



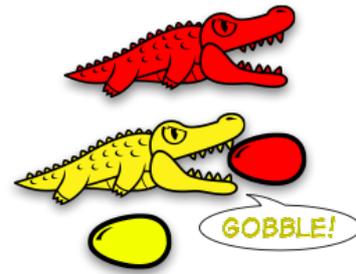
Praxis der Software-Entwicklung 2013/14

Alligatoren lehren Lambda

Prof. Dr. Gregor Snelting – Andreas Zwinkau
Lehrstuhl Programmierparadigmen
Karlsruher Institut für Technologie (KIT)

Stand: 30. Oktober 2013

In diesem Praktikum sollen Sie eine Android-Lernapp realisieren, bei der eine graphische Variante des Lambdakalküls gelehrt wird. Hierbei stellen Alligatoren λ -ausdrücke und Eier die Variablen dar. Der Ausdruck rechts entspricht beispielsweise $\lambda x.(\lambda y.y)x$, wobei x durch Rot und y durch Gelb dargestellt ist.



Eine kurze Bemerkung vorab

Dies ist IHR Projekt. Dieses Dokument ist kein Katalog von Aufgaben, der Punkt für Punkt abgearbeitet werden muss, um nachher den Schein zu bekommen, sondern lediglich eine Reihe von Hinweisen, was wir erwarten. Wie IHR Programm nachher aussieht, müssen SIE selbst entscheiden.

1 Motivation

Der λ -Kalkül ist *die* Grundlage funktionaler Programmierung. Funktionale Programmierung wird gerade in den letzten Jahren wichtiger, da Parallelisierung auf Multicorearchitekturen unter Beachtung von funktionalen Prinzipien (no side effects, higher-order functions) einfacher ist. An Einfachheit mit einer Turing-Maschine vergleichbar, bietet bereits der einfache untypisierte λ -Kalkül gute Möglichkeiten zu Abstraktion an, so dass erstaunlich gut damit programmiert werden kann.

Die formale Notation ($\lambda x.x$) und die Mächtigkeit täuschen eine hohe Komplexität vor. Im Kern ist der λ -Kalkül, aber eigentlich so einfach, dass selbst ein Grundschüler in verstehen kann. Eine Darstellung durch Alligatoren erleichtert den Zugang. Begriffe wie „Alligatorfamilie“ statt „ λ -Term“ sind für Kinder vertrauter und bauen Hürden ab.

Ein Spiel¹ von Bret Victor zeigt eine spielerische Variante. Unter anderem Philip Wadlers 8-jährige Tochter

¹<http://worrydream.com/AlligatorEggs/>

konnte erfolgreich die Beispielaufgabe lösen. Die Verpackung als Puzzlespiel verwandelt trockenen Stoff in eine spannende Herausforderung.

2 Aufgabe

Am Ende soll eine funktionsfähige Android App stehen, die spielerisch den λ -Kalkül für Grundschüler vermittelt.

2.1 Spielregeln

Abweichungen von den offiziellen Spielregeln sind möglich. Offen ist auch die Frage, wie das Programm mit Termination umgeht. Der λ -Kalkül erlaubt potentiell unendlich laufende Programme und gerade der hier relevant sogenannte Y-Kombinator ist ein interessantes Konstrukt. Insbesondere könnten weitere Konzepte vermittelt werden, wie Call-by-Value, Call-by-name, Call-by-reference, Lazyness, oder typisierter λ -Kalkül.

Der λ -Kalkül ist für weitergehende Spielereien benutzbar. Möglich wäre also auch eine Art Sandbox-Modus, in dem der Anwender relativ frei selbst programmieren kann. Das entfernt sich allerdings von dem Spielprinzip und ist rein optional zu sehen.

2.2 Android

Android existiert mittlerweile in der Version 4.3 (Jelly Bean), allerdings ist auch Version 4.0 (Ice Cream Sandwich) noch recht verbreitet. Sie sollten bei Ihrer Entwicklung daher mindestens Version 4.0+ unterstützen. Optional kann auch Android 2.3 (Gingerbread) unterstützt werden, welches immer noch fast 30% Marktanteil hat. Das übliche Interface für Android ist Touch, weshalb dieses auch unterstützt werden muss. Andere Modi sind optional.

2.3 Spaß

Der Spaß ist wichtig, um Kinder länger für das Lernprogramm zu motivieren. Kleine Belohnungen wie beispielsweise Erfolgsanimationen oder Highscores sind eine Möglichkeit. Recheriere doch mal nach dem Stichwort „gamification“.

Wichtig in diesem Punkt ist auch das Interface, das altersgerecht und intuitiv sein sollte. Bedienkonzepte müssen von Grundschulern (und Grundschullehrern) verstanden werden können. Animierte Übergänge statt ruckartiger Bewegung ist ein Aspekt. Ein gute Wahl für Symbole ein anderer.

2.4 Kontrolle

Der Lernfortschritt soll durch Eltern bzw. Lehrer überprüfbar sein. Im einfachsten Fall könnte man sehen, dass ein Kind es bis Level 42 geschafft hat. Komplexere Varianten wären das Übermitteln an einen Server oder eine Fehlerquote bei Wiederholungs- bzw Zufallsaufgaben.

Doch bevor der Erreichen von Lernzielen geprüft werden kann, müssen diese erstmal definiert werden. Mögli-

che Lernziele wären

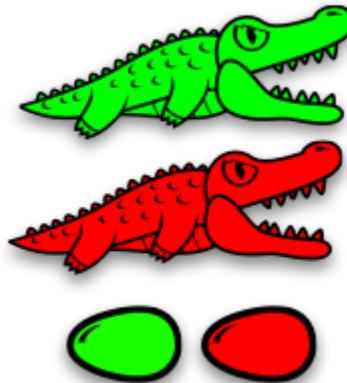
- Einfache Ausdrücke
- Unendliche Auswertung
- Funktionen Höherer Ordnung
- Bindung

Ideen finden sich beispielsweise auch in unserer Vorlesung „Programmierparadigmen“.

2.5 Minimale Leistungsmerkmale

Folgende minimale Leistungsmerkmale soll Ihr Programm erfüllen:

- muss in der Simulationsumgebung von Android laufen
- komplett über das dort simulierte Touchinterface bedienbar sein
- eine modulare Architektur besitzen (z.B. Model-View-Controller-Prinzip)



3 Organisatorisches

Für jede Phase des Praktikums muss ein Phasendokument abgegeben werden. Dieses Dokument ist Grundlage für das Kolloquium am Ende jeder Phase, in dem die Gruppe die Ergebnisse der Phase vorträgt.

3.1 Zeitlicher Ablauf

Phase	von – bis	Dauer
Auftaktveranstaltung	22.10.	
WebInScribe	22.10. – 25.10.	
Erstes Gruppentreffen	30.10.	
Pflichtenheft	04.11. – 24.11.	3 Wochen
Entwurf	25.11. – 22.12.	4 Wochen
Implementierung	23.12. – 02.02.	4 + 2 Wochen
incl. Weihnachtsferien	23.12. – 05.01.	
Qualitätssicherung	03.02. – 23.02.	3 Wochen
Klausurpause	24.02. – 09.03.	
interne Abnahme	10.03. – 16.03.	
Abschlusspräsentation	17.03. – 23.03.	

3.2 Erste Schritte

- Einarbeitung Android
- Einarbeitung λ -Kalkül
- Werkzeuge sichten
 - Versionsverwaltung: Subversion? Git?
 - IDE: Eclipse?
 - Dokumente: \LaTeX ? Office? UML-Diagramme?
 - Testing: JUnit? EMMA Coverage Testing?
- Funktionale Anforderungen überlegen
 - Markt sichten
 - Muss-Kriterien
 - Wunsch-Kriterien
 - Abgrenzungs-Kriterien

4 Bewertung

Die Benotung Ihres Systems richtet sich nach folgenden Kriterien:

- Qualität aller abgegebenen Dokumente
- Qualität der Kolloquien (10min Vortrag)
- Qualität der Abschlusspräsentation
- Erfüllung der minimalen Leistungsmerkmale (s.o.)

- *Sinnvolle* Erweiterungen über diese Merkmale hinaus
- Robustheit des erstellten Programms
- Teamarbeit (TSE)

Diese Liste hat keine Reihenfolge, die einer Gewichtung entspricht. Es gibt sicherlich weitere Punkte, die als selbstverständlich gelten und sich bei Nichterfüllen negativ auswirken (z.B. würde eine völlig unintuitive Bedienbarkeit des Programms zu Abzügen führen).

4.1 Weitere Pflichten

- Anmeldung per Studienportal für PSE und TSE
- Anwesenheit bei den wöchentlichen Treffen mit dem Betreuer (Prüfungsleistung)
- Team-internes Management (z.B. Phasenverantwortliche, Versionskontrolle)
- Artefakte 1–2 Tage vor Kolloquium beim Betreuer

5 Literatur

Wir empfehlen folgende Quellen zur Vorbereitung auf die Aufgabe.

- Unsere Dokumente und Folien
- Softwaretechnik I Vorlesung von Prof. Tichy
- Spielregeln: <http://worrydream.com/AlligatorEggs/>
- Interpreter: <http://metatoys.org/alligator/>
- Android: developer.android.com
- Wikipedia, Google, ...