

Ausgabe: Mittwoch, 02.06.10

Abgabe: Freitag, 18.06.10

3. Übungsblatt: Backgammon in LOS

Um sinnvolle Einsatzmöglichkeiten für Aspekte in der objektorientierten Programmierung kennen zu lernen, implementiert ihr auf Basis unserer Lua **Objekt Sprache** das Brettspiel Backgammon in einer einfachen Form für zwei menschliche Spieler. Dabei handelt es sich um ein Brettspiel mit schwarzen und weissen Steinen und zwei Würfeln. Aspekte dienen in unserer Implementierung der Einhaltung bestimmter Interaktionsregeln und sollen den Spielverlauf dokumentieren.

Spielregeln

Die Backgammon Spielregeln sind zum Beispiel auf Wikipedia erklärt:

<http://de.wikipedia.org/wiki/Backgammon>

Die open source Simulation GNU Backgammon ist hier zu finden:

<http://www.gnubg.org/>

Die Nummerierung der Points, Bereiche der Home- und Outerboards und die Startaufstellung entsprechen in unserem Backgammon dem Wikipedia Artikel. Es werden die Farben schwarz und weiss für die Steine und Spieler verwendet.

Folgende Abweichungen gelten für unser Spiel:

- Es wird ohne Dopplerwürfel gespielt, eine Verdoppelung ist nicht möglich.
- Es macht keinen Unterschied, ob man einen Pasch würfelt oder nicht. Es gibt immer nur maximal zwei Steinbewegungen pro Spielzug.
- Es gibt keinen Eröffnungswurf. Es beginnt immer der Spieler mit der Farbe schwarz.
- Es muss nicht versucht werden, beide Würfelwerte zu ziehen. Zu jedem Zeitpunkt kann der Spielzug beendet werden und der nächste Spieler ist dran.
- Ein ungültiger Zug kann in Abhängigkeit des unten beschriebenen Aspektes wiederholt werden, unabhängig vom Wunsch des Gegners.

- Beim Auswürfeln reicht ein Würfelwert, der größer ist als die Distanz zur 0 bzw. 25, um einen Stein herauszuwürfeln. Innerhalb des Homeboards darf trotzdem gezogen werden.
- Gewonnen hat der Spieler, der als erster seine Steine herausgewürfelt hat. Es gibt keine unterschiedlichen Gewinnstufen.
- Es sind keine der Sonderregeln aus dem Wikipediaartikel zu beachten.

Darstellung

Das aktuelle Spielbrett ist durch Nummerierung der Points und der darauf befindlichen Steine in einer Textausgabe darzustellen. Dabei werden schwarze Steine durch ein "B" und weisse Steine durch ein "W" repräsentiert. Unter dem Spiel werden in einer Zeile die geschlagenen Steine dargestellt und in einer weiteren der aktive Spieler und die zur Verfügung stehenden gewürfelten Werte sowie die vergangene Spielzeit.

```

13 14 15 16 17 18 | 19 20 21 22 23 24
B           W     | W           B
B           W     | W           B
B           W     | W
B           | W
B           | W

W           | B
W           | B
W           B   | B
W           B   | B           W
W           B   | B           W
12 11 10 09 08 07 | 06 05 04 03 02 01

```

Geschlagene Steine: keine

Schwarz ist am Zug und hat nutzbare Wuerfelwerte: 6 1 , Spielzeit 0 s

Interaktion

Das Spiel soll auf zwei Wegen gesteuert werden können: Durch Konsoleneingaben oder durch Methodenaufrufe. Als erstes soll aus dem Lua Code heraus die Methode `:start(true/false)` aufgerufen werden, wobei der Parameter die Benutzerinteraktion über die Konsole ein und ausschaltet.

Ist die Benutzerinteraktion eingeschaltet, sind abwechselnd der schwarze und weisse Spieler an der Reihe. Zu Beginn und nach jeder Steinversetzung ist das aktuelle Spielfeld darzustellen, eine Liste der geschlagenen Steine und welcher Spieler am Zug ist mit seinen zur Verfügung stehenden, gewürfelten Werten. Die Würfelwerte werden beim Spielerwechsel zufällig ermittelt. Dazu genügt die Lua Bibliotheksfunktion `math.random(6)`.

Der Benutzer hat folgende Eingabemöglichkeiten:

- Ein Stein wird durch Eingabe *Startpoint Leerzeichen Zielpoint* versetzt, z.B. *12 9*.
 - Einstellige Points werden einstellig eingegeben.
 - Geschlagene Steine werden mit Startpoint 0 wieder ins Spiel geholt.
 - Auswürfelzielpoint für schwarz ist 0, für weiss 25.
- Eingabe *restart* startet das Spiel neu.
- Eingabe *end* beendet das Spiel und das Programm.
- Eingabe *next* beendet den Spielzug, der nächste Spieler ist an der Reihe.
- Eingabe *check* und *log* schalten die jeweiligen Aspekte ein und aus

Eine Meldung soll am Ende des Spiels den Sieger bekannt geben.

Ist die Benutzerinteraktion ausgeschaltet, können die Spielzüge wie oben beschrieben nur durch einen Methodenaufruf der Form `:moveStone('3 6')` ausgeführt werden. Das Spielbrett mit anhängenden Informationen wird nur nach explizitem Aufruf einer `:print()` Methode im aktuellen Zustand dargestellt. Es können die aktuellen Würfelwerte durch einen Methodenaufruf der Form `:setDice(3,4)` manuell gesetzt werden. Das Ende des Spiels wird mit einer Siegesmeldung bekannt gegeben. Ein Neustart ist durch Aufruf der Methode `:restart()` möglich.

Aspekt zur Fehlerbehandlung

Es soll ein Aspekt implementiert werden, der die Konsoleneingabe und Methodenaufrufparameter auf Syntaxfehler prüft und unerlaubte Spielzüge erkennt und verhindert. Entsprechend soll eine eigene Fehlermeldung sowohl bei aktivem als auch bei inaktivem Benutzerinteraktionsmodus in die Konsole ausgegeben werden, die den Programmfluss nicht unterbricht, sondern eine erneute Eingabe ermöglicht. Nennt diesen Aspekt **MoveChecker**. Es soll möglich sein, mit `MoveChecker:enable()` oder `MoveChecker:disable()` oder im Interaktionsmodus durch die Eingabe von *check* Syntaxprüfung und Regelprüfung während des laufenden Spiels ein- und auszuschalten. Für die Einhaltung der Syntax- und Spielregeln und der damit verbundenen eventuell auftretenden Fehler ist im ausgeschalteten

Zustand der Benutzer verantwortlich. Es muss möglich sein, mit ausgeschaltetem Aspekt Steine regelwidrig zu verschieben und damit gezielt Spielsituationen hervorzurufen. Gebt ihr für unterschiedliche Regelverletzungen unterschiedliche und auf die konkrete Regelverletzung angepasste Fehlermeldungen heraus, gibt das einen Bonuspunkt gegenüber einer einzigen Fehlermeldung für alle Verstöße.

Dokumentation des Spielablaufs

Es soll ein weiterer Aspekt implementiert werden, der den Spielablauf dokumentiert. Dabei sollen nur ausgeführte, gültige Spielzüge aufgenommen werden und die Zeit dokumentiert werden, die seit Spielbeginn vergangen ist. Das Logging soll während des laufenden Spiels ein- und ausschaltbar sein und den Spielverlauf nicht beeinflussen. Geloggt werden soll in eine Textdatei mit dem Namen `BGLog.txt` im Ausführungsverzeichnis Ihres Programms. Ein Logeintrag besteht aus einer Zeile für eine erfolgreich ausgeführte Steinbewegung oder einen Spielerwechsel mit seinen gewürfelten Werten, jeweils mit der Zeit. Ist der MoveChecker Aspekt ausgeschaltet und der MoveLogger Aspekt eingeschaltet, dürfen alle ausgeführten Spielzüge geloggt werden. Beispiel für Log-Einträge:

Schwarz würfelt eine 3 und eine 2, Zeit 10s

Schwarz setzt einen Stein von Point 24 nach Point 22, Zeit: 23s

Schwarz setzt einen Stein von Point 8 nach Point 5, Zeit: 37s

Weiss würfelt eine 5 und eine 6, Zeit 45s

Werden geschlagene Steine wieder ins Spiel gebracht, kann als Quelle "von Point 0" angegeben werden, für das Herauswürfeln eines Steines als Ziel "nach Point 0" für Schwarz bzw. "nach Point 25" für Weiß.

Das Ende des Spiels durch einen Sieg oder durch Eingabe von *end* im interaktiven Modus muss nicht geloggt werden. Ein Neustart soll durch einen Eintrag zu erkennen sein. Das Logfile muss nie geleert werden, auch nach Neustart der Anwendung oder des Spiels kann ans Ende des Logfiles angefügt werden.

Nennt diesen Aspekt `MoveLogger`. Es soll möglich sein, mit `MoveLogger:enable()` oder `MoveLogger:disable()` oder im Interaktionsmodus durch die Eingabe von *log* das Logging ein- und auszuschalten.

Zum Erstellen eines Textfiles kann die Lua io Library hilfreich sein:
<http://lua-users.org/wiki/IOLibraryTutorial>

Hinweise

Es ist erlaubt (und sicher hilfreich), neben den Standardtypen der LOS String, Number und Boolean auch den Lua Typ `table` bei der Attributdeklaration von Klassen und Aspekten zuzulassen.

Achtet auf die Nutzung des Doppelpunktes für Methodenaufrufe an Objekten und die Verwendung des Schlüsselwortes `self` mit einem Punkt für Objektattribute.

Bearbeitung und Abgabe

Die Aufgabe soll in Gruppen von 3 Personen bearbeitet werden. Vorgegebene Methoden- und Aspektnamen und die Eingabesyntax sind genau einzuhalten. Die Abgabe erfolgt in eurem Gruppenforum. Legt dazu einen neuen Beitrag mit dem Betreff *Abgabe Übung 3* an und fügt die Dateien gepackt in einem einzelnen `zip`-Archiv als Anhang ein. Im Kopf jeder Datei müssen (als Kommentar) Namen und Matrikelnummern aller aktiv mitarbeitenden Gruppenmitglieder stehen.