

Dokumentation
zur Studienarbeit im Fach
Praktikum Softwareengineering

- Entwicklung einer Lernsoftware -
”myMemo”

von Susanne Kießling
Dozent: Prof. Dr. Martin Thost

1. Juli 2013

Inhaltsverzeichnis

1	Anforderungen des Kunden	3
1.1	Systemvoraussetzungen	3
1.2	Zielgruppe	3
2	Lastenheft	4
2.1	Zielbestimmung	4
2.2	Produkteinsatz	4
2.3	Produktfunktionen	4
2.4	Produktdaten	5
2.5	Produktleistungen	5
2.6	Qualitätsanforderungen	5
2.7	Ergänzungen	5
3	Aufwandskalkulation	6
3.1	Function-Point-Analyse	6
3.2	6
4	Use Case Diagramme	8
4.1	Spieler verwalten - \LF10\,\LF11\,\LF12\	8
4.2	Anzahl der Spieler wählen - \LF20\	9
4.3	Thema wählen - \LF30\	10
4.4	Spielfeldgröße wählen - \LF40\	11
4.5	Spiel starten - \LF50\	12
4.6	Highscore - \LF60\,\LF61\	13
4.7	Vokabeltraining - \LF70\	14
4.8	Audiodaten abspielen - \LF80\	15
5	Use Case Beschreibungen	16
5.1	Spieler erstellen	16
5.2	Spieler laden	17
5.3	Spielmodus wählen	18
5.4	Thema wählen	19

6	Projektplan	20
7	Klassendiagramm (vorläufig)	21
7.1	21
7.2	21
8	Sequenzdiagramme	22
8.1	SD Spieler erstellen	23
8.2	SD Spieler laden	25
8.3	SD Spiel starten - LFX	26
9	Klassendiagramm (final)	27
10	Weitere Diagramme	29
10.1	Zustandsdiagramm - Memorykarte	29
10.2	Aktivitätsdiagramm - Spielablauf	30
11	Implementierung	31
11.1	Allgemeines	31
11.2	Umfang	31
11.3	31
12	Test	32

1 Anforderungen des Kunden

Die Diakonie Hochfranken möchte das Angebot an Lernsoftware in ihren Kindergärten und Kindertagesstätten erweitern. Es soll eine Lernsoftware entwickelt werden, die Gedächtnistraining und die Erweiterung des Wortschatzes spielerisch umsetzt.

1.1 Systemvoraussetzungen

Aktuell verfügt der Kunde über Einzelplatz-PC's, die in den Jahren 2005 bis 2010 angeschafft wurden. Als Betriebssystem kommt Linux und Windows zum Einsatz.

1.2 Zielgruppe

Die Besucher der Einrichtung im Alter von fünf bis zehn Jahren bilden die Zielgruppe der Software.

2 Lastenheft

2.1 Zielbestimmung

Mit der Lernsoftware wird die Möglichkeit geschaffen, das pädagogisch wertvolle Prinzip des klassischen Memory-Spiel auf eine digitale Plattform zu übertragen. Zusätzlich zum Gedächtnistraining trägt die Vokabelfunktion zur Erweiterung des Wortschatzes bei. Die Führung einer Highscore ermöglicht den Vergleich der erreichten Punkte und steigert die Motivation der Benutzer, eine bessere Platzierung zu erreichen. Das erhöht wiederum den Lerneffekt. Neben dem Spielen an sich wird der Umgang mit dem Computer spielerisch erlernt. Vor Spielbeginn sind Attribute festzulegen und Meldungen der Software zu beachten. Das schult zusätzlich Logik. Der 2-Player-Modus unterstützt außerdem die Kommunikation mit anderen Spielern.

2.2 Produkteinsatz

Die Lernsoftware kommt in den Kindertagesstätten der Diakonie Hochfranken zum Einsatz. Anwender der Software sind Besucher der Kindertagesstätte im Alter von fünf bis zehn Jahren.

2.3 Produktfunktionen

\LF10\	Spieler neu erstellen
\LF11\	Spieler laden
\LF12\	Spielerdaten ändern
\LF20\	Anzahl der Spieler wählen
\LF30\	Thema wählen
\LF40\	Spielfeldgröße wählen
\LF50\	Spiel starten
\LF60\	Highscore anzeigen
\LF61\	Urkunde drucken

\LF70\	Vokabeltraining
\LF80\	Audiodaten abspielen

2.4 Produktdaten

\LD10\	Spielerdaten
\LD20\	Highscoredaten

2.5 Produktleistungen

2.6 Qualitätsanforderungen

Funktionalität:	gut
Zuverlässigkeit:	sehr gut
Benutzbarkeit:	gut
Effizienz:	normal
Änderbarkeit:	normal
Portierbarkeit:	sehr gut
Spassfaktor:	sehr gut

2.7 Ergänzungen

Die Umsetzung der Software erfolgt in der Programmiersprache Java. Da der Kunde Linux und Windows als Betriebssystem einsetzt stellt dies die notwendige Portierbarkeit sicher.

3 Aufwandskalkulation

3.1 Function-Point-Analyse

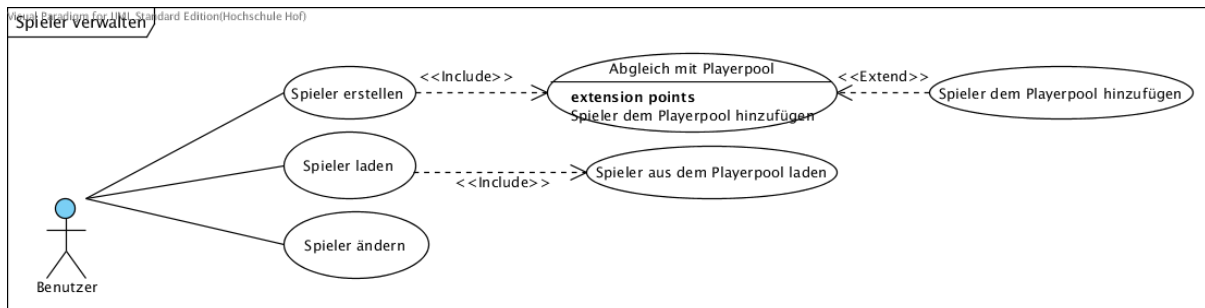
3.2

Function Point - Methode				
Kategorie	Anzahl	Klassifizierung	Gewichtung	Zeilensumme
Eingaben	3	Einfach	3	9
	4	Mittel	4	16
	0	Komplex	6	0
Abfragen	2	Einfach	3	6
	1	Mittel	4	4
	2	Komplex	6	12
Ausgaben	8	Einfach	4	32
	2	Mittel	5	10
	3	Komplex	7	21
Datenbestände	0	Einfach	7	0
	1	Mittel	10	10
	1	Komplex	15	15
Referenzdaten	1	Einfach	5	5
	0	Mittel	7	0
	0	Komplex	10	0
Summe			E1	140
Einflußfaktoren		1 Verflechtung mit anderen Anwendungssystemen (0-5)		0
	(ändern den Function Point-Wert um +/- 30%)	2 Dezentrale Daten, dezentrale Verarbeitung (0-5)		0
		3 Transaktionsrate (0-5)		0
		4 Verarbeitungslogik		
		A Rechenoperationen (0-10)		0
		B Kontrollverfahren (0-5)		1
		C Ausnahmeregelungen (0-10)		
		D Logik (0-5)		3
		5 Wiederverwendbarkeit (0-5)		1
		6 Datenbestandskonvertierungen (0-5)		0
		7 Anpaßbarkeit (0-5)		2
Summe der 7 Einflüsse			E2	7
Faktor Einflußbewertung = $(E2/100) + 0,7$			E3	0,77
Bewertete Function Points: $E1 * E3$				107,8

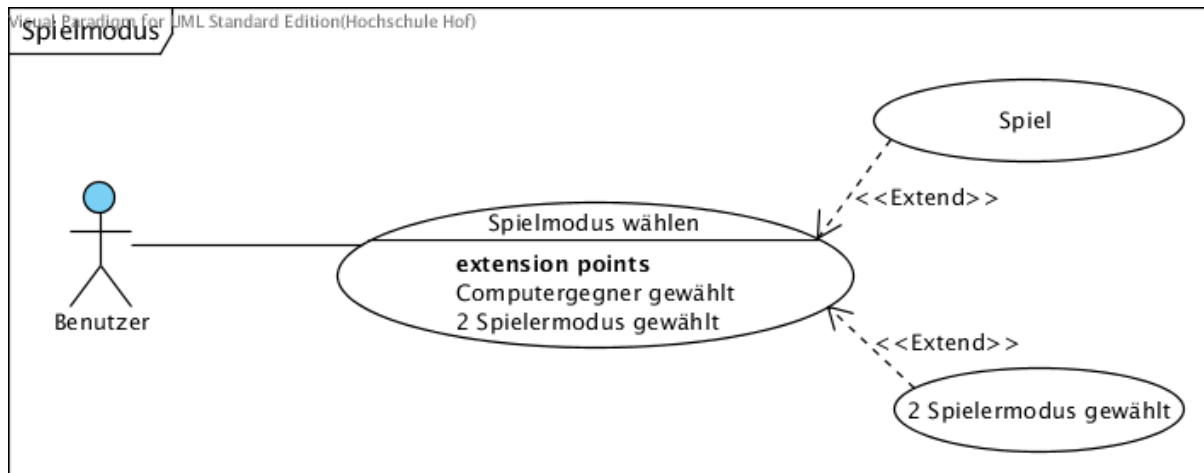
4 Use Case Diagramme

Folgend wird die Software als Use Case Diagramme dargestellt. Use Cases geben die Außensicht des Systems wieder. Es werden typische Funktionalitäten beschrieben, die der Benutzer mit dem System ausführt.

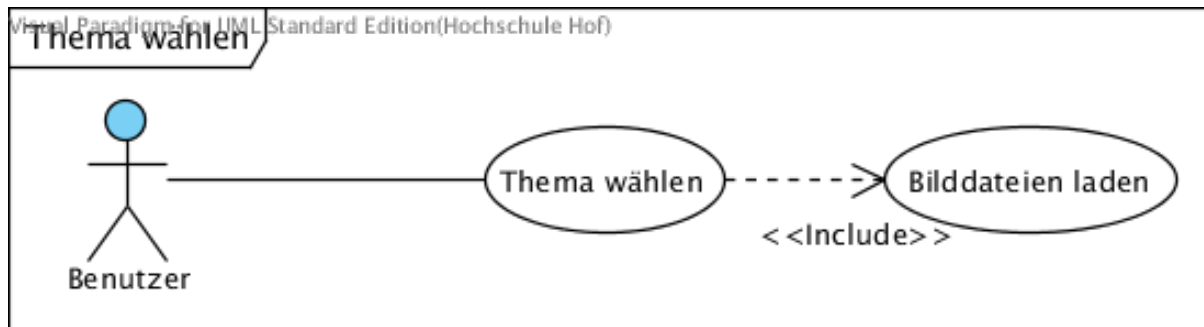
4.1 Spieler verwalten - \LF10\,\LF11\,\LF12\



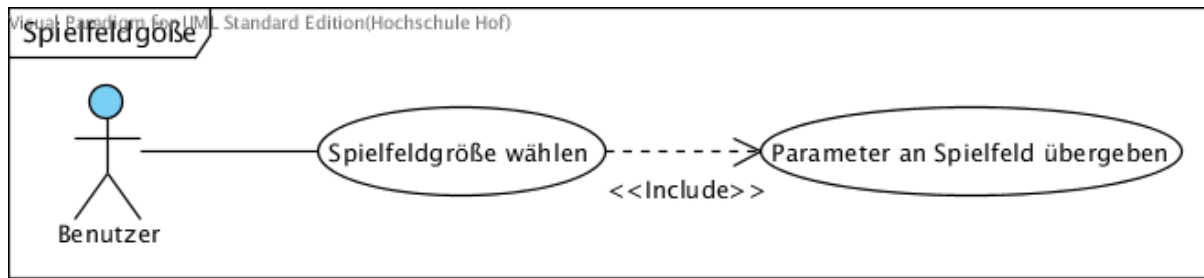
4.2 Anzahl der Spieler wählen - \LF20\



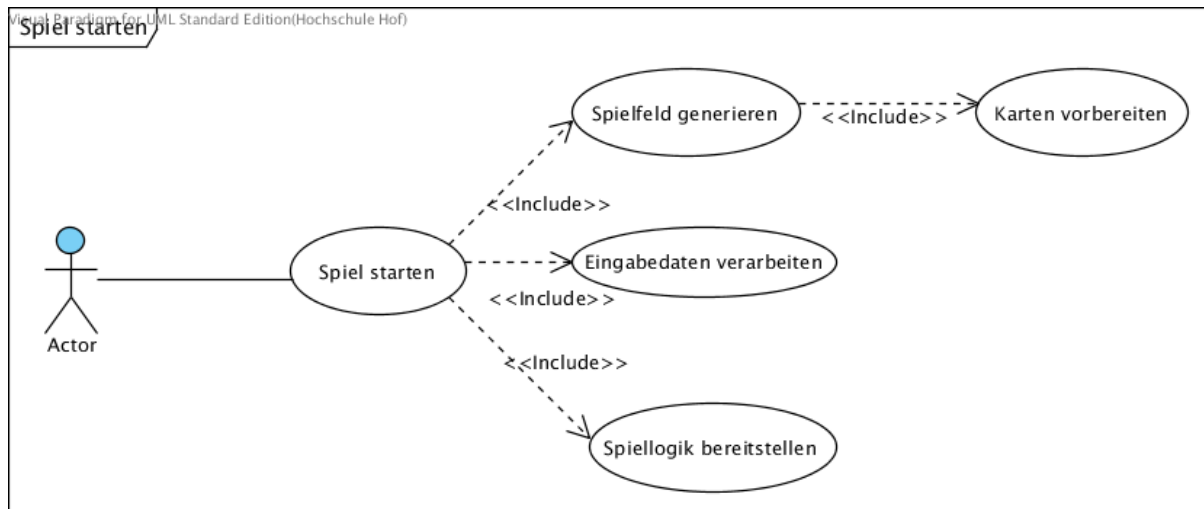
4.3 Thema wählen - \LF30\



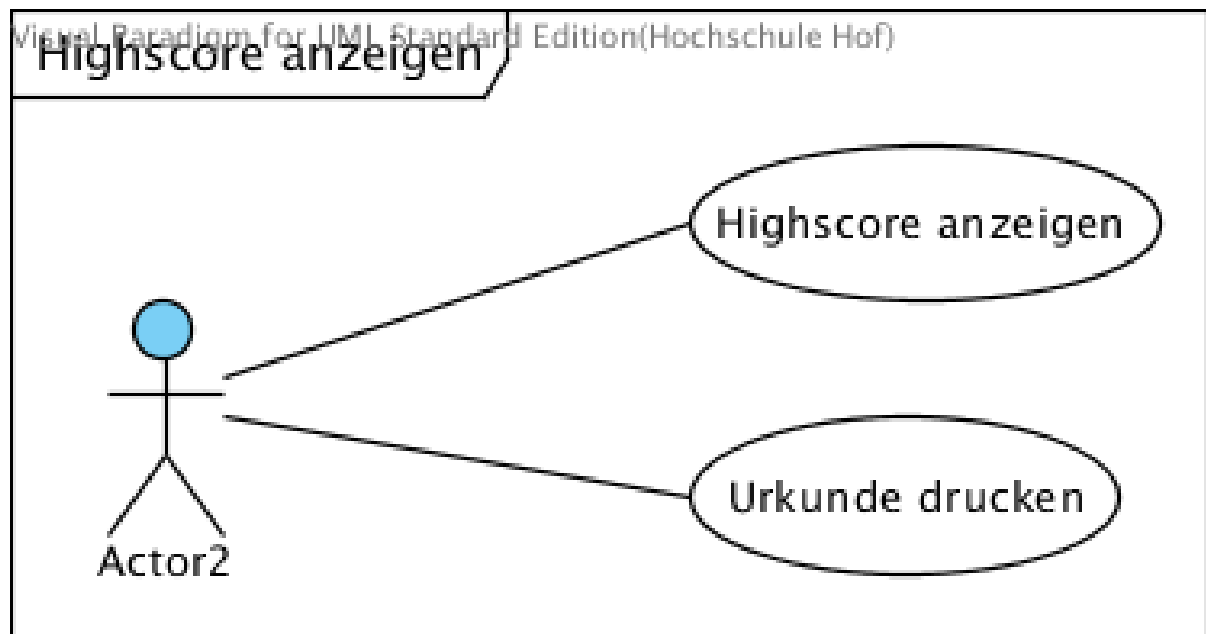
4.4 Spielfeldgröße wählen - \LF40\



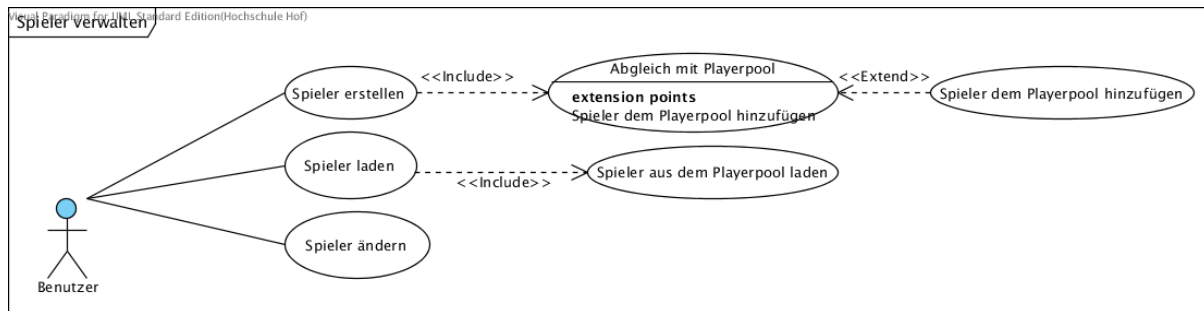
4.5 Spiel starten - \LF50\



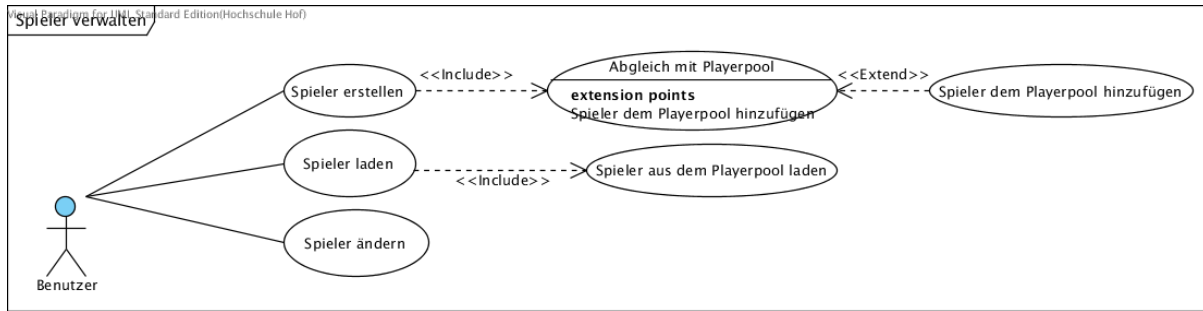
4.6 Highscore - \LF60\,\LF61\



4.7 Vokabeltraining - \LF70\



4.8 Audiodaten abspielen - \LF80\



5 Use Case Beschreibungen

5.1 Spieler erstellen

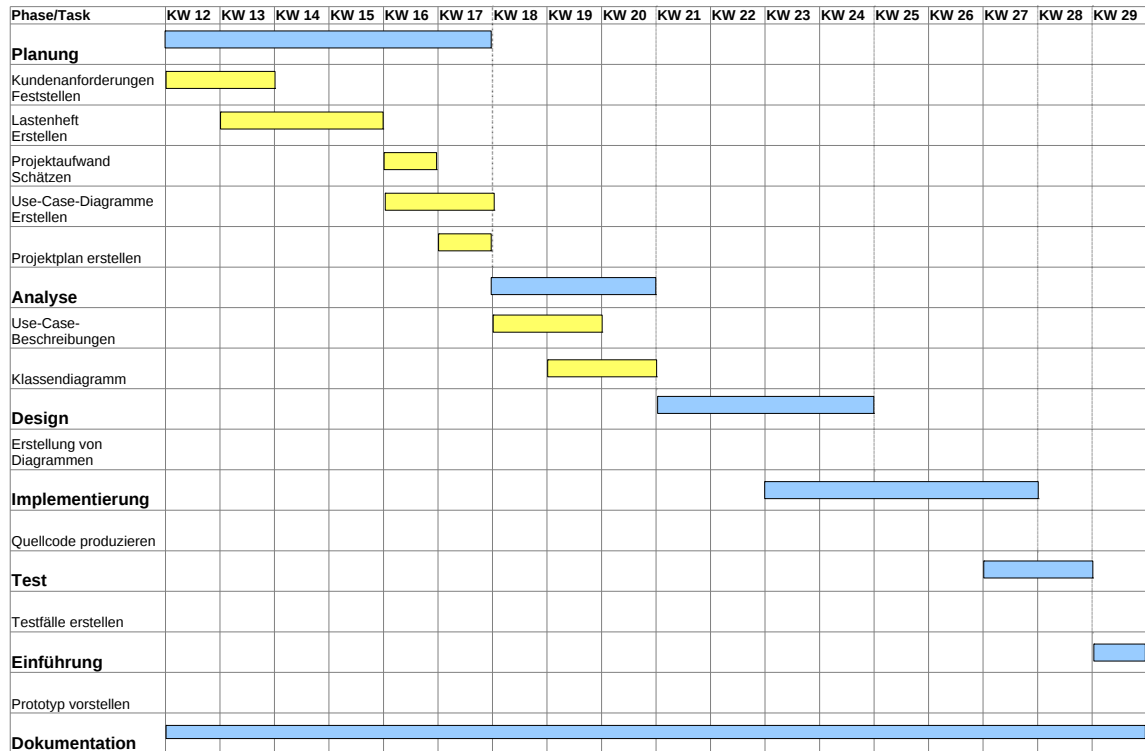
Use Case:	Spieler erstellen	
Actors:	Benutzer	
Purpose:	Benutzer erstellt durch Eingabe seines Namens einen Spieler	
Entry Cond:		
Overview:	Für den Benutzer ist noch kein Spieler erstellt. Der Benutzer trägt seinen Namen ein und e	
Exit Cond:		
Includes:		
Special Req:		
Category:		
Cross Ref:	auf /LF10/ aus Lastenheft	
Ablauf:	Actor Action:	System Response:
	1. Spieler erstellen wählen	
	2. Spielernamen eintragen	

5.2 Spieler laden

5.3 Spielmodus wählen

5.4 Thema wählen

6 Projektplan



7 Klassendiagramm (vorläufig)

7.1

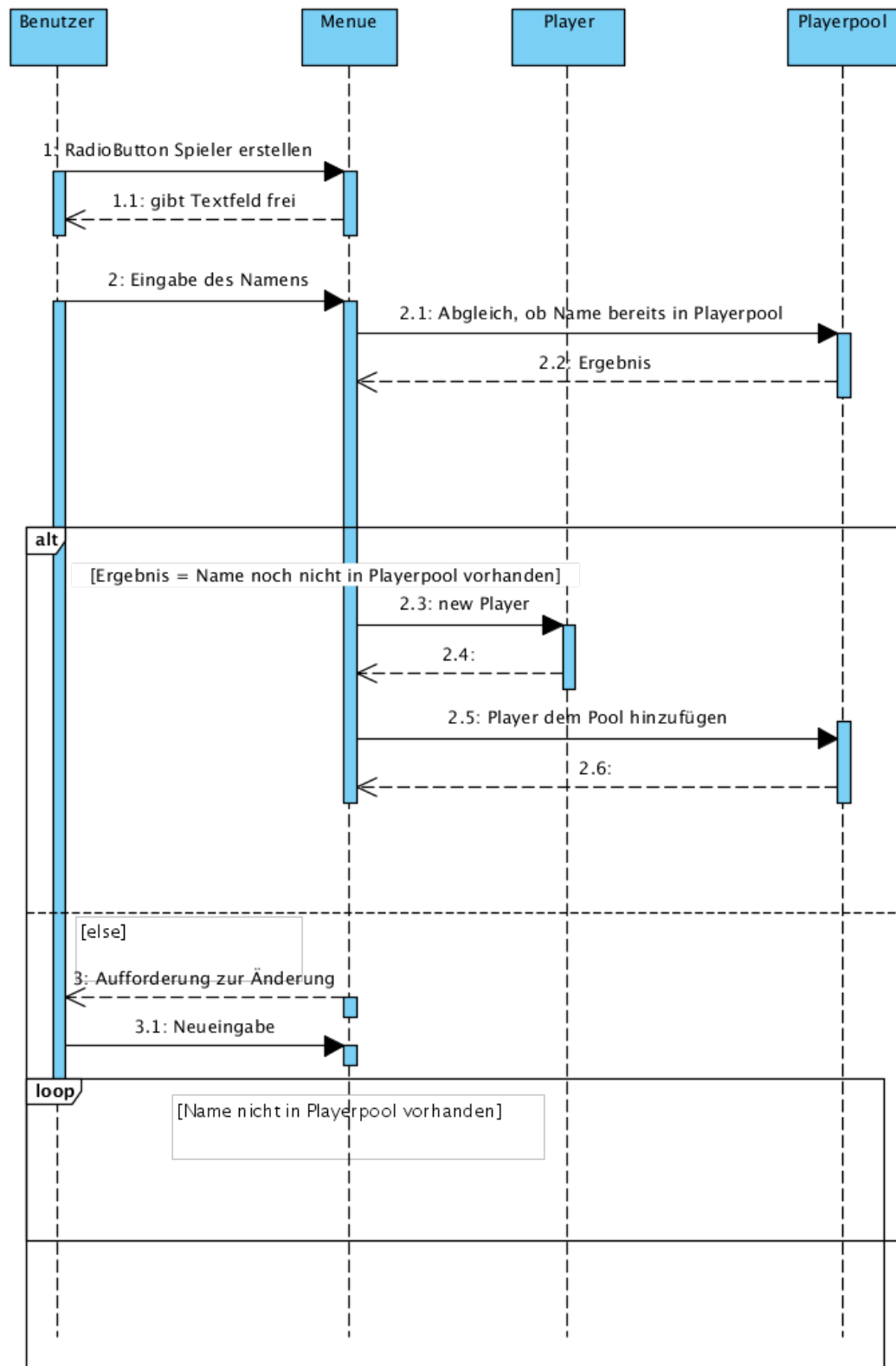
7.2

8 Sequenzdiagramme

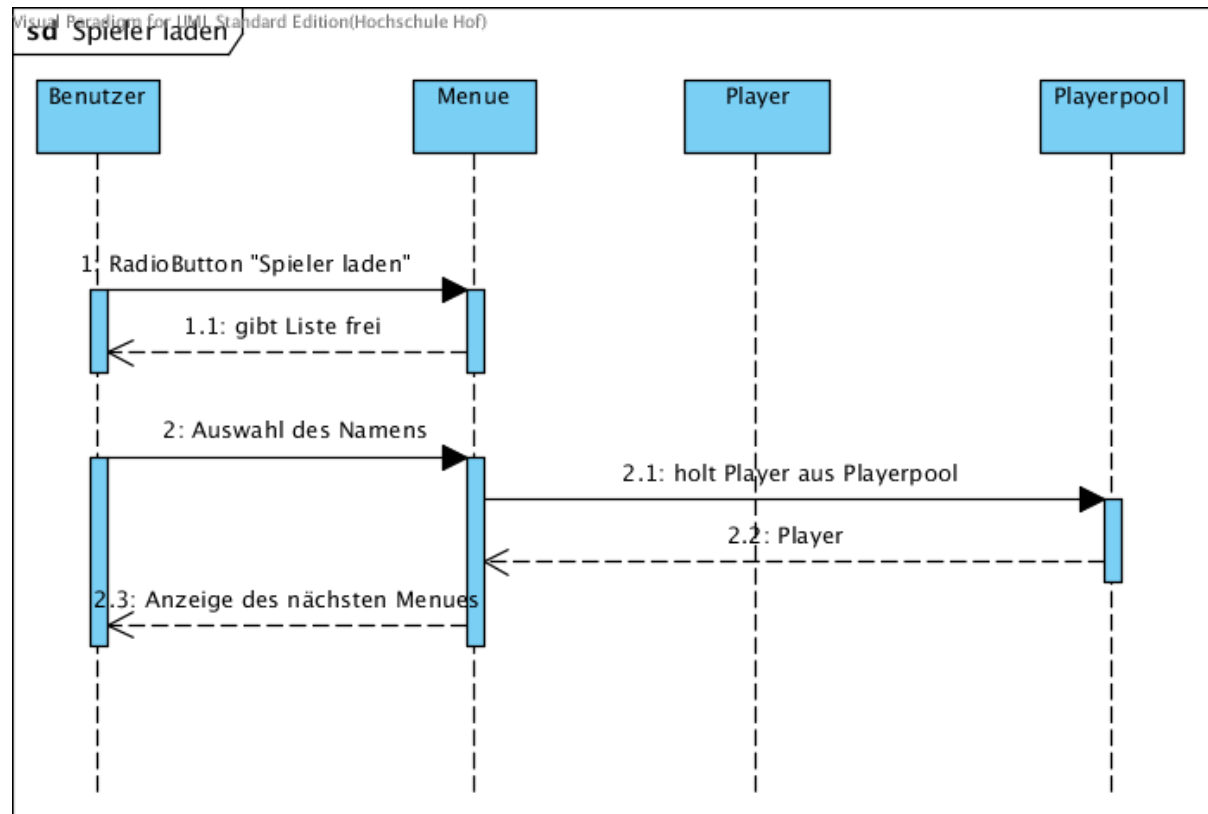
Sequenzdiagramme bla bla bla

8.1 SD Spieler erstellen

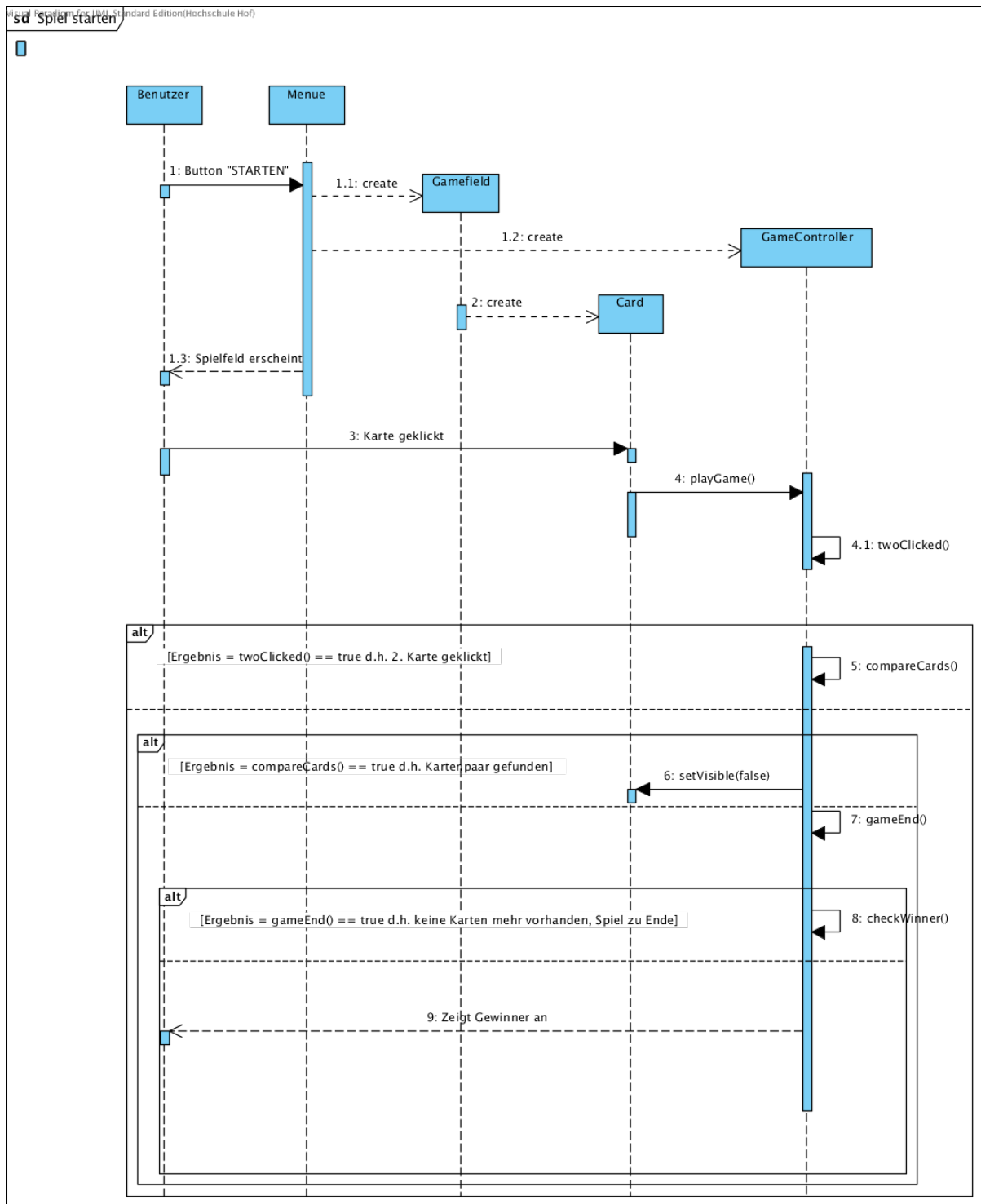
sd Spieler erstellen



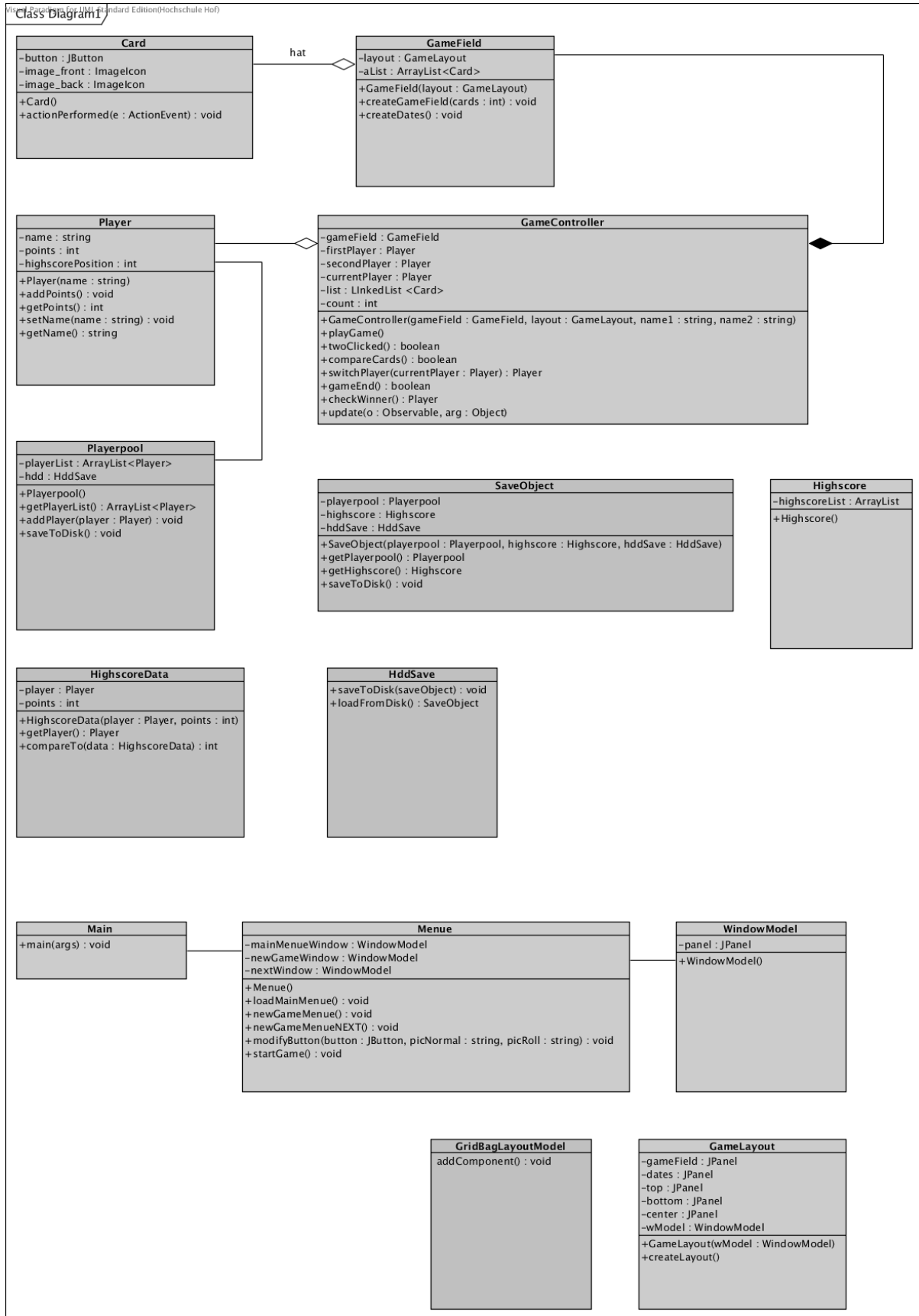
8.2 SD Spieler laden



8.3 SD Spiel starten - LFX



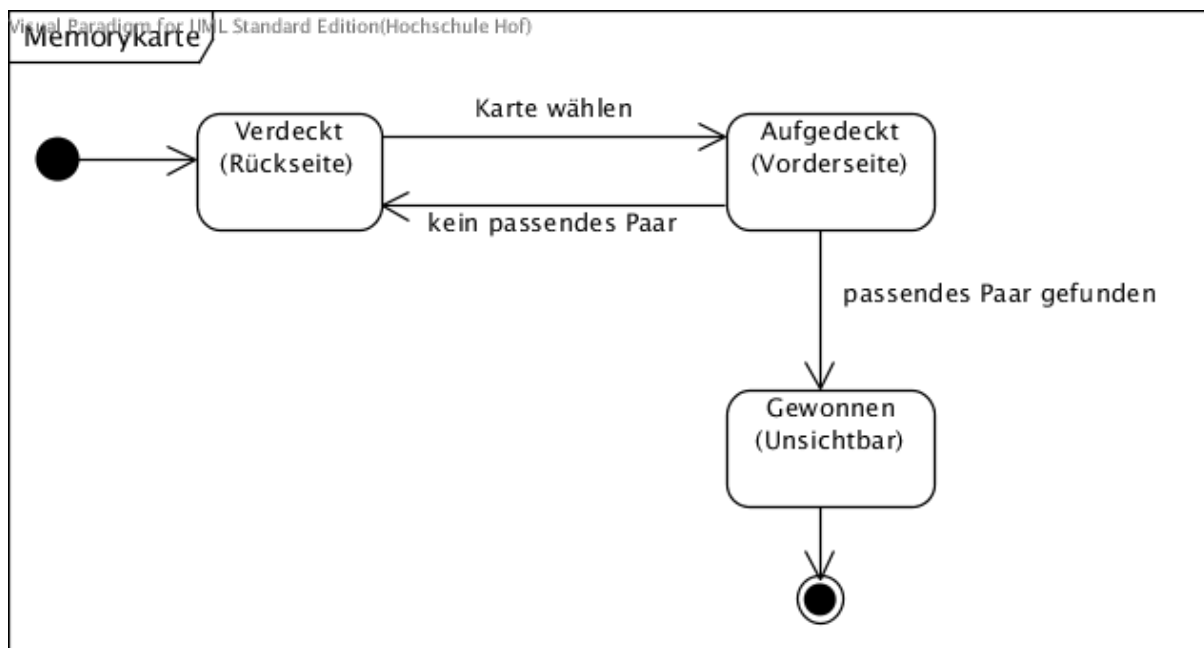
9 Klassendiagramm (final)



10 Weitere Diagramme

Sequenzdiagramme bla bla bla

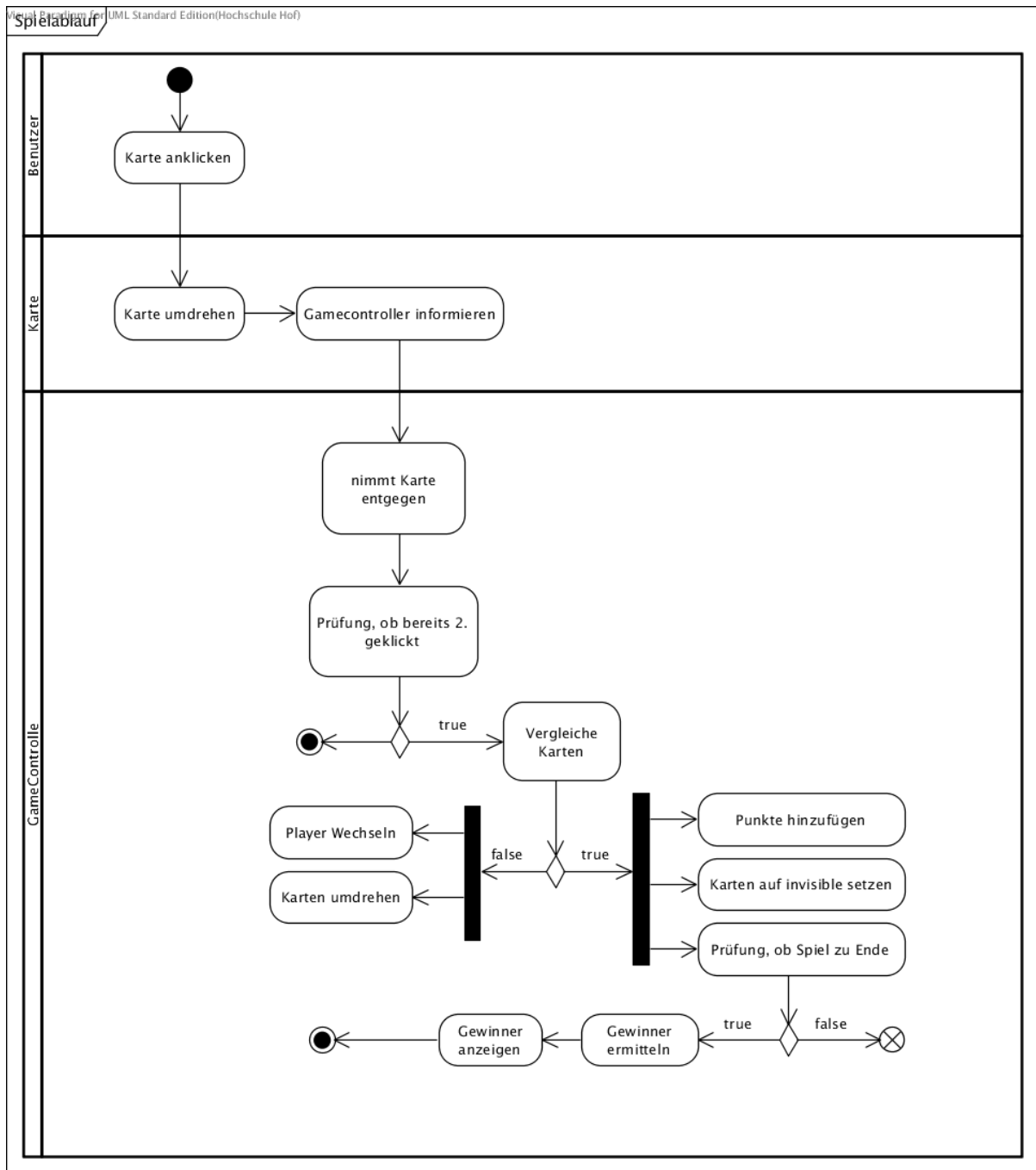
10.1 Zustandsdiagramm - Memorykarte



10.1.1 Beschreibung

Die Karte kann die drei Zustände Verdeckt, Aufgedeckt und Gewonnen annehmen. Wird eine verdeckte Karte ausgewählt, wechselt sie in den Zustand Aufgedeckt. Der Zustand Aufgedeckt kann entweder durch die Feststellung, dass ein passendes Kartenpaar gefunden wurde in den Zustand Gewonnen übergehen, oder falls es sich um nicht zwei übereinstimmende Karten handeln, wieder in den Zustand Verdeckt wechseln.

10.2 Aktivitätsdiagramm - Spielablauf



11 Implementierung

11.1 Allgemeines

Die Implementierung wurde für alle Lastenheftfunktionen der Priorität 1 umgesetzt. Alle Elemente der grafischen Benutzeroberfläche sind ohne GUI-Builder erstellt worden. Der Programmcode liegt als .java Dateien vor und kann über den Aufruf von Main.java ausgeführt werden.

11.2 Umfang

Es wurden 16 Klassen mit insgesamt X Lines of Code implementiert.

```
http://cloc.sourceforge.net v 1.53  T=0.5 s (32.0 files/s, 3324.0 lines/s)
```

Language	files	blank	comment	code
Java	16	342	39	1281
SUM:	16	342	39	1281

11.3

12 Test

