6
2.2	UI-Controller	7
2.3	Console-Controller	8
3	Datenbank	10
3.1	PouchDB	10
3.2	Hosting-Service IrisCouch	10
3.3	Database-Controller	10
4	Charaktergalerie ‚Samples‘	11
5	Benutzte Javascript Bibliotheken	12

Abbildungen

Abbildung 1 – Anwendung 'DesignURChar'	5
Abbildung 2 – Auswahlmenü.....	7
Abbildung 3 – Aufruf des virtuellen Konstruktors.....	8
Abbildung 4 – Attribute setzen & zeichnen	8
Abbildung 5 – Charaktergalerie.....	11

1 Einleitung

Viele Anfänger stehen zu Beginn ihres Studiums vor einem Rätsel. Die meisten von ihnen werden zum ersten Mal in ihrem Leben mit informationswissenschaftlichen und –technischen Problemen konfrontiert und sehen sich dadurch zunächst überfordert.

Das Wort ‚Code‘ steht frei übersetzt für etwas, dass schwer zu lesen und zu verstehen ist. Dadurch wirkt es für viele auf den ersten Blick eventuell abschreckend.

Unsicherheit entsteht auch durch mangelnde Kenntnis der Materie. Anders als im technischen Bereich bietet die Informationswissenschaft im Vergleich mit anderen Bereichen wenige Berührungspunkte im Alltag. Jedes Kind weiß, dass ein Haus aus Steinen gebaut wird. Jedes Kind weiß auch, wie ein Stein aussieht. Im Gegensatz dazu ist die Architektur von Computerprogrammen für viele Menschen ein Mysterium.

Diese Anwendung soll einen Einblick in das Vokabular und die Syntax der Programmiersprache Java geben und so Studienanfänger an diese heranführen.

2 Die Anwendung ‚DesignURChar‘

Die Anwendung ‚DesignURChar‘ bietet die Möglichkeit mit einer primitiven Zeichenfläche einen persönlichen Charakter zu zeichnen und diesen in anderen URGaming-Spielen zu verwenden.

Zusätzlich generiert die Web-App aus jeder Aktion eines Benutzers fiktiven Java-Quellcode und gibt diesen in einer Konsole aus. So soll bei Studienanfängern mit informationswissenschaftlichem Hintergrund ein Gefühl für Quellcode im Allgemeinen und die Java-Syntax im Speziellen geschaffen werden.

Zum Zeichnen kann der Nutzer aus drei verschiedenen geometrischen Formen (Kreis, Rechteck, Stern) und einem Pinsel wählen. Die Ausgabe passt sich in Echtzeit dem gewählten Werkzeug an. Im Hintergrund werden alle gezeichneten Charaktere in einer Datenbank gespeichert und in einer Galerie am linken Bildschirmrand angezeigt.

Diese Charaktergalerie ist auf jeder UR-Gaming Seite vorhanden und gibt dem Nutzer in einem späteren Spiel die Möglichkeit, seinen persönlichen Avatar auszuwählen. In der folgenden Abbildung sehen Sie die Anwendung ‚DesignURChar‘ mit einer Avatargalerie (links), einer Zeichenfläche (Mitte), einer Ausgabekonsole (unten) und einem Menü (rechts).

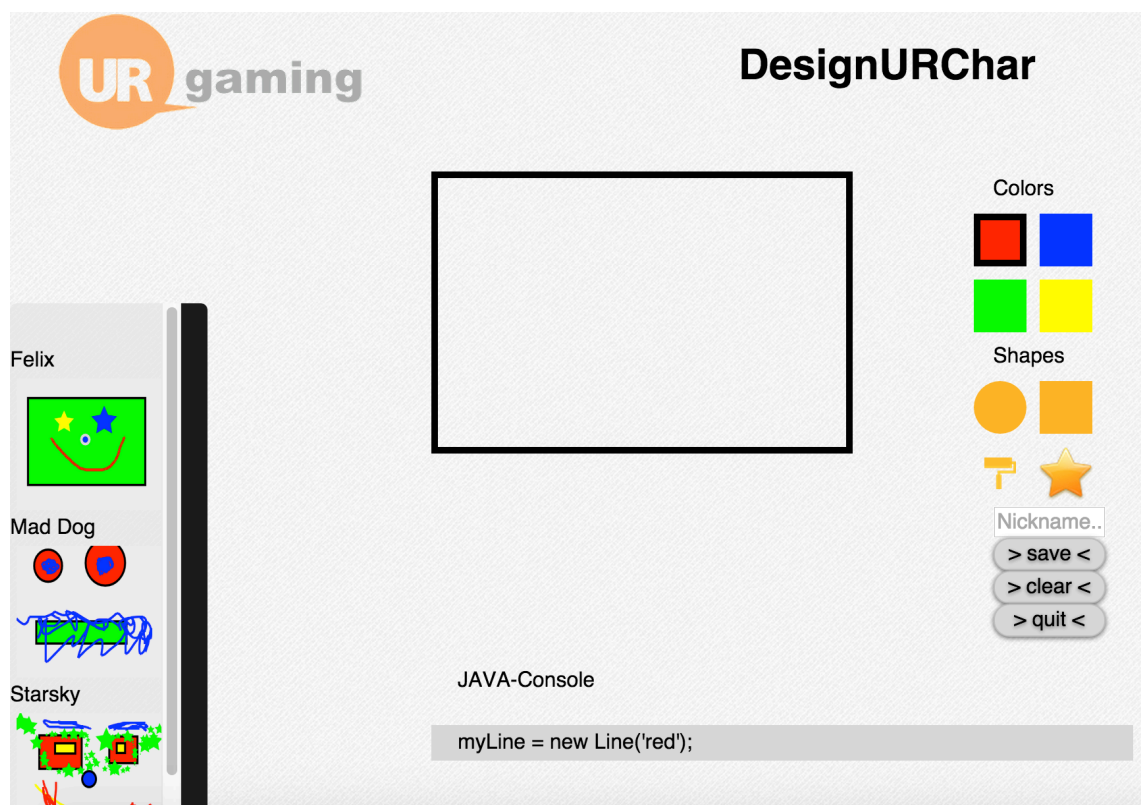


Abbildung 1 – Anwendung 'DesignURChar'

2.1 Canvas-Controller

Der Canvas-Controller verwaltet alle Aktionen, die sich innerhalb der Zeichenfläche („Canvas“) abspielen. Hier werden die entsprechenden Objekte gezeichnet.

Zunächst werden alle Mausbewegungen innerhalb des Canvas abgefangen. Diese wirken sich auf Größe und Position der gezeichneten Objekte aus.

Außerdem verfügt der Canvas-Controller über Instanzvariablen, in denen die Farb- und Formauswahl des Nutzers gespeichert werden. Durch entsprechende Setter-Methoden werden diese Variablen bei jeder Veränderung der Auswahl aktualisiert.

Weitere Bestandteile sind die vier verschiedenen Zeichenmethoden für vier verfügbare Nutzeroptionen:

- `drawLine()`: zeichnet eine Linie auf die aktuelle Mausposition.
- `drawRect()`: zeichnet ein Rechteck.
Position, Breite und Höhe abhängig von Mausposition und `-drag`.
- `drawCircle()`: zeichnet einen Kreis.
Position und Radius abhängig von Mausbewegung und `-drag`.
- `drawStar()`: zeichnet einen Stern.
Position und Größe abhängig von Mausbewegung und `-drag`.

2.2 UI-Controller

Der UI-Controller verwaltet alle Nutzereingaben außerhalb der Zeichenfläche und gibt diese an die entsprechenden Module der Anwendung weiter.

Im Menü rechts der Zeichenfläche verfügt der Nutzer über verschiedene Optionen bezüglich Farbgebung und Formenauswahl. Außerdem kann das Bild hier gespeichert oder gelöscht werden. Für das Speichern eines Bildes und die spätere Zuordnung ist ein Nutzernamen notwendig. Durch Klick auf den Button ‚quit‘ gelangt man zurück zur UR-Gaming Startseite.

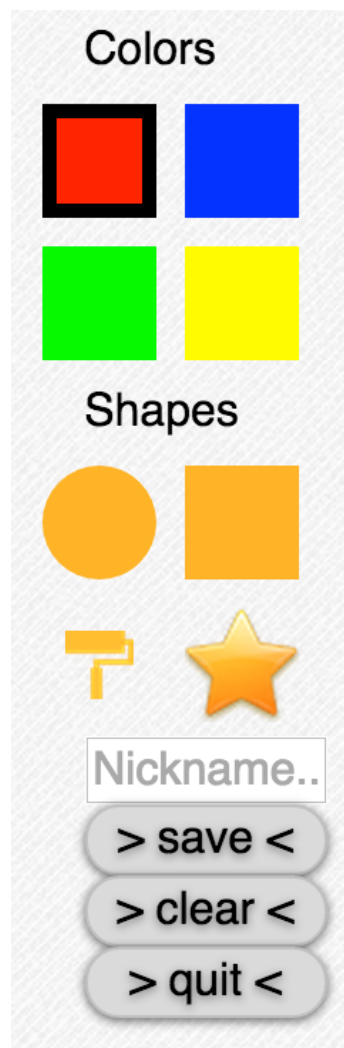


Abbildung 2 – Auswahlmenü

2.3 Console-Controller

Der Console-Controller ist für die Ausgabe des Java-Codes zuständig. Sobald der Nutzer eine Änderung innerhalb der Zeichenfläche oder des Menüs vornimmt wird der entsprechende Quellcode erzeugt. Dieser Code ist von Form, Farbe, Position und Größe des gezeichneten Objekts abhängig.

Die Auswahl von Form und Farbe des Objekts entsprechen dem Aufruf eines Konstruktors. Ein virtuelles, noch unsichtbares Java-Objekt (im folgenden Beispiel ein blauer Kreis), wird erzeugt und kann nun verändert werden.

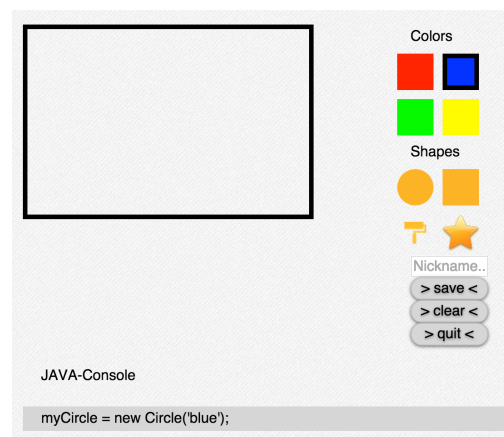


Abbildung 3 – Aufruf des virtuellen Konstruktors

Größe und Position eines Objekts werden anschließend durch entsprechende Mausebewegungen definiert. Die Java-Konsole ordnet diese Parameter mithilfe von Setter-Methoden dem virtuellen Objekt zu. Im Anschluss daran wird das Objekt gezeichnet.

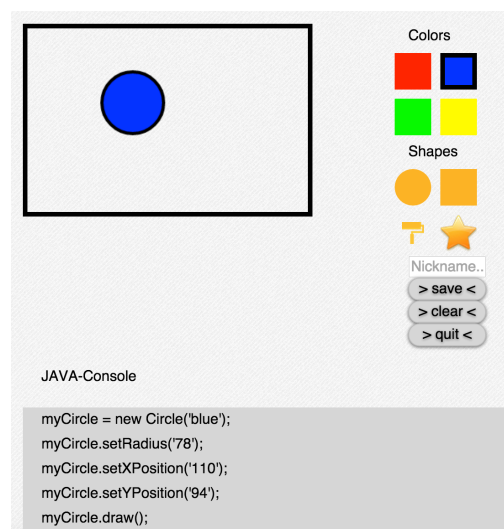


Abbildung 4 – Attribute setzen & zeichnen

Eine Übersicht über die je nach Kontext unterschiedlichen Konsolenausgaben finden Sie in dieser Tabelle:

Aktion	Ausgabe
Zeichnen eines Rechtecks	<pre>myRect = new Rect('color'); myRect.setWidth('width'); myRect.setHeight('height'); myRect.setXPosition('x-pos'); myRect.setYPosition('y-pos'); myRect.draw();</pre>
Zeichnen eines Kreises	<pre>myCircle = new Circle('color'); myCircle.setRadius('radius'); myCircle.setXPosition('x-pos'); myCircle.setYPosition('y-pos'); myCircle.draw();</pre>
Zeichnen eines Sterns	<pre>myStar = new Star('color'); myStar.setSize('size'); myStar.setXPosition('x-pos'); myStar.setYPosition('y-pos'); myStar.draw();</pre>
Zeichnen einer Linie	<pre>myLine = new Line(); myLine.setColor('color'); myLine.drawAt('x-pos', 'y-pos');</pre>

3 Datenbank

3.1 PouchDB

Das permanente Speichern von Avataren geschieht mithilfe von PouchDB.

PouchDB ist eine open-source JavaScript-Datenbank und basiert auf Apaches dokumentenorientierter Datenbank CouchDB.

PouchDB ist für Webanwendungen optimiert und besticht einerseits durch einfache Implementierung, andererseits durch sehr schnelle Serveranfragen.

3.2 Hosting-Service IrisCouch

Die verwendete PouchDB läuft auf dem Server von IrisCouch. Dieser kostenlose Hosting-Dienst ersetzt einen CouchDB-Server. Zusätzlich stellt IrisCouch CouchDB's ‚Futon‘ zur Verfügung, ein natives User-Interface, dass die Administration der Datenbank erleichtert.

Hier finden Sie die URL zur Datenbank und die Zugangsdaten zu einem Administratorkonto:

URL:	http://urgaming.iriscouch.com:5984/_utils/
Administrator:	urgadmin
Passwort:	adminpw

3.3 Database-Controller

Der Database-Controller ist für die Kommunikation der Anwendung URGaming mit der Datenbank zuständig. Er legt neue Charaktere in der Datenbank ab und liest diese wieder aus, wenn neue Informationen verfügbar sind. Im Anschluss daran gibt er die Daten an die Galerien aller vier UR-Gaming-Spiele weiter.

In der Methode `saveImage()` wird aus dem gezeichneten Bild und dem eingegebenen Nutzernamen ein JSON-Objekt erzeugt, dass dann als Dokument in der Datenbank abgelegt wird.

4 Charaktergalerie ‚Samples‘

Die Galerie aller bisher erstellten Usercharaktere befindet sich bei jedem UR-Gaming-Spiel auf der linken Seite und wird bei jeder Daten. Durch Klick auf eines der Bilder wählt sich der Spieler seinen Charakter aus. Hierbei ist jeder erstellte Avatar auch für jeden Spieler verfügbar. Zur Formatierung der Bilder wird das Framework ‚underscore.js‘ verwendet, welches Templates zum Zusammensetzen von HTML5-Elementen unterstützt.

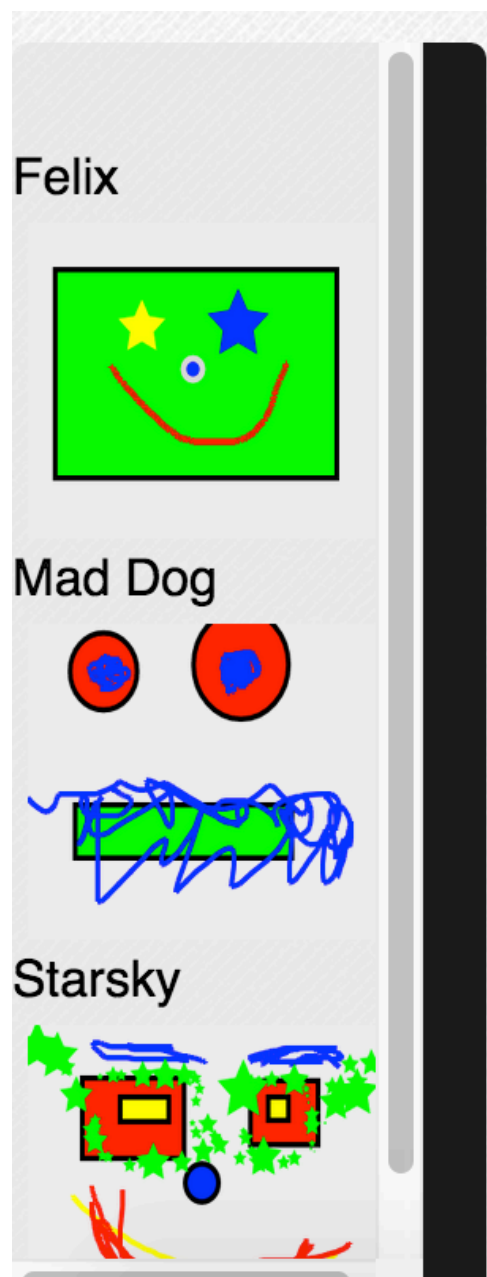


Abbildung 5 – Charaktergalerie

5 Benutzte Javascript Bibliotheken

- JQuery (Version 2.1.4)
- PouchDB (Version 4.0.1)
- Underscore.js (Version 1.8.3)

