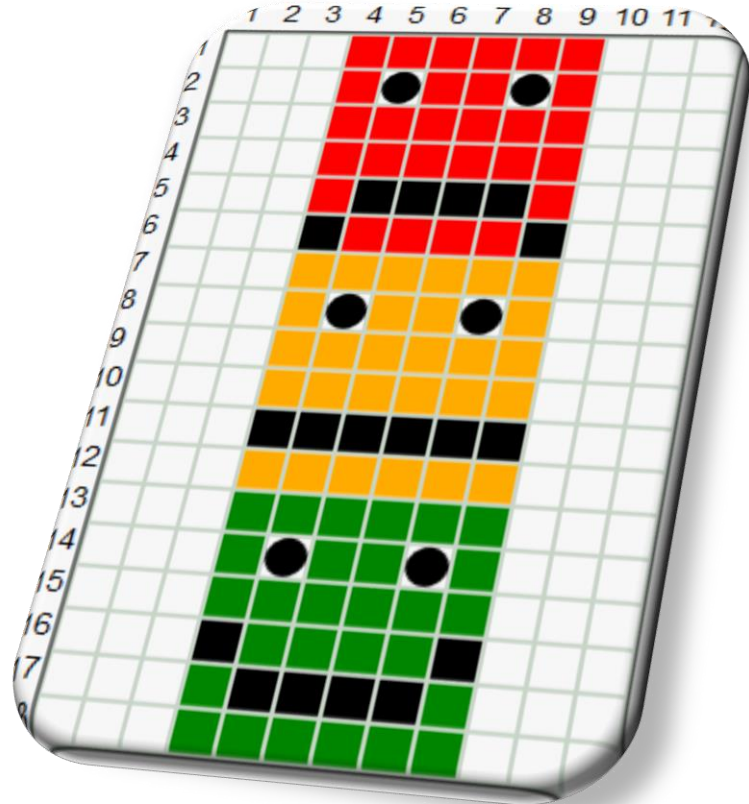


# Mosaics Game

Martin Heller, Sebastian Sopp, Farah Tahir

# Agenda

1. Einleitung
2. Demo
3. Umsetzung
4. Ausblick



# Einleitung

## Mosaikspiel mit SVGs als Webseite



# Einleitung

## Mosaikspiel mit SVGs als Webseite

- Mosaikelemente
  - Kreis,
  - Quadrat (und Rechteck),
  - Dreieck,
  - Linie
- Befehle (und Parameter)
  - Parsen
  - Validieren
- Mustervergleich (Validierung)
- XMLs + Schema
  - Speicherung der Highscores
  - Katalogisierung der SVGs
  - Auslagerung der Hinweise

# Einleitung

## Mosaikspiel mit SVGs als Webseite

- Editor
  - Mustergöße auswählen
  - Muster erstellen
  - Muster speichern
- Spiel
  - Muster auswählen
  - Spielen
  - Gewinnen
- Highscores
- Administration
  - Muster freigeben
  - Mustereigenschaften editieren
  - Muster löschen
- Dokumentation

Demo

# Umsetzung

1. Syntax, Parsen, Parameter- und Muster-Validierung
2. SVG-Dateien abspeichern und verwalten
3. Syntax-Katalog und SVG-Selektor
4. Highscoreliste

# Umsetzung

## Syntax

- Befehl: <Befehlsname>(<Parameter1>,...<ParameterN>)
  - Befehlsnamen: square, circle, rectangle, line, triangle und clearcell.
- Parameter: <Position>, <Breite/Höhe>, <Punkt> und <Farbe>
  - Breite/Höhe = Integer
  - Farbe = HTML-Farbname, z.B. black oder fuchsia
  - Position <Spalte>,<Zeile>
    - *Spalte/Zeile: <Startposition>[[:<Schrittweite>]:<Endposition>]]*
      - Start-/Endposition und Schrittweite = Integer
  - Punkt: <Horizontalankerpunkt><Vertikalankerpunkt>
    - *Horizontal: "L" (Left), "C" (Center) und "R" (Right).*
    - *Vertikal: "T" (Top), "C" (Center) und "B" (Bottom).*



# Umsetzung

## Parsen

1. Befehlsname  
Switch/Case mit Befehlsname (Rückgabe: Parameteranzahl)
2. Klammern  
Mit `String.indexOf()`
3. Parameteranzahl  
Mit gleichem Switch/Case

Entsprechende Ausgabe bei Fehlern

("Unknown command", "Invalid number of parameters (required x, got y).", etc.)

# Umsetzung

## Parameter-Validierung

### 1. Position

- Prüfen, ob...
  - *Integer*
  - *innerhalb der Grenzen*
  - *Schrittweite > 0, wenn StartPos < EndPos und umgekehrt*

### 2. Punkt

- Prüfen, ob...
  - *genau zwei Ankerpunkte angegeben wurden*
  - *jeweils ein Horizontal- und Vertikalankerpunkt angegeben wurde*

# Umsetzung

## Parameter-Validierung

### 3. Linie

- Prüfen, ob Punkte verschieden (via berechnete Offsets)

### 4. Dreieck

- Prüfen, ob Fläche  $\neq 0$  mit 2D-Vektorrechnung

### 5. Farbe

- Prüfen, ob in Array mit gültigen Farben vorhanden

# Umsetzung

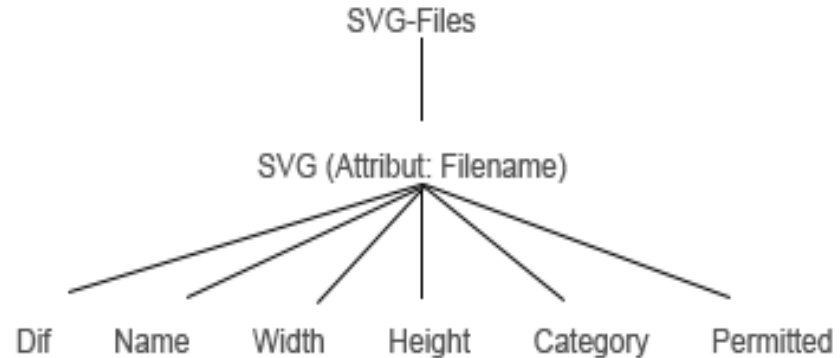
## Muster-Validierung/Gewinnermittlung

1. Prüfen, ob Anzahl der Mosaikelemente gleich
  2. Je Mosaikelement:
    1. Prüfen, ob im Muster an gleicher Position ein Element ist
    2. Prüfen, ob Anzahl der Attribute gleich
    3. Je Attribut:
      - *Prüfen, ob Name und Wert gleich (außer bei ID-Attribut)*
- Bei erstem negativen Ergebnis wird abgebrochen
  - Bei Erfolg: Gewinn-Meldung, Highscore-Liste

# Umsetzung

Speichern und Katalogisieren der SVG Grafiken

- XML für Katalog aller SVGs
- Suchkriterien
- Metadaten



# Umsetzung

## Speichern und Katalogisieren der SVG Grafiken

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="SVGFiles" type="SVGFileType"/>

  <xsd:complexType name="SVGFileType">
    <xsd:sequence>
      <xsd:element name="SVG" type="SVGType" maxOccurs="unbounded" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="SVGType">
    <xsd:sequence>
      <xsd:element name="Name" type="xsd:string"/>
      <xsd:element name="Category" type="xsd:string"/>
      <xsd:element name="Width" type="xsd:positiveInteger"/>
      <xsd:element name="Length" type="xsd:positiveInteger"/>
      <xsd:element name="Dif" type="xsd:string"/>
      <xsd:element name="Permitted" type="xsd:string"/>
    </xsd:sequence>
    <xsd:attribute name="Filename" type="xsd:string" use="required" />
  </xsd:complexType>

</xsd:schema>
```

# Umsetzung

## Speichern und Katalogisieren der SVG Grafiken

```
<?xml version="1.0"?>
<SVGFiles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../xml/SVG_index.xsd">
  <SVG Filename="America.svg">
    <Name>America</Name>
    <Category>Countries and Flags</Category>
    <Width>18</Width>
    <Length>12</Length>
    <Dif>Normal</Dif>
    <Permitted>true</Permitted>
  </SVG>
  <SVG Filename="Germany.svg">
    <Name>Germany</Name>
    <Category>Countries and Flags</Category>
    <Width>18</Width>
    <Length>12</Length>
    <Dif>Easy</Dif>
    <Permitted>true</Permitted>
  </SVG>
</SVGFiles>
```

# Umsetzung

## Speichern und Katalogisieren der SVG Grafiken

- Erste lokale Überprüfungen (Attribute, RegExp)
- Senden der Daten per AJAX an PHP-Script auf Server
- Serverseitige Überprüfung auf duplizierten Namen
- Child-Knoten Eintragen und SVG Speichern



# Umsetzung

## Verwaltung der SVG Grafiken

- Passwortgeschützte Admin-Seite
- Freigabe von SVGs
- Löschen, Umbenennen, Kategorie und Level ändern

# Umsetzung

## Syntaxkatalog und SVG-Selektor

- Syntaxkatalog-Daten als XML
- Anzeige in Akkordions
- Vorschau der SVGs durch Auslesen der XML
- Filtern nach Kategorie & Schwierigkeitsgrad

# Umsetzung

## Highscoreliste

- nur Top 10 werden gespeichert
- kein doppelter Eintrag des Usernamens
- Verbesserung des Spielers übernehmen
- Sortieren der Top 10 bei der Anzeige

Ausblick

# Ausblick

- Erweiterung des Spiels auf Zeit
- Evaluation (Benutzerfreundlichkeit, Spaßfaktor)
- Lokalisierung
- Veröffentlichen