X Stack Multiclient Blackjack

Janik Schnellenback <janik.schnellbach@tum.de>

Maximilian Karpfinger <maximilian.karpfinger@tum.de>

Felix Hennerkes <ga38hom@mytum.de>

Daniel Meint <d.meint@tum.de>

Table of Contents

Introduction

Current XML technologies provide a full stack of modeling languages, implementation languages, and tools for web applications that is stable, platform independent, and based on open standards. A particularly strong point of what we call the X stack is that data are encoded with XML end-to-end and that XML technologies can be used whereever XML data need to be processed. Combining principles and proven practices from document and software engineering, this paper documents architecture, modeling techniques, and implementation strategies for the simple browser game Guess the Number (GN).

Description of the case study

We implement the popular casino game Blackjack as a multi-client web application. Globally, there are a variety of rulesets with slight differences. We seek to present the most universally accepted variation, commonly found in Las Vegas casinos.

Up to five *players* compete not against each other, but separately against the *dealer*. The objective of each player is to draw cards and maximize the sum of their respective values (*hand value*) without exceeding a sum of 21 (*bust*). Players win by achieving one of the following final game states:

- A hand value of 21 with only two cards (*blackjack*)
- A value higher than dealer's without exceeding 21
- A value less than 21 while the dealer busts

The game is played with a single deck of french playing cards. Number cards are worth their value, e.g. the seven of hearts is worth seven points, face cards (Jack, Queen, and King) are worth ten points, and an Ace can be counted as either one or eleven, depending on what is more favorable in a specific situation. A hand containing an Ace counted as eleven is referred to as a *soft* hand, because the value of the Ace will change to a one to prevent the player from busting if he was to draw another card and otherwise exceed 21. A card's suit is irrelevant in Blackjack.

Preparation

Before the actual playing begins, players place their bets. Our version uses US-dollars as currency and does not limit players in the amount they want to bet. It does, however, prohitibt them from playing without betting at all. We will refer to this stage of the game as the *betting phase* throughout the following sections.

Course of a round

At the beginning of the *playing phase* each player is dealt two cards face up. The dealer receives one exposed card that everyone can see and one hidden card.

The dealer subsequently asks each player, going clockwise around the table, whether they want to improve their respective hand by drawing additional cards. Each player has the following options:

Table 1.

Action	Description	Condition
Stand	Ends the player's turn	Always possible
Hit	The player receives another card from the deck	Availble if the player's hand value is less than 21
Double	The player doubles his bet and receives exactly one more card before finishing his turn	Available if the player only has his two initial two cards
Insurance	The player puts a side-bet worth half his initial bet on the outcome that the dealer has a Blackjack	Available if the first card of the dealer is an Ace

If a player busts, they lose the round and have to pay their bet to the dealer

After all players have finished their turn, the dealer players in a predetermined manner: He draws cards as long as his hand is worth less than 17 points and must stand on a soft 17 or better.

Architecture

The component Model

Modeling the component Model

Implementing the component Model

The component View

Modeling the component View

Implementing the component View

The component Controller

Modeling the component Controller

Implementing the component Controller

Conclusions

Bibliography

[A14] Adamkó, Attila. "Internet Tools and Services". Lecture notes. https://gyires.inf.unideb.hu/GyBITT/08/index.html. Last accessed on 2019 February 20.

[BaseX] BaseX homepage. http://basex.org. Last accessed on 2019 February 25 .

[B16] Brüggemann-Klein, Anne. "The XML Expert's Path to Web Applications: Lessons learned from document and from software engineering." Presented at XML In, Web Out: International Symposium on sub rosa XML, Washington, DC, August 1, 2016. In Proceedings of XML In, Web Out: International Symposium on sub rosa XML. Balisage Series on Markup Technologies, vol. 18 (2016). https://doi.org/10.4242/BalisageVol18.Bruggemann-Klein01.

[BD13] Brügge, Bernd and Allen H. Dutoit. "Object-Oriented Software Engineering Using UML, Patterns, and Java." Pearson 2013 (Kindle Edition).

[BMRSS00] Buschmann, Frank, Regine Meunier, Hans Rohner, Peter Sommerlad, and Michael Stal. "Pattern-Oriented Software Architecture, A System of Patterns." Wiley 2000 (Kindle Edition).

[E03] Eric Evans. "Domain-Driven Design: Tackling Complexity in the Heart of Software." Addison-Wesley 2003 (Kindle Edition).

[F06] Fowler, Martin. "Development of Further Patterns of Enterprise Application Architecture." https://www.martinfowler.com/eaaDev/. Last accessed on 2019 February 20.

[F12] Fowler, Martin. "Patterns of Enterprise Application Architecture." Addison-Wesley 2012 (Kindle Edition).

[G06] Goetz, Brian. "Java Concurrency in Practice." Addison-Wesley 2006 (Kindle Edition).