Dokumentation - Aufgabe 3

Gruppe: HTML Forms

Table of Contents

Design

Bei Blackjack handelt es sich um ein Kartenspiel, das mit sechs französischen Decks gespielt wird. Bis zu sieben Spieler spielen gegen die Bank. Ziel der Spieler ist es, mit zwei oder mehr Karten mehr Punkte als der Croupier zu erreichen, ohne 21 Punkte zu überschreiten. Umgekehrt ist es das Ziel des Croupiers, mindestens so viele Punkte wie jeder Spieler zu erreichen, ohne 21 Punkte zu überschreiten.

Konzeptdesign

Zu Beginn tätigt jeder Spieler seine Einsätze. Neben einem Mindesteinsatz, wie bei Poker, gibt es zudem ein Limit, also einen Maximaleinsatz. Jeder Spieler tätigt zunächst den Einsatz in seiner eigenen Box. Anschließend ist auch ein Einsatz in fremde Boxen möglich. Jeder Spieler erhält dann zwei Karten, der Croupier - der Vertreter der Bank - eine. Zieht ein Spieler hierbei ein Ass und eine 10 bzw. Ass und König/Dame/Bube, so hat er einen Blackjack. Zieht hingegen der Croupier als erste Karte ein Ass, so haben die Spieler die Möglichkeit durch Setzen der Hälfte ihres regulären Einsatzes als Insurance-Einsatzes auf die Insurance-Line auf das anschließende Ziehen einer 10 / eines Buben / einer Dame / eines Königs durch den Croupier zu wetten. Gewinnen die Spieler die Wette, so erhalten sie das Doppelte ihres Insurance-Einsatzes als Gewinn, andernfalls verlieren sie diesen Einsatz. Der Reihe nach kann dann jeder Spieler so lange weitere Karten ziehen (Take), bis er exakt 21 Punkte erreicht, 21 Punkte überschreitet, oder freiwillig seinen Zug beendet. Vor jedem Take hat ein Spieler die Möglichkeit seinen Einsatz zu verdoppeln; er zieht anschließend noch genau eine Karte. Erreicht ein Spieler mit mehr als zwei Karten 21 Punkte, so hat er lediglich den maximal erreichbaren Punktestand, aber keinen Blackjack erreicht. Zudem kann ein Spieler, der über doppelte Hände verfügt (Karten liegen doppelt vor), seine Hand in zwei Hände aufspalten (Split); für die zweite Hand ist derselbe Einsatz wie für die erste Hand zu tätigen. Ein Spieler mit mehreren Händen zieht für jede Hand Karten, es handelt sich aber nur dann um eine Sieg, wenn der Spieler mit beiden Händen gewinnt, und nur dann um eine Niederlage, wenn der Spieler mit beiden Händen verliert; in allen anderen Fällen handelt es sich um ein Unentschieden. Haben alle Spieler und der Croupier ihren Zug getätigt, endet die Runde.

Jeder Spielteilnehmer mit der höchsten Punktezahl kleiner gleich 21 gewinnt. Ein Blackjack ist hierbei höher zu gewichten als 21 Punkte die mit mehr als zwei Karten erreicht wurden. Gewinnt lediglich der Croupier, verlieren sämtliche Spieler ihren Einsatz, gewinnt ein einzelner oder gewinnen mehrere Spieler, so gewinnen sie den Gegenwert ihres Einsatzes und erhalten ihren Einsatz zurück. Gewinnen sowohl ein oder mehrere Spieler, als auch der Croupier, so erhalten diese Spieler ihren Einsatz zurück, die anderen Spieler verlieren ihren aber.

Implementierungsdesign

- Das Programm beginnt im Hauptmenü. Ein Neues Spiel kann gestartet oder ein zuvor begonnenes kann wiederaufgenommen werden.
- Die Spieler können Einsätze in Fünferschritten zwischen 5 und 100 Währungseinheiten tätigen.

- Der Croupier zieht bis zu einem Punktestand von 16 immer eine Karte (Take) und beendet ab einem Punktestand von 17 stets seinen Zug (Stand).
- Wir verzichten auf Ausnahmeregeln beim Split.
- Wir verzichten auf Varianten-Regeln.

Implementierung

Zustandsausgabe mit XSLT

Grundlage: SVG

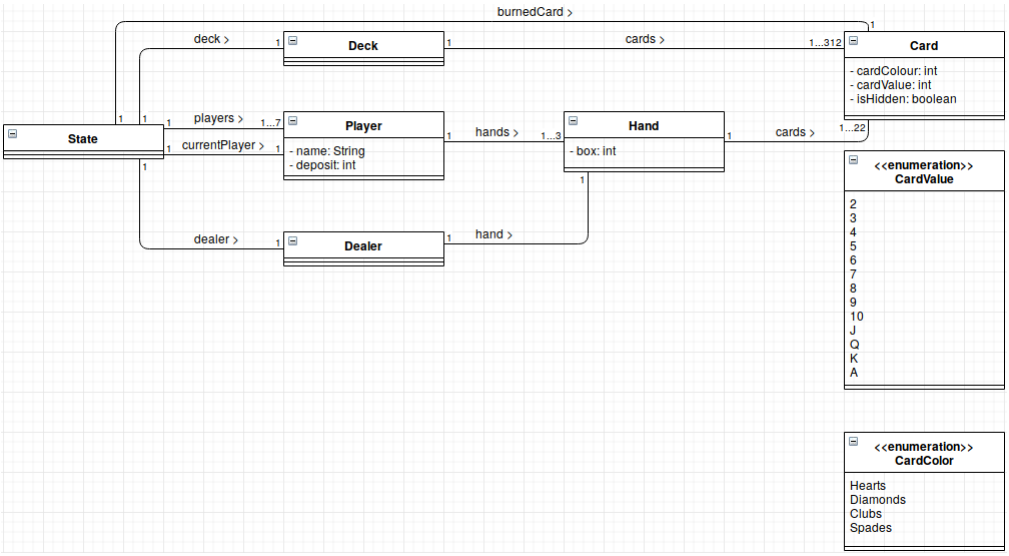
Bei Aufgabenblatt 2 haben wir einige SVGs geschrieben, eine für den Spieltisch, eine für Chips, die anderen für die Karten. Genauer haben wir die Grundelemente der Karten gemalt: Buchstaben A / J / Q / K, Zahlen von 2 bis 10, die vier Farbsymbole, und den Hintergrund.

Bei Aufgabenblatt 3 haben wir diese SVG-Dateisfolder weiterentwickelt, bzw. bauen wir jetzt eine SVG-Template für Karten. Jede Karte besteht aus drei Komponenten, erstens aus dem Hintergrund, zweitens aus der Kartenfarbe und zuletzt aus dem Kartenwert.

Konzept: UML

Um das Spiel übersichtlich darstellen zu können, wird das folgende UML-Diagramm dargestellt, wobei die einzelnen Daten folgendermaßen aussehen:

Figure 1. UML-Diagramm



Implementierung: DTD & XML

Für die Implementierung vom XMl wurde das DTD basend auf das UML Diagram so strukturiert , dass es 6 Elemente enthält und die sind : State , Deck , Dealer , Player , Hand und Card . Jedes Element hat auch einige Attribute , die sind alle mit dem Word Required deklariert . Durch die Angabe von Required müssen die Type im XML angegeben werden und mit CDATA ist Charakter-Daten gemeint.

State

Dieses Element enthält 4 Element-Content (deck , card , dealer und player +). deck bezeichnet das Kartendeck , card ist die Burned karte , mit dealer ist der Croupier gemeint und player + bedeutet ,

dass mindestens ein spieler auf dem Tisch sitzt . Das Element State hat ein Attribut currentPlayer , das zu dem Player referenziert , der gerade spielt.

Deck

Dieses Element enthält 1 Element-Content (card*) . card* bedeutet , dass das Deck aus mehreren karten besteht.

Dealer

Dieses Element enthält 1 Element-Content (card*). card* bedeutet , dass der Dealer mehrere karten hat.

Player

Dieses Element enthält 1 Element-Content (hand+). hand+ bedeutet , dass der Player mindestens ein Hand hat . Das Element Player hat 3 Attribute (ID , name , deposit). ID ist für die Referenzierung zuständig , name bezeichnet den Spielername und mit deposit ist der Balance gemeint.

Hand

Dieses Element enthält 1 Element-Content (card*). card* bedeutet , dass jeder Hand aus mehreren Karten besteht.

Card

Dieses Element enthält kein Element-Content und bei EMPTY ist gemeint ,dass keine Charakter-Daten eingegeben müssen. Das Element card hat 3 Attribute(cardColour , cardValue , isHidden). isHidden bezeichnet , ob die karte verdeckte ist.

Konvertierung: XSL

Wir verwenden sechs Templates für die Konvertierung der fünf einzelnen Elemente, State, Deck, Dealer, Player und Card. Für Card werden zwei verschiedene Templates verwendet. Insbesondere wurde bei der Formatierung mittels Inline-CSS auf ein responsives Layout geachtet, d.h. die Element skalieren mit der Größe des Browserfensters.

State

Im zugehörigen Template wird das Grundgerüst des resultierenden HTML-Dokuments gelegt. Der Spieltisch wird als Hintergrundgrafik eingebunden und weitere Templates werden aufgerufen.

Deck

In diesem Template werden lediglich drei Kartenrückseiten als SVGs zur Repräsentation des Kartendecks ausgegeben.

Dealer

In diesem Template wird lediglich das Template für die Karten des Dealers aufgerufen.

Player

Zunächst werden in diesem Template der Name und das Vermögen des jeweiligen Spielers ausgegeben. Anschließend werden bis zu drei Hände ausgegeben. Die Ausgabe für jede einzelne Hand setzt sich aus der Ausgabe der zugehörigen Wette und einem Aufruf des Templates für die Karten des Spielers zusammen.

Card

Wir haben je ein XSLT-Template für die Generierung der Karten des Spielers und des Dealers zu unserer XSL-Datei hinzugefügt. Das Template für den Dealer unterschiedet sich vom Template für den

Spieler dahingehend, dass es eine Fallunterscheidung nach verdeckten und offenen Karten trifft. Für erstere wird lediglich ein Kartenhintergrund ausgegeben. Letztere werden ebenso behandelt wie es mit den Karten des Spielers geschieht: Aus Basis der Attributwerte des zugrundeliegenden XML-Elements vom Typ "card" für Kartenfarbe und Kartenwerte werden Pfade erzeugt und somit SVGs schrittweise zu einer vollständigen Karte zusammengesetzt. Hierdurch kann Code-Redundanz reduziert werden, da nicht für jede einzelne Karte ein separates Codesegment zur Generierung erforderlich ist, sondern sämtliche Karten durch ein einzelnes Codesegment generiert werden können.