

# **Entwicklung eines “Kniffel“ Computer-Spiels**

## **Teil II**

### **Design**

**Dipl.-Inform(FH) Adelheid Knodel**

1	Allgemeines .....	3
2	Schichtenmodell.....	3
2.1	Benutzerschnittstelle.....	3
2.2	Anwendungsschicht.....	3
2.3	Persistenzschicht.....	3
3	Use Cases in der ersten Realisierungsphase .....	3
4	Bedienkonzept .....	3
4.1	Spiel beginnen und Spiel beenden .....	4
4.2	Spielzug durchführen.....	6
5	System Use Cases .....	7
5.1	System-Use-Case Diagramm .....	7
5.2	System-Use-Case Beschreibungen.....	7
5.2.1	System-Use-Case “Kniffel starten” (SUC-1) .....	7
5.2.2	System-Use- Case “Kniffelspielzug” (SUC-2) .....	8
5.3	Detaillierte Beschreibung der System Use Cases durch Aktivitätsdiagramme .....	9
5.3.1	Aktivitätsdiagramm zu System Use Case “Kniffel starten” (SUC-1).....	9
5.3.2	Aktivitätsdiagramm zu System Use Case “Kniffelspielzug” (SUC-2).....	10
5.4	Systemoperationen.....	11
5.4.1	Systemoperationen im Rahmen des System Use Case “Kniffel starten”	11
5.4.2	Systemoperationen im Rahmen des System Use Case „Kniffelspielzug“	13
6	Zustandsmodell .....	14
7	Applikationsschicht .....	14
7.1	Die detaillierten Systemoperationen des Kniffelstarts.....	15
7.1.1	SYSOP-ST1 newGame():State.....	15
7.1.2	SYSOP-ST2 insertName(name: String):State .....	16
7.2	Die detaillierten Systemoperationen des Kniffelspielzugs.....	17
7.2.1	SYSOP-SP1 throwDices():State .....	17
7.2.2	SYSOP-SP2 chooseDice(numberDice:int):State .....	18
7.2.3	SYSOP-SP3 writePoints(fieldnumber:IFieldnumber):void .....	19
7.3	Die detaillierte Systeminitialisierung .....	20
8	Das Objektmodell.....	21
9	Anbindung der Oberfläche .....	22

# **1 Allgemeines**

Das vorliegende Dokument beschreibt das Design des Kniffel – Computerspiels. Grundlage für das Design ist das in der ersten Phase erarbeitete Analysedokument Kniffel-Analyse.pdf.

## **2 Schichtenmodell**

Die primäre Architektur des Kniffel-Spiels beruht auf einem Drei-Schichtenmodell. Die einzelnen Schichten nutzen lediglich die Funktionalität der eigenen Schicht oder die der darunter liegenden.

### **2.1 Benutzerschnittstelle**

Diese Schicht bietet den Benutzern die Möglichkeit mit einem Programm Daten auszutauschen und stellt dafür die Schnittstelle bereit. Hier werden Eingaben in Dienstanforderungen an die Applikationsschicht gewandelt und die aktuelle Spielsituation angezeigt. Die Bedienoberfläche verfügt selbst über keine weitere Funktionalität der Anwendungslogik. Für die aktuelle Aufgabe wird ein grafisches Bedienkonzept beschrieben und realisiert. Dies ist die Grundlage für die weitere Betrachtung der Use-Cases. Die daraus resultierende Zustandsmaschine wird nicht in der Benutzerschnittstelle, sondern ebenfalls in der Logik der Applikationsschicht realisiert. Die Benutzerschnittstelle wird lediglich über die Zustandswechsel informiert.

### **2.2 Anwendungsschicht**

Die Anwendungsschicht ist die Schicht, die die eigentliche Funktionalität erbringt. Sie verfügt über eine funktionale Schnittstelle, die es ermöglicht, das Spiel zu spielen ohne auf eine spezielle Benutzerschnittstelle angewiesen zu sein. Aufgabe dieser Schnittstelle ist es darüber hinaus, die darunter liegenden Subsysteme zu kapseln und deren Details zu verbergen. Dies entkoppelt die einzelnen Schichten und erleichtert lokale Änderungen.

### **2.3 Persistenzschicht**

In dieser Schicht sind die grundlegenden Dienste angesiedelt, die den Zugriff auf Daten der Highscoreliste realisieren und deren persistente Speicherung ermöglichen

## **3 Use Cases in der ersten Realisierungsphase**

In der ersten Phase der Realisierung werden die Use Cases „Spiel starten“ und „Spielzug durchführen“ realisiert, da dies die Wichtigen sind um das Spiel spielen zu können.

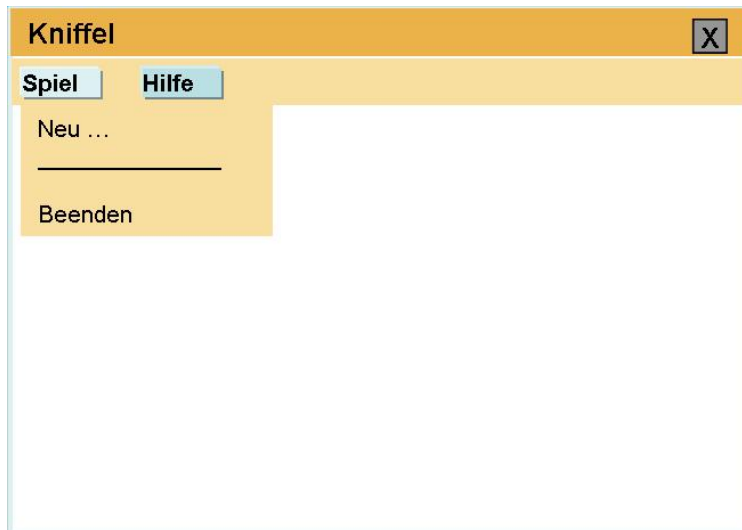
## **4 Bedienkonzept**

Die Aufteilung des Bedienkonzepts erfolgt gemäß der Einteilung der Use-Cases in die Dialoge für „Spiel starten“, „Gewinner in Highscoreliste eintragen“, „Spielzug durchführen“.

## 4.1 Spiel beginnen und Spiel beenden

Nach dem Start des Spiels steht dem User ein Dialog zur Verfügung, der ihm die Möglichkeit bietet

- Neu ...:  
ein neues Spiel zu beginnen, siehe Use-Case "Spiel starten"
- Beenden:  
das Spiel zu beenden, hier wird nochmals nachgefragt, ob das Spiel tatsächlich beendet werden soll.



## Spielernamen eingeben

Nach der Wahl des Menüpunktes "Neu" wird den Spielern über einen Dialog die Möglichkeit gegeben ihre Namen in den Spielplan einzutragen.



The image shows a software window titled "Kniffel" with a standard Windows-style title bar (yellow background, close button 'X' on the right). Below the title bar is a menu bar with two items: "Spiel" and "Hilfe", both highlighted in light blue. The main content area of the window is white. In the center, there is a yellow rectangular box containing the text "Spielernamen eingeben (max. 6 Namen):". Below this text is a label "Name:" followed by a white text input field. At the bottom of the yellow box are two buttons: "speichern" and "fertig", both with grey backgrounds and black text.

## Spielende

Nachdem alle Runden gespielt sind, wird der Sieger bestimmt und angezeigt.

## 4.2 Spielzug durchführen

Für die Durchführung von "Spielzug durchführen" (siehe Use Case "Spielzug durchführen") wird der folgende Dialog verwendet.

	Name1	Name2	Name3	Name4	Name5	Name6
• • • zählen 1er						
• • • • zählen 2er						
• • • • • zählen 3er						
• • • • • • zählen 4er						
• • • • • • • zählen 5er						
• • • • • • • • zählen 6er						
Oben gesamt:						
Bonus (+35 Pkt., ab 63Pkt.)						
Oben gesamt (inkl. Bonus):						
Alle Augen Dreierpasch						
Alle Augen Viererpasch						
25 Punkte Full House						
30 Punkte Kleine Straße						
40 Punkte Große Straße						
50 Punkte Kniffel						
Alle Augen Chance						
100 Punkte ExtraKniffel *						
unten gesamt:						
Gesamte Punktzahl:						

Im Dialog "Spielzug durchführen" wird angezeigt: die Runde, in der gerade gespielt wird, der Name des Spielers, der an der Reihe ist, und dessen Wurfanzahl.

Der Spieler kann durch Klicken auf den Button "Würfeln", das Würfeln starten. Die Ergebnisse werden angezeigt. Die passenden Felder werden mit orangenem Rand, die nichtpassenden Felder mit blauem Rand dargestellt.

Der Spieler kann durch Klicken auf die Würfel auswählen, mit welchen weiter gewürfelt werden soll. Die Farbe der Würfel wechselt von grün zu rot bzw. von rot auf grün. Dies kann in beliebiger Reihenfolge wiederholt werden.

Durch Klicken auf ein Punktfeld wird der ermittelte Wert eingetragen, falls die Augenzahlen den geforderten Würfelbildern entsprechen, andernfalls wird "0" als Ergebnis eingetragen.

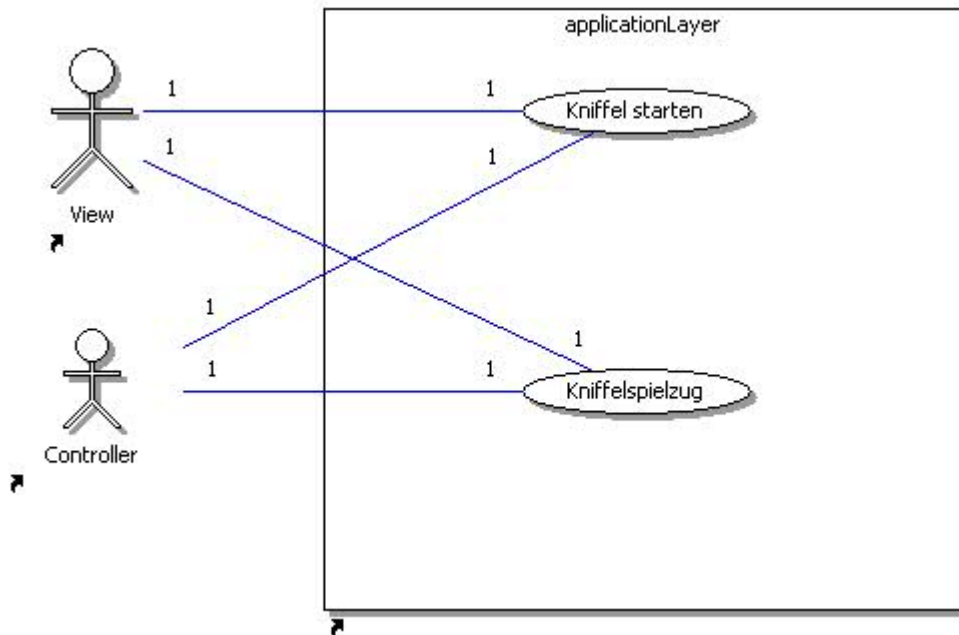
Nach dem 3. Wurf wird der Würfel deaktiviert und der Spieler muss sich durch Klicken für ein Punktfeld entscheiden.

Sind alle 13 Runden gespielt, werden die Gesamtpunktzahlen und der Sieger angezeigt.

## 5 System Use Cases

Im Folgenden werden die zentralen Spielsituationen in Form von System Use Cases beschrieben. Jeder Use-Case umfasst neben einer kurzen Beschreibung des Anwendungsfalls und den beteiligten Akteuren eine Referenz auf die den Anwendungsfall betreffenden Anforderungen und den entsprechenden Anwendungsfall des Analysemodells.

### 5.1 System-Use-Case Diagramm



### 5.2 System-Use-Case Beschreibungen

#### 5.2.1 System-Use-Case “Kniffel starten“ (SUC-1)

Akteure: View, Controller

Priorität: Hoch

Referenz: UC-1(Kniffel starten), F-1, F-2, F-23, E1, E2, E3, E4

Beschreibung: Nach der Wahl des Menüpunktes „Neu ...“, muss der Spielplan (E1) mit den einzelnen Punkte- (E2) und Summenfeldern (E3) initialisiert werden. Die User bekommen einen Dialog präsentiert, der es ihnen ermöglicht Namen einzugeben, diese werden in die Namensfelder (E4) eingetragen. Nach jeder Namenseingabe, kann der User den Eingabedialog beenden, durch klicken auf „fertig“ oder einen weiteren Namen eingeben und speichern durch click auf „speichern“, bis max. 6 Namen eingegeben wurden. Wurden 6 Namen eingegeben ist keine weitere Eingabe möglich.

Vorbedingung: Das Spiel ist initialisiert und es wurde ein neues Spiel gestartet.

## 5.2.2 System-Use- Case “Kniffelspielzug“ (SUC-2)

Akteure: View, Controller

Priorität: Hoch

Referenz: UC-2 (Kniffel spielen),  
F-3, F-4, F-5, F-6, F-7, F-8, F-9, F-19, F-20, F-21, F-22, F-24, F-25  
E5, E6, E7, E2

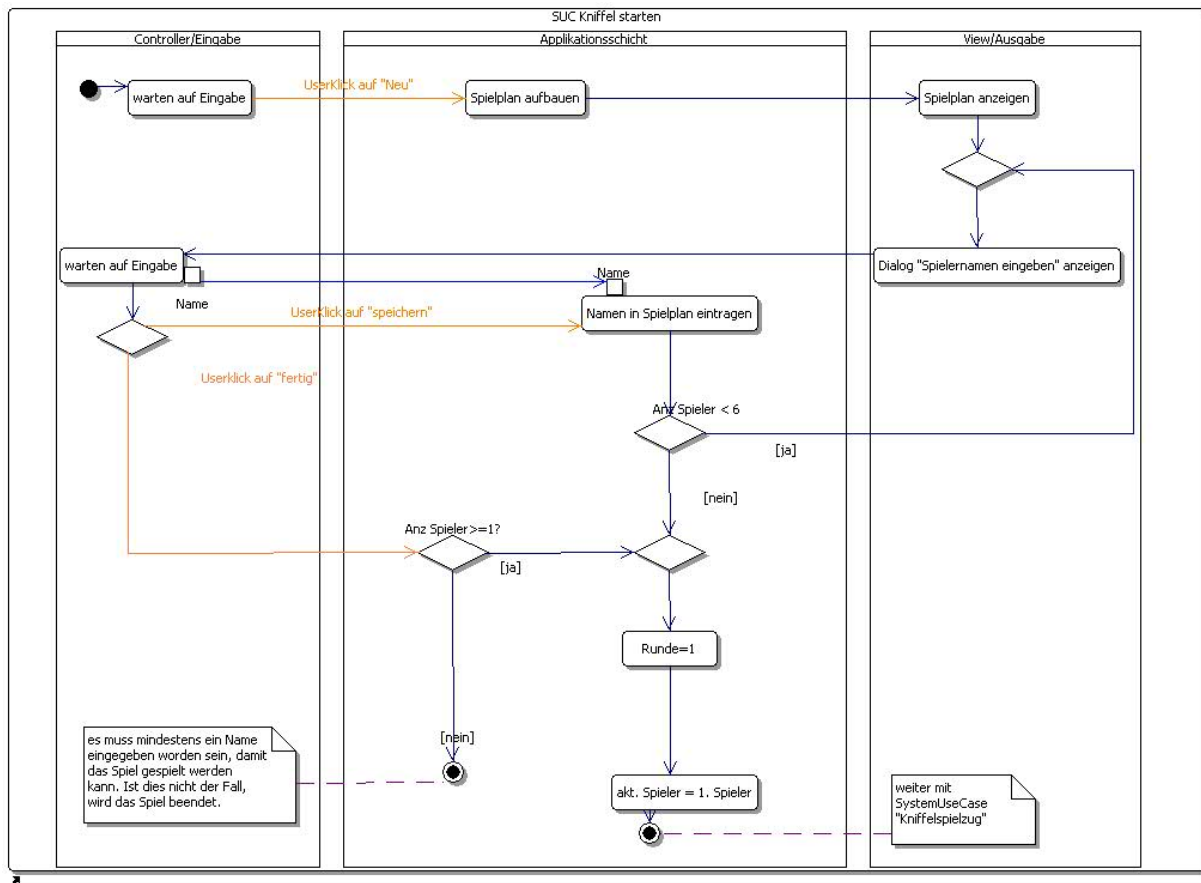
Beschreibung: Den Usern wird im Feld „Runde:“ angezeigt in welcher Spielrunde sie sich gerade befinden, im Feld „Spieler:“ wird der Name des Users, der an der Reihe ist angezeigt, und im Feld Würfle, wird die Anzahl der bereits durchgeführten Würfe angezeigt. Der User klickt auf den Button “Würfeln” und löst damit das Würfeln für alle 5 Würfel aus. Die erzielte Augenzahl wird durch die Würfel angezeigt, alle Würfel werden in der Farbe grün dargestellt. Nun kann der User durch klicken auf die Würfel auswählen, mit welchen nicht mehr gewürfelt werden soll, diejenigen werden rot dargestellt. Anschließend kann er wieder auf den Button “Würfeln” drücken und den nächsten Wurf mit den nicht zurückgehaltenen Würfeln starten. Falls er mit den gewürfelten Punkten zufrieden ist, kann er auf eines der Punktefelder klicken, dadurch werden die erzielten Punkte in das entsprechende Punktefeld eingetragen, falls die Augenzahlen den geforderten Würfelbildern entsprechen, andernfalls wird “0” als Ergebnis eingetragen. Nach dem Eintragen der Punkte werden die entsprechenden Summenfelder aktualisiert. Nach dem zweiten Wurf kann der Spieler wieder aus allen 5 Würfeln diejenigen auswählen, die er halten bzw. diejenigen mit denen er noch mal würfeln möchte. Nach dem 3. Wurf muss der Spieler sich für ein Punktefeld durch Drücken eines Punktebuttons entscheiden, der Button „Würfeln“ wird gesperrt. Nach dem Eintrag der Punkte wird die Wurfanzahl auf 0 gesetzt und der nächste Spieler kommt an die Reihe. Dies wird in der Runde solange wiederholt bis alle Punktefelder ausgefüllt sind. Sind alle 13 Runden gespielt, werden die Gesamtpunktzahlen und der Sieger ermittelt. Der Gewinner wird gefragt, ob er sich in die Highscoreliste eintragen möchte.

offene Punkte:

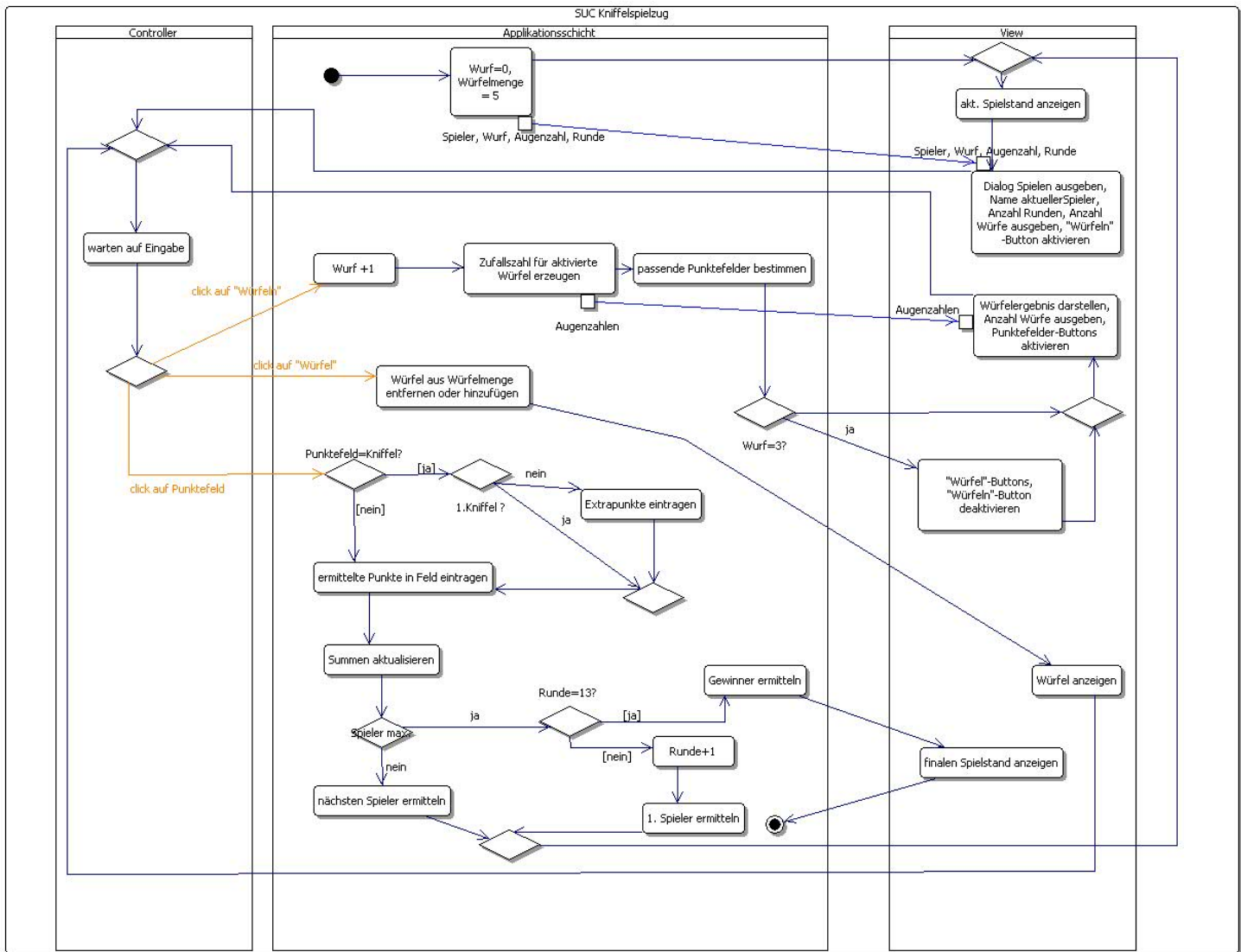


## 5.3 Detaillierte Beschreibung der System Use Cases durch Aktivitätsdiagramme

### 5.3.1 Aktivitätsdiagramm zu System Use Case “Kniffel starten” (SUC-1)



### 5.3.2 Aktivitätsdiagramm zu System Use Case “Kniffelspielzug“ (SUC-2)



## 5.4 Systemoperationen

Systemsequenzdiagramme stellen Systemoperationen eines System Use Cases dar.

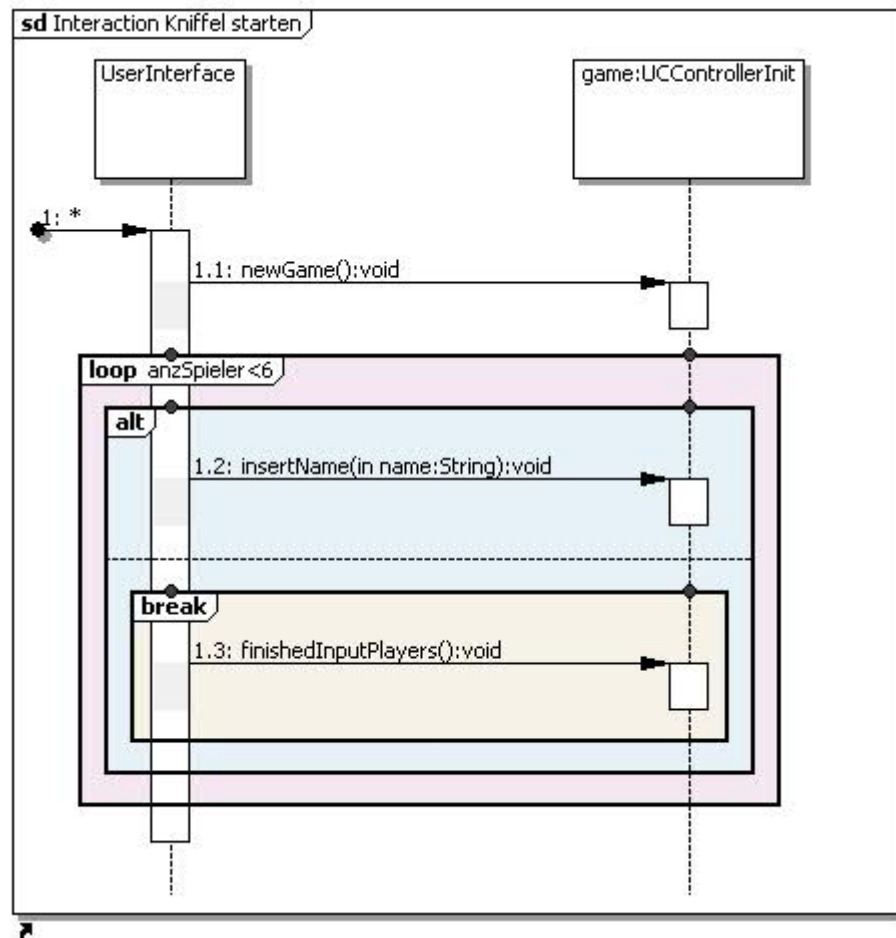
Eine Systemoperation wird aufgerufen als Reaktion auf ein Systemereignis, sie verändert den Zustand des Systems.

Ein Systemereignis ist eine Nachricht, die ein Akteur an das System sendet.

### Beschreibung der Systemoperationen

Name:	<u>Name der Operation</u>
Verantwortlichkeit:	Aufgabe/Ziel der Operation
Referenzen:	Operation gehört zu welchem UseCase
Bemerkungen:	
Ausnahmen:	Ausnahmefälle, die zum Abbruch der Operation führen
Output:	Ausgaben an andere Akteure
Vorbedingungen:	Anforderungen an den Anfangszustand, Bedingungen, die nicht geprüft werden, die aber gelten müssen, damit die Systemoperation erfolgreich ist, z.B. das Vorliegen der Ergebnisse von vorherigen Systemoperationen
Nachbedingungen:	beschreibt Änderungen zu Anfangszustand, es wird nicht die Aktion beschrieben, es wird nur ein "Schnappschuss" des Systems nachher mit einem vorher verglichen. Es wird nicht beschrieben wie der neue Zustand erreicht wird.

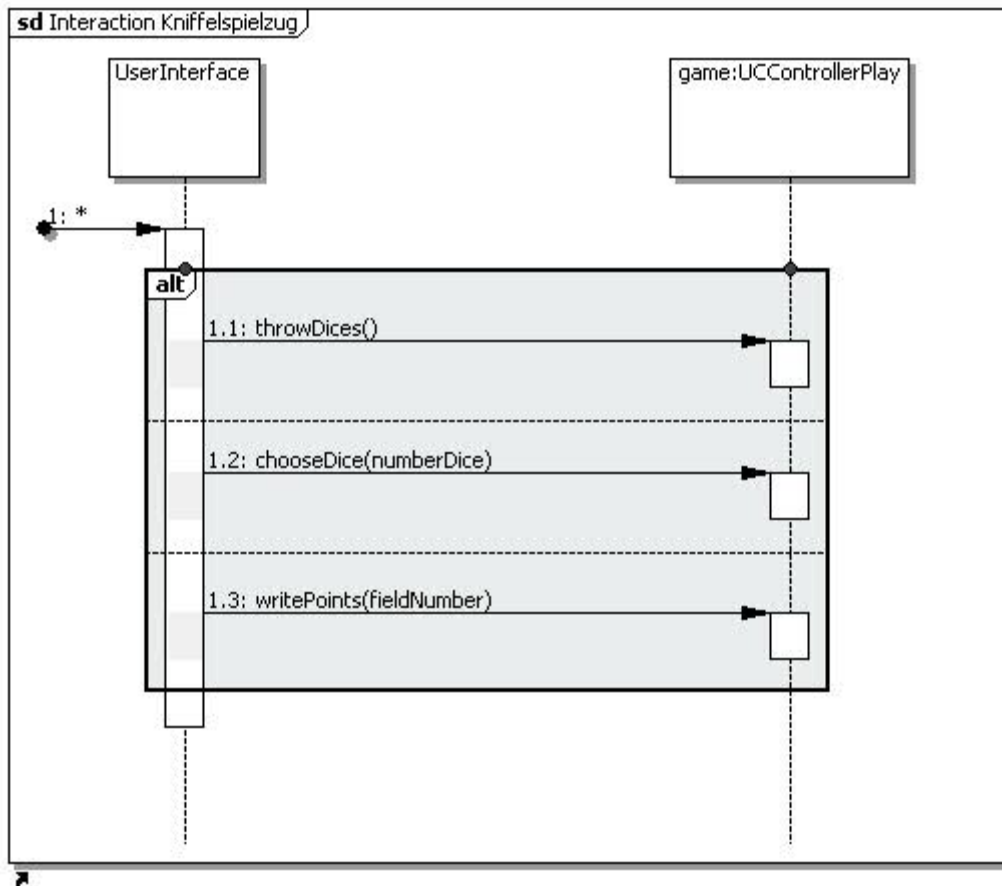
#### 5.4.1 Systemoperationen im Rahmen des System Use Case "Kniffel starten"



**Name:** **SYSOP-ST1 newGame():void**  
**Verantwortlichkeit:** Die Operation initialisiert ein neues Kniffelspiel, erzeugt den Spielplan. Wird die Operation während des Spielverlaufs aufgerufen, muss die bisherige Spielsituation bereinigt werden.  
**Referenzen:** System Use Case SUC-1 "Kniffel starten"  
**Bemerkungen:** Ein Neustart des Spiels ist jederzeit möglich.  
**Ausnahmen:**  
**Output:**  
**Vorbedingungen:**  
**Nachbedingungen:** Das Spiel ist initialisiert, die Container für die Spielerinformationen stehen bereit. Das Spiel ist im Zustand „Start“.

**Name:** **SYSOP-ST2 insertName(name: String ):void**  
**Verantwortlichkeit:** Die Operation legt ein neues Spielerobjekt an, stellt die Spalte in der Spieltabelle bereit und trägt den Namen in die Spalte ein.  
**Referenzen:** System Use Case SUC-1 "Kniffel starten"  
**Bemerkungen:**  
**Ausnahmen:** es wurden keine Zeichen im Namensfeld eingegeben  
**Output:**  
**Vorbedingungen:** Das Spiel ist initialisiert und es sind max. 5 Spieler eingetragen.  
**Nachbedingungen:** Eine neue Spalte in der Spieltabelle wurde angelegt, und evtl. die Eingabe der Spielernamen beendet.

## 5.4.2 Systemoperationen im Rahmen des System Use Case „Kniffelspielzug“

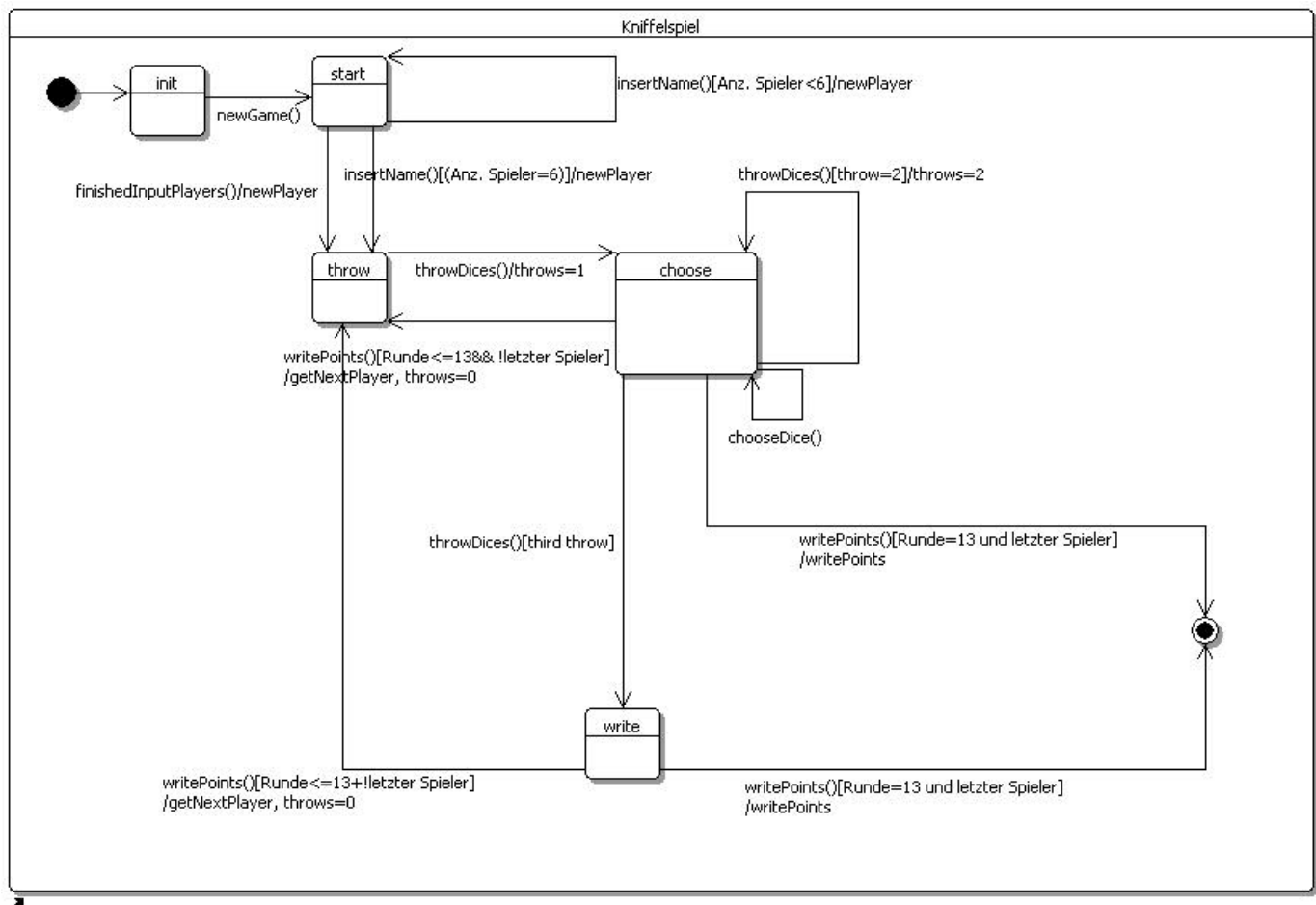


**Name:** **SYSOP-SP1 throwDices():void**  
**Verantwortlichkeit:** erzeugt Augenzahlen durch einen Zufallsgenerator für die Würfel mit denen gewürfelt werden soll  
**Referenzen:** System Use Case SUC-1 "Kniffelspielzug"  
**Bemerkungen:**  
**Ausnahmen:**  
**Output:** gewürfelte Augenzahlen  
**Vorbedingungen:** die Namenseingabe ist abgeschlossen, es ist mindestens ein Name eingetragen, es wurden noch keine 3 Würfe für den aktuellen Spieler getätigt  
**Nachbedingungen:** die gewählten Würfel haben eine neue Augenzahl, die Anzahl der Würfe wurde um 1 erhöht.

**Name:** **SYSOP-SP2 chooseDice(numberDice:IDicenumber):void**  
**Verantwortlichkeit:** ist der Würfel aktiviert, wird er deaktiviert  
 ist der Würfel deaktiviert, wird er wieder aktiviert,  
 d.h. der Würfel wird entweder dem Set zum Würfeln hinzugefügt oder daraus entfernt  
**Referenzen:** System Use Case SUC-1 "Kniffelspielzug"  
**Bemerkungen:**  
**Ausnahmen:**  
**Output:**  
**Vorbedingungen:** es wurde mindestens 1x gewürfelt  
**Nachbedingungen:** der Status der Würfel haben sich entsprechend geändert

<b>Name:</b>	<b>SYSOP-SP3 writePoints(fieldnumber:IFieldnumber):void</b>
Verantwortlichkeit:	die Punkte von den passenden Würfeln werden in das gewählte Feld eingetragen, die Summen neu berechnet. Falls noch nicht alle Runden gespielt sind, ist der nächste Spieler ist an der Reihe, falls eine neue Runde beginnt, wird die Anzahl der Runden erhöht.
Referenzen:	System Use Case SUC-1 "Kniffelspielzug"
Bemerkungen:	
Ausnahmen:	
Output:	alle Punkte- und Summenfelder
Vorbedingungen:	es wurde mindestens 1x gewürfelt
Nachbedingungen:	die Punkte sind eingetragen, die Summen aktualisiert.

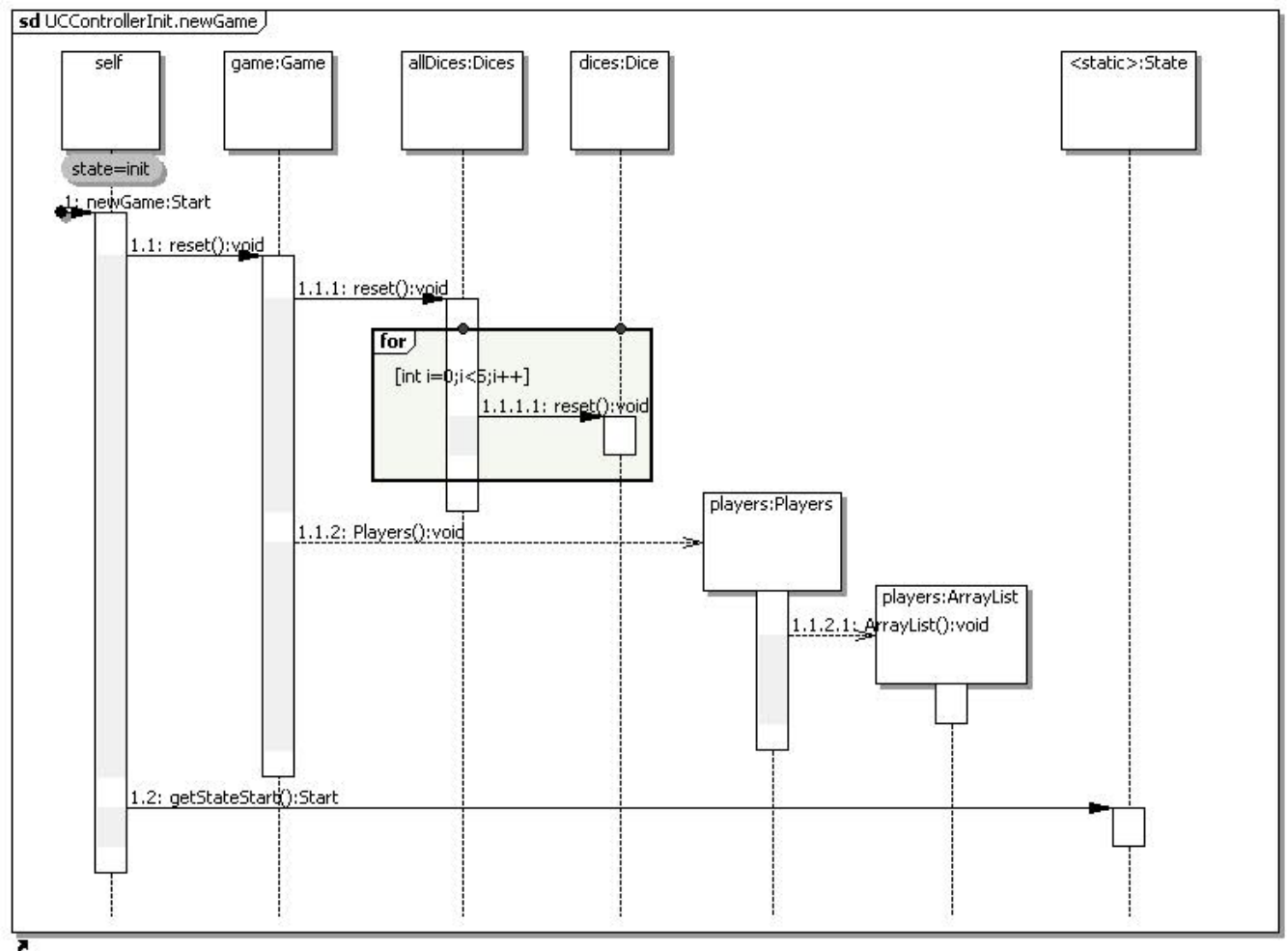
## 6 Zustandsmodell



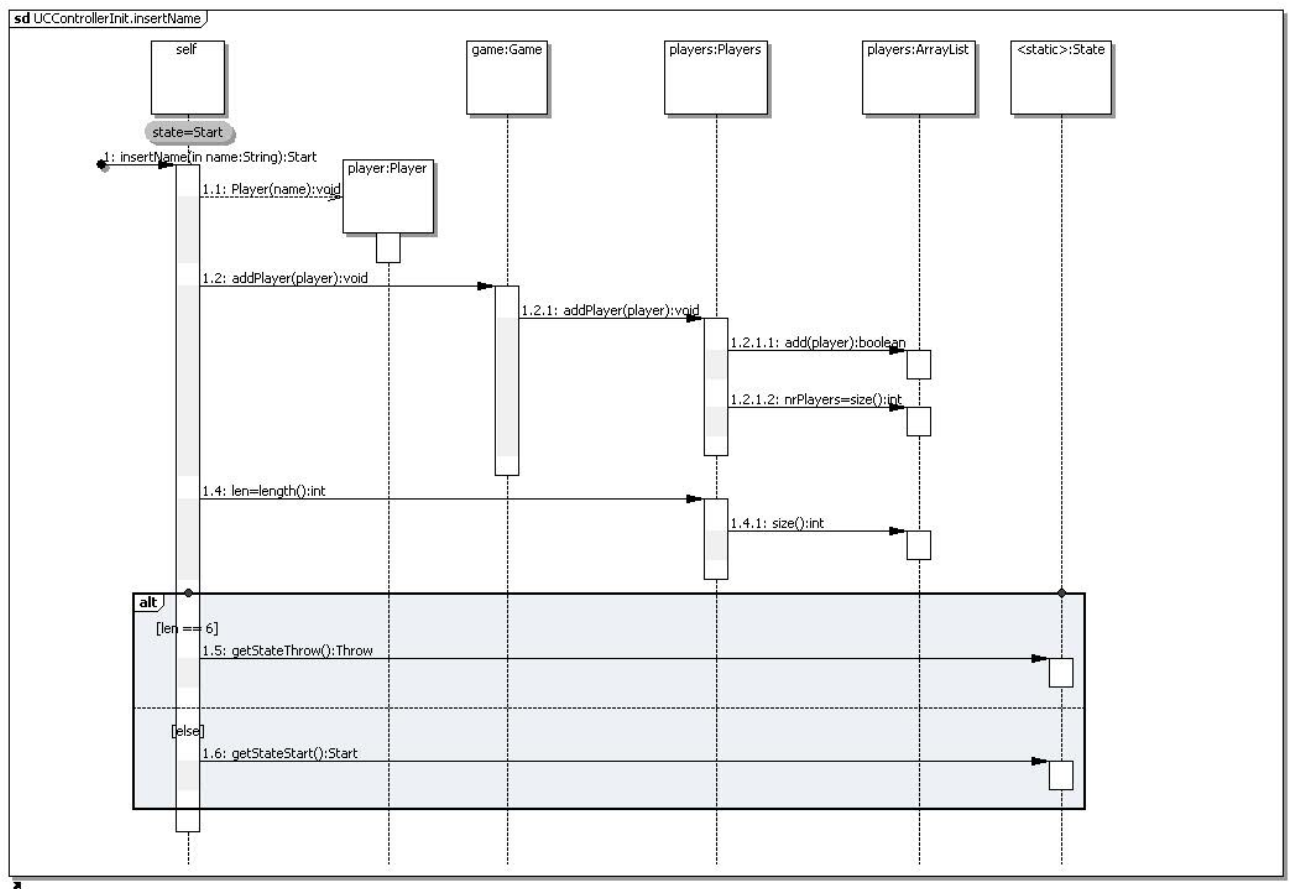
## 7 Applikationsschicht

## 7.1 Die detaillierten Systemoperationen des Kniffelstarts

### 7.1.1 SYSOP-ST1 newGame():State



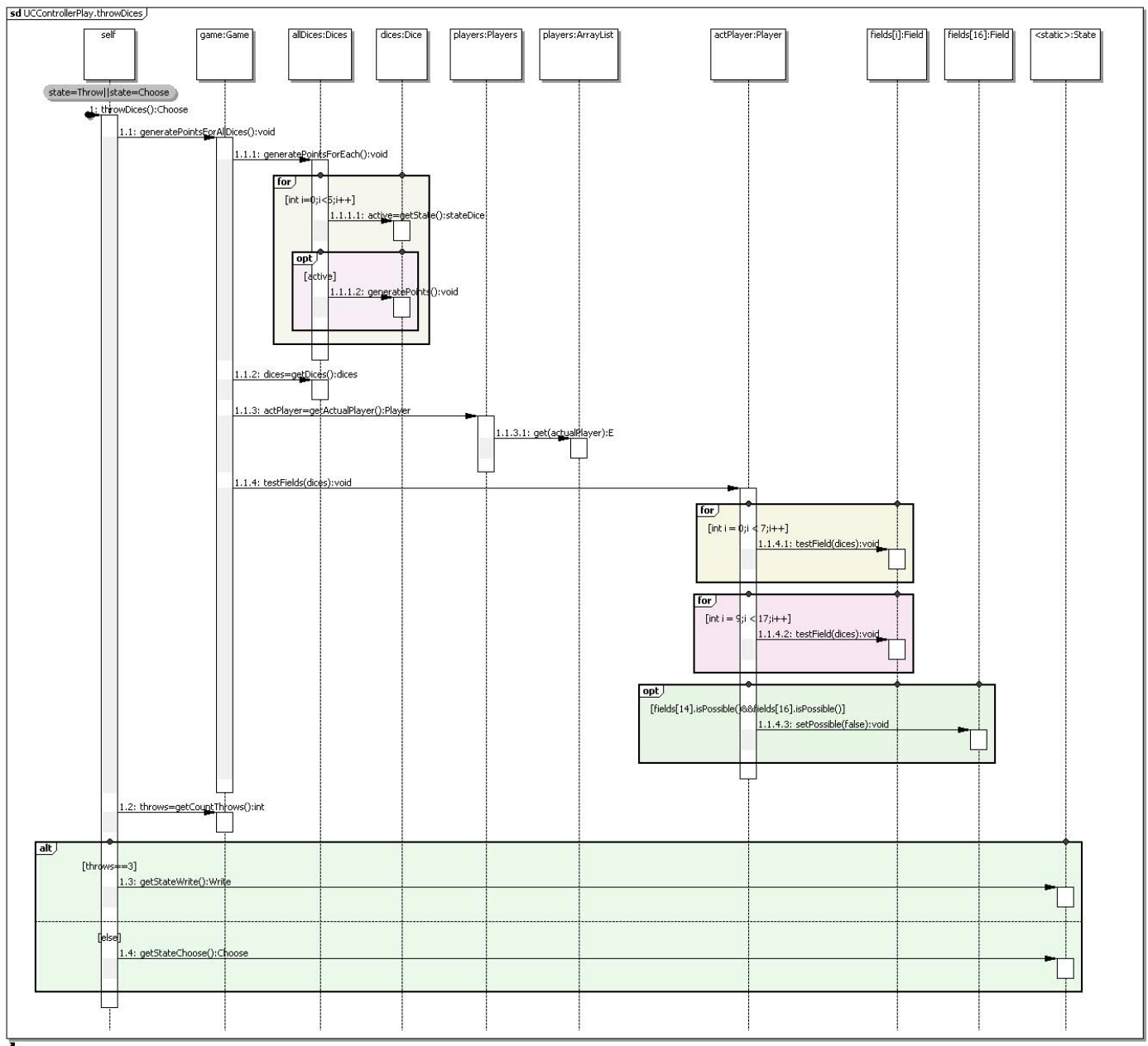
## 7.1.2 SYSOP-ST2 insertName(name: String):State



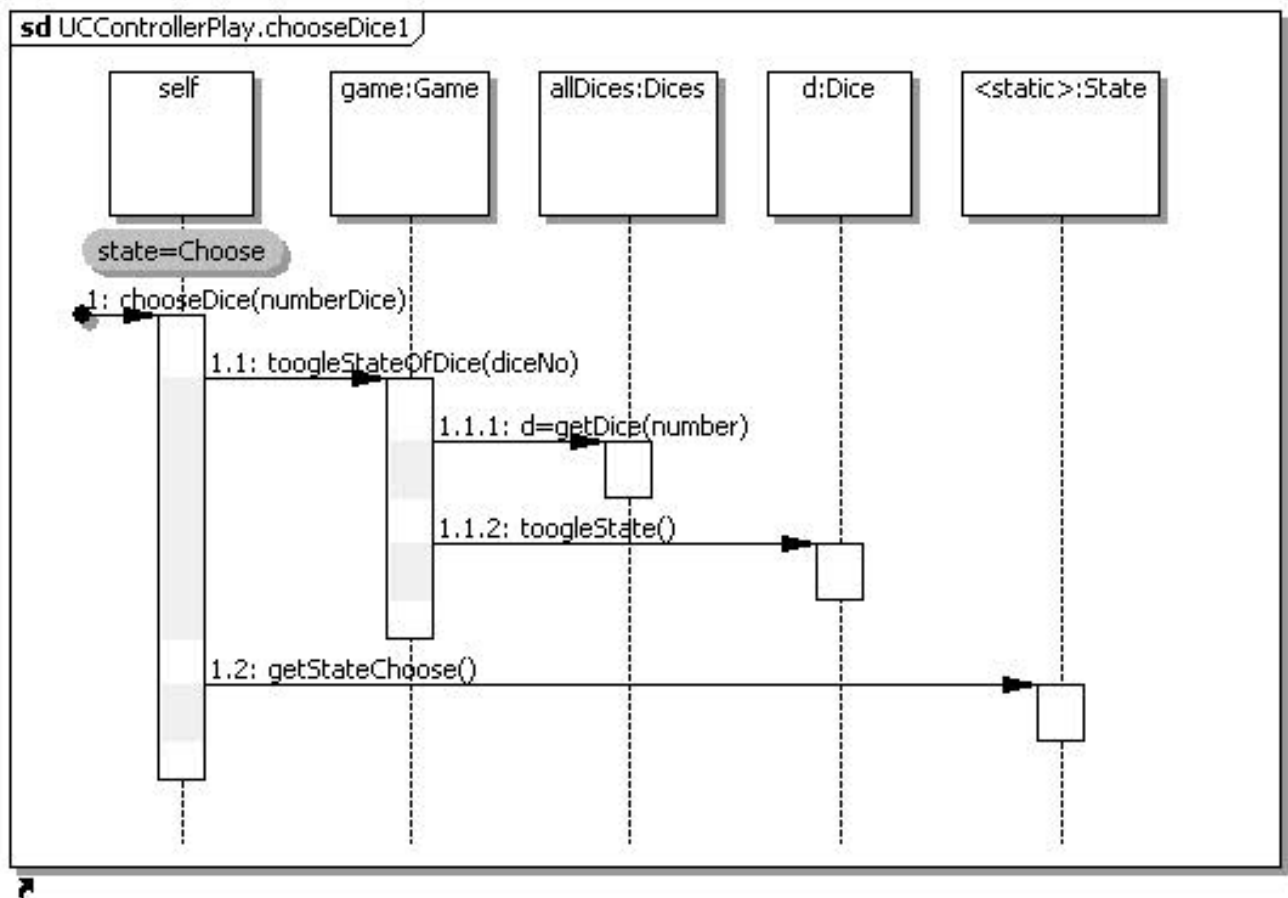


## 7.2 Die detaillierten Systemoperationen des Kniffelspielzugs

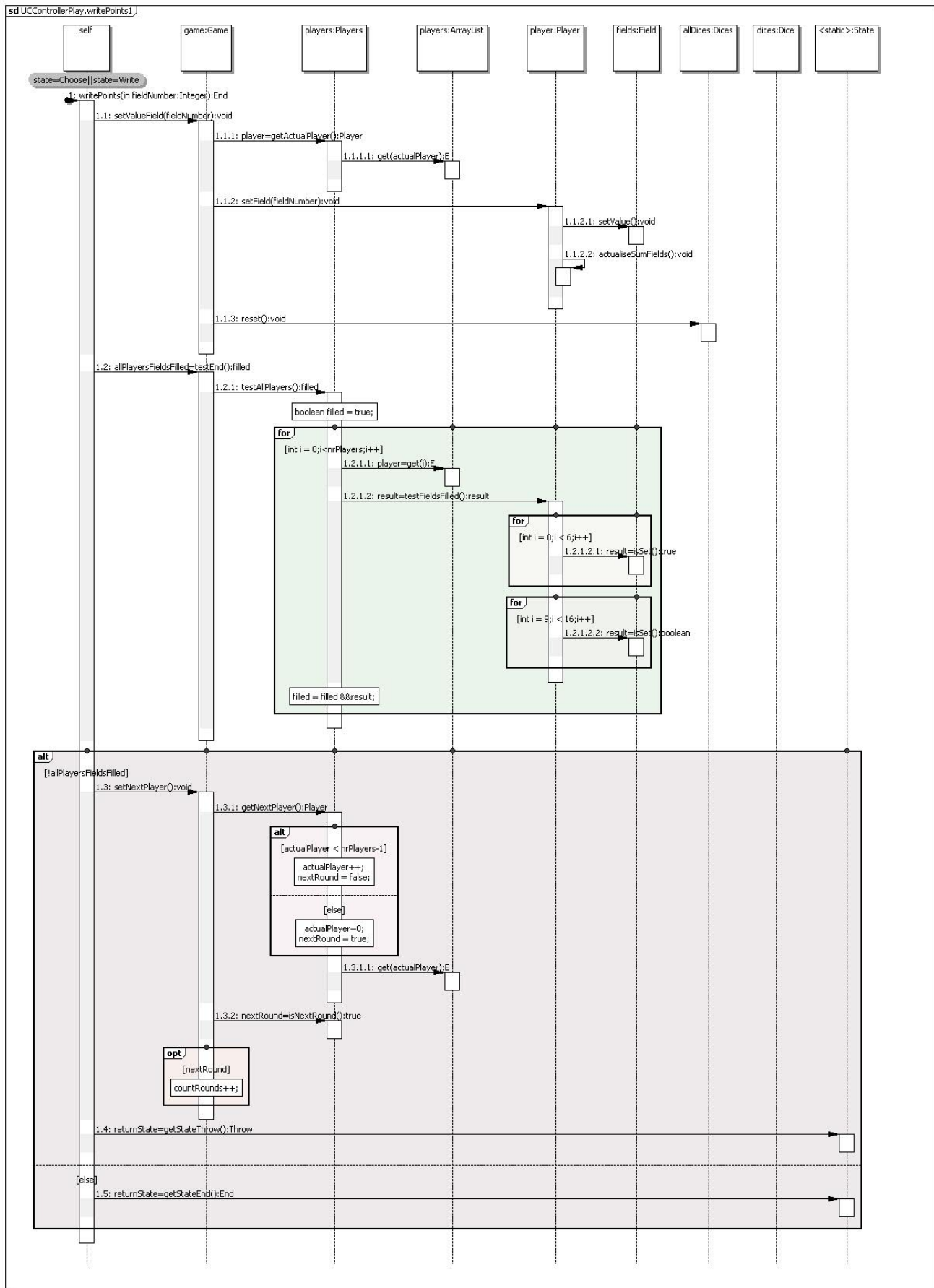
### 7.2.1 SYSOP-SP1 throwDices():State



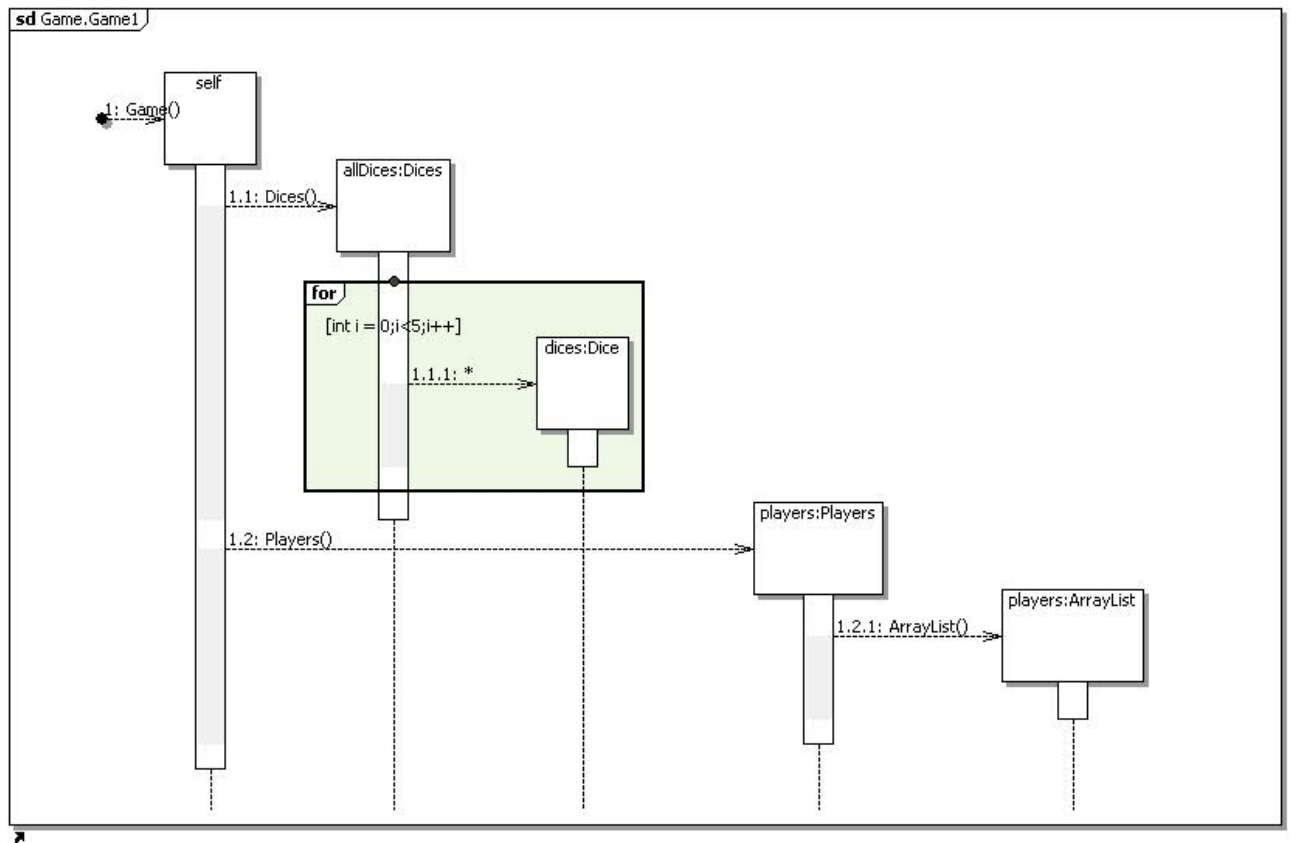
### 7.2.2 SYSOP-SP2 chooseDice(numberDice:int):State



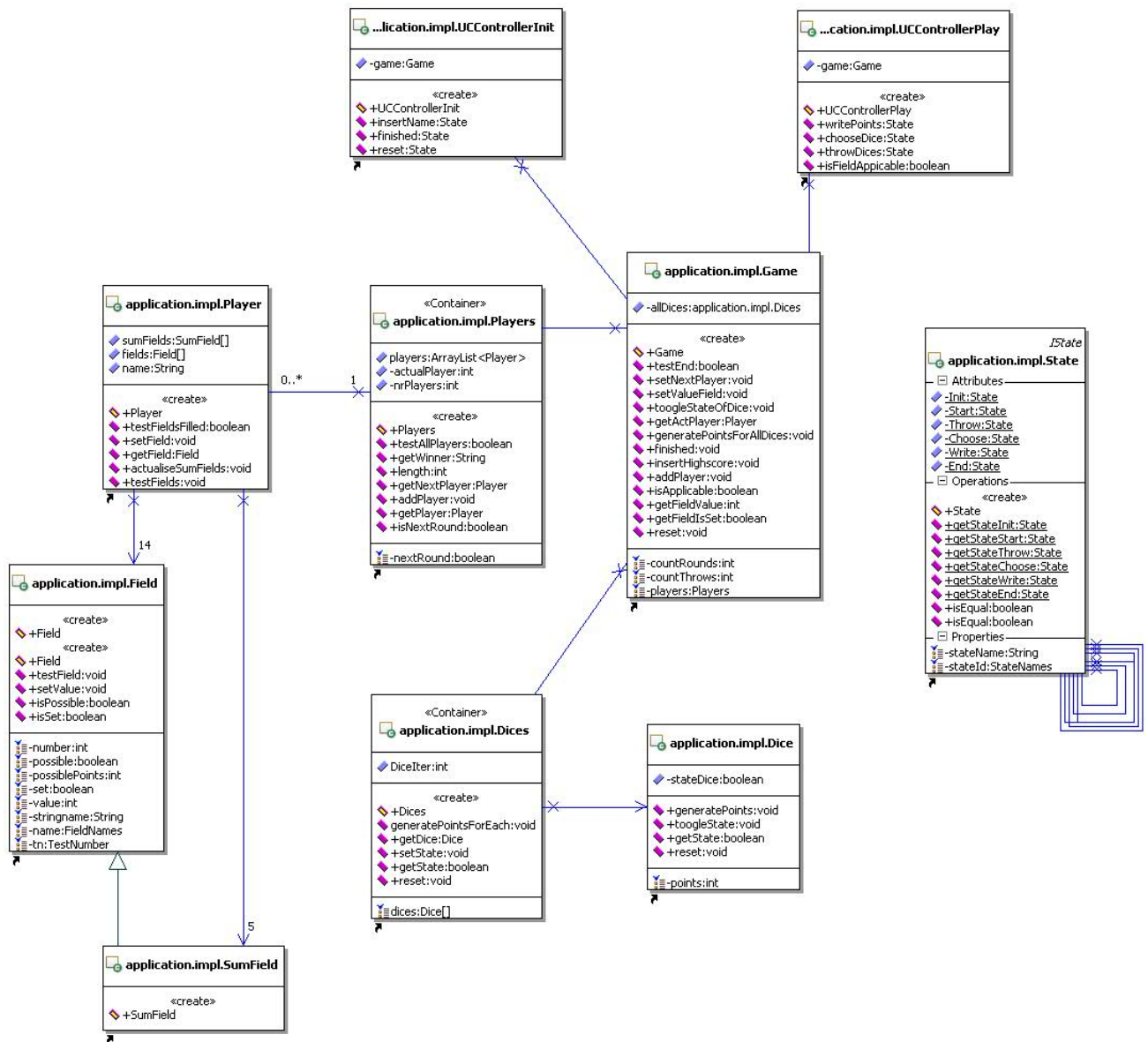
## 7.2.3 SYSOP-SP3 writePoints(fieldnumber:IFieldnumber):void



### 7.3 Die detaillierte Systeminitialisierung



## 8 Das Objektmodell



## 9 Anbindung der Oberfläche

Zustand Init: Dialog „neues Spiel beginnen“

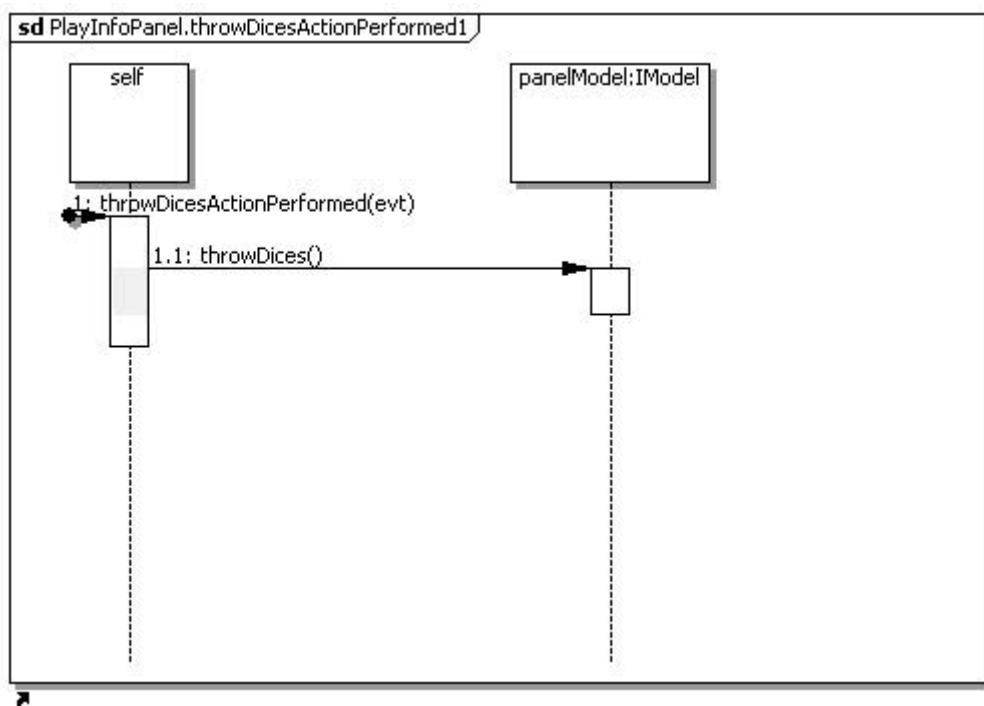
Zustand Start: Dialog „Spielernamen eingeben“

Zustand Throw: Würfel-Button ist aktiviert, die einzelnen Würfel sind deaktiviert, die Punktefelder sind deaktiviert

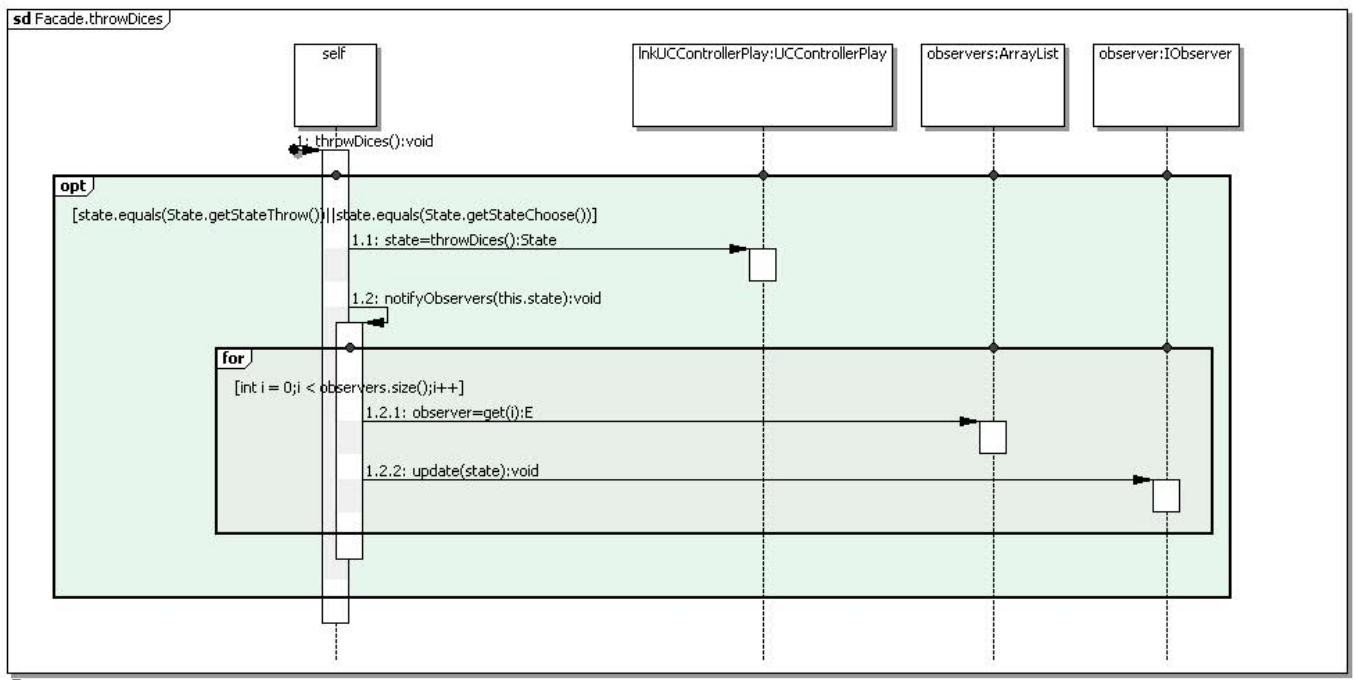
Zustand Choose: Würfel-Button ist aktiviert, die einzelnen Würfel sind aktiviert, die Punktefelder sind aktiviert, falls noch nicht gesetzt

Zustand Write: Würfel-Button ist deaktiviert, die einzelnen Würfel sind aktiviert, die Punktefelder sind aktiviert, falls noch nicht gesetzt

Anbindung der Oberfläche am Beispiel der Würfel-Aktion



Durch das Klicken des Würfel-Buttons wird der ActionListener `throwDicesActionPerformed` ausgelöst, dieser ruft im Modell die Methode `throwDices` auf.



Das Modell wird durch die Klasse Facade implementiert und dort die Methode throwDices aufgerufen. In dieser Methode wird überprüft, ob der Zustand der Anwendung dem für die Operation geforderten Zustand entspricht, falls ja wird die Operation im UseCaseControllerPlay aufgerufen. Anschließend werden die registrierten Beobachter über die Methode update über den neuen Zustand benachrichtigt.