

XML Technology - WS19/20 - BlackJack Project - Technische Universität München (TUM)

Mr. Ikbal Yesiltas

Mr. Lorenzo Brazzi

Mr. Markus Zuber

Mr. Patrick Reto

XML Technology - WS19/20 - BlackJack Project - Technische Universität München (TUM)

Mr. Ikbal Yesiltas
Mr. Lorenzo Brazzi
Mr. Markus Zuber
Mr. Patrick Reto

Table of Contents

1. Design Choices	1
Scalable Vector Graphics of Game Elements	1
Cards	1
Table	1
Chips	1
XSLT Processor	1

Chapter 1. Design Choices

Our Implementation covers all mandatory Rules and Structures of the classic Blackjack game and is leaned towards the exercise sheets provided by the Instructors.

Scalable Vector Graphics of Game Elements

Cards

We implemented the cards as XML objects containing 3 elements (hidden, color, and value). The set is the classic 52-piece french deck.

First, we created each individual symbols and inserted them in a card template. This template covers a number at the upper right corner, which we mirrored on both axes to get the impression of a more realistic card. For the special cards, J, Q, K, we used images from the web. Lastly, we extended our implementation by multiple symbols for higher numbers. This could be achieved very straightforwardly, as a number usually is built upon and complements a lower number, e.g. the 4 uses two 2s.

To reduce redundancy, we used plenty of defs and use, with which also the XSLT-processor can create cards dynamically.

Table

The table is a rather simple but intuitive round blackjack table, achieved by using clip-paths.

Chips

For the currency in the game, we used generic blackjack chips, i.e. 1,5,25,100,500, and 1000. In addition to that, we colorized each value relative distinctively, so that the players can immediately/intuitively see which coin to choose.

XSLT Processor

One single xsl file is responsible for the transformation of all SVGs. This Processor adjusts the table, cards, chips, etc. dynamically, and returns an updated html file, as soon as the database is changed by the course of the game.

The Processor waits for an Input, the main game xml file, and transforms each element into correct SVGs by using multiple nested xsl:choose and xsl:when instructions. Hence, all SVG definitions are also contained in this file.