DiGaL

Artur Sterz, Jonas Höchst, Andreas Morgen

17. Juli 2015

Inhaltsverzeichnis

Kapitel 1

Würfelspiele

1.1 Einordnung und geschichtlicher Hintergrund

Würfelspiele sind Glücksspiele, die im Wesentlichen mit Spielsteinen, sogenannten *Würfeln* gespielt werden. Dabei besteht jedes Spiel aus einem oder mehreren Würfeln, die nacheinander oder gleichzeitig geworfen werden, um ein bestimmtes Ergebnis zu erzielen. Von den Spielern werden kombinatorische Fähigkeiten verlangt.

1.2 Wichtige Konzepte

Für unsere Sprache sind im Wesentlichen drei Konstrukte wichtig, die im nachfolgenden erläutert werden.

1.2.1 Spieler

Das erste wichtige Konzept ist der *Spieler*. Den Spieler zeichnet aus, dass er die ausführende Kraft bei einem Spiel ist. Er würfelt mit den Würfeln, bei Entscheidungen muss er diese treffen und auch die Punkteverwaltung hat er inne.

Damit kommen wir auch auf den nächsten Punkt. Der Spieler benötigt Variablen, in denen aktuelle Werte gespeichert werden. Dies ist zwar sehr technisch ausgedrückt, beschreibt aber im Wesentlichen das, was "Spieler haben Punkte" bedeutet.

Und schließlich können Spieler aktiv oder inaktiv sein. Beispielsweise kann ein Spieler auf Grund seiner auf 0 reduzierten Punkte aus dem Spiel ausscheiden, er kann aber auch, wie bei dem Spiel "UNO Würfel" bei einer bestimmten Augenzahl eine Runde aussetzen.

1.2.2 Würfel

Ein weiterer Kernbestandteil sind die Würfel. Dabei kann ein Spiel mehrere Würfel haben, jedoch mindestens einen. Desweiteren hat ein Würfel sogenannte *Augen*. Das bedeutet, dass einer bestimmten Seite ein numerischer Wert zugeordnet wird. Dabei hat ein klassischer Würfel sechs Seiten, ist also von 1 bis 6 durchnummeriert. Es gibt jedoch Würfel mit einer beliebigen Anzahl an Seiten, mindestens jedoch zwei.

Außerdem muss erwähnt werden, dass es Spiele gibt, die keine numerischen Werte besitzen, sondern mit Piktogrammen bedruckt sind. Diese finden in unserer Sprache jedoch keine Anwendung, da diese Bildern auch mit einem numerischen Wert kodieren werden können.

1.2.3 Spielverlauf

Das letzte domänenspezifische Konzept, das in unserer Sprache enthalten sein muss ist der Spielverlauf. Im wesentlichen sind damit die Regeln gemeint, die beschreiben, was ein Spieler zu tun

hat, wenn er an der Reihe ist.

Dabei werden zum einen Aktionen definiert, wie zum Beispiel Würfel werfen. Zum anderen werden aber auch Konditionen geprüft, die erfüllt sein müssen, damit eine Aktion ausgeführt wird. So muss beispielsweise bei dem Spiel *Mäxchen* die Augenzahl 21 gewürfelt werden, damit allen anderen Spielern ein Punkt abgezogen werden kann.

1.3 Sprachziele

Das oberste Ziel unserer Sprache war es sie so einfach zu gestalten, so dass auch Personen, die keine Programmiererfahrung haben, den vorliegenden Quellcode verstehen können. Die Syntax ist dabei anders als bei herkömmlichen Sprachen und wird hauptsächlich durch Worte ausgedrückt. Außerdem haben wir versucht die Domäne so genau wie möglich in unserer Sprache abzubilden. Wie wir diese Ziele erreicht haben und welche Kompromisse wir dabei eingehen mussten, wird in den kommenden Kapiteln 2 und 3 diskutiert.

Kapitel 2

Die Sprache

2.1 Grundidee

Bestandteile eines Würfelspiel sind neben den benötigten Materialien wie den Würfeln vor Allem das Regelwerk. Dieses enthält alle Informationen, die benötigt werden, um das Spiel korrekt, also im Sinne des Erfinders, zu spielen. Dabei besteht das Regelwerk aus folgenden Bestandteilen:

- Anzahl der Spieler,
- Voraussetzungen für das Spiel,
- Aktionen der Spieler,
- Spielziel und Bedingungen für das Spielende und,
- Bewertungsgrundlagen

Unser Ziel war es nun DiGaL so zu gestalten, dass es als Regelwerk eines Spiels erkannt und auch von allen so gelesen werden kann. Damit dies deutlich wird, soll ein Beispiel gegeben werden:

2.1.1 Ein erstes Beispiel

Um einen ersten Eindruck der Sprache zu erhalten, soll hier zunächst ein Beispielprogramm angegeben werden:

```
EINS wird so gespielt:

das spiel ist für 2 bis 10 spieler geeignet.
das spiel hat folgende würfel:
würfel A hat diese seiten: 1 2 3 4 5 6
würfel B hat diese seiten: 1 2 3 4 5 6.

spieler haben die werte PUNKTE ist 0.
spieler haben die werte GEWONNEN ist 0.

ist ein spieler am zug macht er folgendes:
würfelt mit allen würfeln.
rechter spieler ist dran, wenn würfel 0 gleich 1 oder würfel 1
   gleich 1.
aktueller spieler PUNKTE ist aktueller spieler PUNKTE + die summe
   von allen würfeln.
```

```
wenn aktueller spieler PUNKTE größergleich 100, dann setze
   aktueller spieler GEWONNEN auf 1 und spiel ist zu ende.
rechter spieler ist dran.

gewonnen hat der spieler, bei dem GEWONNEN gleich 1.
```

2.2 Kontrollierte Sprache

Das oben gezeigte Beispiel, gibt einen ersten Eindruck der Syntax unserer Sprache. Wir haben versucht DiGaL an die natürlichen Sprache anzulehnen. Es ist unmöglich bzw. ein langer Prozess, alle Eigenheiten der deutschen Sprache zu implementieren und abzubilden. Ein Kompromiss aus unserem Design Goal und der Umsetzbarkeit ist eine kontrollierte Sprache: eine Sprache, die auf der einen Seite zwar intuitiv verständlich ist, auf der anderen Seite jedoch die deutsche Sprache so weit einschränkt, dass sie sinnvoll zu analysieren und grammatisch beschreibbar ist. Das Deklinieren und Konjugieren von Worten ist für das intuitive Verstehen einer Sprache aus unserer Sicht von großer Bedeutung und trägt maßgeblich zum Lesefluss bei. Im obigen Beispiel sei dazu auf auf die Verwendung von `alle würfel` verwiesen, der im Beispiel schon in seiner deklinierten Variante `allen würfeln` verwendet wird. Auch die Satzstellung ist für das Verständnis einer natürlichen Sprache wichtig. Sätze einer kontrollierten Sprache können schnell sperrig wirken. Unsere Sprache bietet daher verschiedene Optionen um das gleiche auszudrücken. Hier sei als Beispiel das `wenn` Konstrukt von oben angegeben, dass folgenden beiden Varianten valide ist:

```
rechter spieler ist dran, wenn würfel 0 gleich 1 oder würfel 1
   gleich 1.
wenn würfel 0 gleich 1 oder würfel 1 gleich 1, dann rechter
   spieler ist dran.
```

Designziel der Syntax unserer Sprache ist es zunächst, möglichst viele Varianten eines gegebenen Konstruktes anzubieten, in der Implementierung zeigen sich jedoch Schwierigkeiten, auf die in Kapitel **TODO: Ref. einsetzen** näher eingegangen wird.

2.3 Satzzeichen

Wie auch in der natürlichen Sprache, verwenden wir in unserer kontrollierten Sprache Satzzeichen um Mehrdeutigkeit für den Interpretierenden zu verhindern. Die Leser unserer Sprache bestehen nicht nur aus Personen, sondern insbesondere auch vom Parser unserer Implementierung. Die Verwendung von Satzzeichen ist daher obligatorisch.

2.3.1 Punkt (.)

Punkte schließen Sätze der natürlichen Sprache ab. Unsere kontrollierte Sprache übernimmt dieses Paradigma, um Anweisungen in unserer Sprache abzuschließen.

2.3.2 Doppelpunkt (:)

Der Doppelpunkt leitet Aufzählungen ein. Das erste mal tritt nach dem initialen Satz auf und Leitet die Aufzählung der Definitionsblöcke ein, danach dient er zum Beispiel noch zur Aufzählung er Würfel, deren Seiten, oder der Aktionen, die ein Spieler vollzieht. Der Doppelpunkt dient eher der Übersichtlichkeit als der Eindeutigkeit.

2.3.3 Komma (,)

Das Komma der natürlichen Sprache trennt Teilsätze ab, die semantisch zusammengehören. In unserer Sprache dienen Kommas dazu mehrere Anweisungen aufzuzählen, und `dann` und `sonst`

Anweisungen von bedingten Anweisungen abzugrenzen, und die Eindeutigkeit der Programme zu erhalten.

2.3.4 Semikolon (;)

Wir verwenden das Semikolon, um den Schleifenrumpf von weiteren folgenden Anweisungen abzutrennen. Dies ist ebenfalls nötig, um die Eindeutigkeit der Sprache zu gewährleisten.

2.3.5 Variablen

Unsere Sprache bietet dem Anwender die domänenspezifischen Objekte *Spieler* und *Würfel*, Variationen davon (*linker/rechter Spieler, Würfel 1, ...*) als Konzept der Sprache an. Darüber hinaus ist es möglich globale Variablen zu definieren, sowie Variablen pro Spieler festzulegen. Spielabhängige Variablen werden in Großbuchstaben geschrieben, um die Trennung von der Syntax zu verdeutlichen (siehe 2.5).

2.4 Blöcke und Absätze

Absätze haben in der natürlichen Sprache den Zweck Abschnitte, die einen eigenen Sinnzusammenhang oder ein eigenes kleines Thema haben von einander abzugrenzen. Dieses Paradigma haben wir auch in der Syntax von DiGaL übernommen. So haben Programm, die in DiGaL geschrieben sind vier Absätze, die nach ihrem Sinn unterteilt sind.

Zunächst beginnt ein Programm mit einer Überschrift. Anschließend kommt ein Abschnitt, der die Spielinitialisierung übernimmt. Es werden globale Variablen, Anzahl und Art der Würfel und die Anzahl der Spieler festgelegt. Auch eine Bedingung, wie lange das Spiel läuft, wird hier definiert.

Anschließend folgt ein Abschnitt, in dem die Spieler initialisiert werden, zum Beispiel Spielerspezifische Variablen oder eine Bedingung, wann ein Spieler aktiv ist.

Danach folgt der Abschnitt, in dem die Regeln festgelegt werden, was geschehen soll, wenn ein Spieler an der Reihe ist.

Letztlich gibt es eine Art Schlusssatz, der eine Bedingung definiert, wann ein Spieler gewonnen hat.

2.5 Groß- und Kleinschreibung

In diesem Punkt weichen wir bewusst von der natürlichen deutschen Sprache ab. Es wäre möglich auch hier die Regeln der deutschen Sprache einzuführen. Das hätte jedoch negative Auswirkungen auf die Lesbarkeit eines Programms. Da DiGaL eine Programmiersprache ist, bei der es wie bei anderen Programmiersprachen darum geht den Zustand eines Programms zu ändern und das im Wesentlichen über Variablen funktioniert, haben wir uns dazu entschieden Variablenbezeichner komplett groß (z.B. `EINS` oder `PUNKTE` in obigem Beispiel) und den Rest komplett klein zu schreiben. Dies erhöht die Lesbarkeit eines Programms und vereinfacht es so Programmierern ein valides und korrektes Programm zu schreiben.

TODO: Syntax Tree visualisieren

Kapitel 3

Designentscheidungen

Kapitel 4

Diskussion

Kapitel 5

Erfahrungen