

Projekte der Veranstaltung “Fortgeschrittene funktionale Programmierung in Haskell”

Bestehenskriterien

Zum Bestehen reicht es, wenn Projekte/Aufgaben/Teilaufgaben im Wert von *100* Punkten zufriedenstellend abgegeben wurden. Mindestens ein Projekt muss dabei *50* Punkte oder mehr ausmachen.

Die Punkte sind unterteilt in Kategorien, aus denen nicht mehr als die angegebene Anzahl berücksichtigt wird:

- Parser: 30 Punkte
- div. Libraries: 40 Punkte
- “Core”-Haskell: 50 Punkte (“Core”-Haskell umfasst die Libraries `lens`, `mtl` bzw. `transformers`, `attoparsec` oder andere Parser, `gloss` oder andere GUI)
- Visualisierung: 50 Punkte
- Parallelism/Concurrency: 30 Punkte
- Binärparsing: 20 Punkte

Dies dient dazu, dass man sich nicht nur auf eine Sache konzentriert, sondern verschiedene Aspekte und deren Zusammenspiel verstanden haben muss.

Ebenfalls sind diese Projekte alle offen gehalten. Wenn ihnen eine sinnvolle Erweiterung einfällt, sprechen sie die Tutoren an und diese Erweiterung wird in dieses Dokument aufgenommen und eine Bepunktung hierfür festgelegt.

Projekte

Von den Projekten “DisCaptive” und “Snake” darf nur eines gewählt werden.

DisCaptive (Summe: 80 Punkte)

OPJ-Spiel “DisCaptive”

Anforderungen an das Ergebnis analog zu OPJ. Interne Struktur, verwendete Bibliotheken etc. sind allerdings freigestellt.

Bepunktung

- Visualisierung: 10 Punkte (Visualisierung)
- Level-Parser: 10 Punkte (Parser)

- Benutzung von State: 15 Punkte (Core)

Optionale Zusätze:

- Verwendung von Lens: 5 Punkte (Core)
- Einlesen/Anzeigen von Bildern mittels Library: 5 Punkte (div. Libraries)
- Binärparser für Bitmaps und anzeigen: 15 Punkte (Binärparsing)
- Kraftfelder/Türen: 2 Punkte (Core) + 3 Punkte (Visualisierung)
- Vereister Boden: 2 Punkte (Core) + 3 Punkte (Visualisierung)
- Sensor: 2 Punkte (Core) + 3 Punkte (Visualisierung)
- Schwere Kisten: 2 Punkte (Core) + 3 Punkte (Visualisierung)

Snake (Summe: 95)

Snake mit verschiedenen Level programmieren.

Das Level muss hierzu eingelesen und dargestellt werden.

Ablauf: Wenn kein Essen auf dem Spielfeld zu sehen ist, dann erscheint auf einem freien Feld etwas zu Essen. Die Schlange kann gesteuert werden, läuft aber automatisch nach vorn, wenn keine Taste gedrückt wurde. Wenn der Kopf der Schlange Essen aufnimmt, dann verschwindet es und die Schlange beginnt zu wachsen. Ziel des Spiel ist es möglichst viele Punkte zu sammeln, bevor die Schlange vor die Wand oder sich selbst läuft.

Wenn ein Level an der Außenseite keine Wand hat, dann kommt die Schlange auf der anderen Seite des Levels wieder herein.

Multiplayer-Ablauf: Jeder Spieler steuert eine Schlange. Tote Schlangen verschwinden vom Spielfeld. Wer zuletzt überlebt, hat gewonnen.

Bepunktung

- Visualisierung: 10 Punkte (Visualisierung)
- Level-Parser: 5 Punkte (Parser)

Optionale Zusätze:

- Benutzung von State: 15 Punkte (Core)
- Verwendung von Lens: 5 Punkte (Core)
- Einlesen/Anzeigen von Bildern mittels Library: 5 Punkte (div. Libraries)
- Binärparser für Bitmaps und anzeigen: 15 Punkte (Binärparsing)
- Multiplayer an einem PC: 5 Punkte (Core)
- Multiplayer über Netzwerk: 5 Punkte (Concurrency) + 5 Punkte (Binärparsing) + 10 Punkte (Core)
- Dedizierter Multiplayer-Server: 15 Punkte (Concurrency)

Datamining (Summe: 50 Punkte)

Es wird Ihnen eine CSV-Datei des Haushaltsplanes der Landesregierung NRW von 2015 gegeben. Lesen die Datei ein und visualisieren Sie diese in einer geeigneten Weise. Für eine optionale interaktive Visualisierung gibt es die 20 optionalen Visualisierungspunkte.

Bepunktung

- Parsing von CSV-Daten mit `,` und `\n`: 3 Punkte (Parser)
- Verifikation der Daten: 2 Punkte (Parser)
- Erstellung der Visualisierung: 15 Punkte (Visualisierung)

Optionale Zusätze:

- Erkennen/Angabe von Feldtrennern und Zeilentrennern: 5 Punkte (Parser)
- Sinnvolle interaktive Visualisierung: 20 Punkte (Visualisierung) - z.B. wie <https://haushalt.piratenfraktion-nrw.de/>
- Daten im JSON-Format annehmen: 5 Punkte (Parser) bei selbst-schreiben, 5 Punkte (div. Libraries) bei Nutzung von externen Libs.

Verteiltes Rechnen (Summe: 68 Punkte)

Dieses Projekt entspricht dem Betriebssysteme-Projekt des rechnenden Servers. Ihr Programm soll einen Port öffnen und dort Verbindungen akzeptieren. Diese schicken "Aufträge" (simple Arithmetik mit `+-*/`) an den Server, der einige Zeit brauchen soll, diese zu bearbeiten (2-30 Sekunden). Man kann mehrere "Jobs" starten, sich nach ihrem Zustand erkundigen, die Ergebnisse abholen oder auf Ergebnisse warten.

Bepunktung

- Minimal-Server: 15 Punkte (Parallelism/Concurrency)

Optionale Zusätze:

- Parsing der Eingaben mittels Parser: 3 Punkte (Parser)
- Verwendung von Lenses: 5 Punkte (Core)
- Verteiltes Rechnen: 15 Punkte (Parallelism/Concurrency) + 30 Punkte (div. Libraries) bei Verwendung von CloudHaskell - man rechnet nicht auf einem Server, sondern auf einer Cloud

Jabber-Bot (Summe: 60 Punkte)

Das XMPP-Protokoll (auch Jabber genannt) ist eine offene Schnittstelle. Über die Techfak bekommen sie einen eigenen Jabber-Account (username@techfak.de). Schreiben sie einen Bot, der einfache Aufgaben in einem Jabber-Chat (oder Konferenz) übernimmt.

Aufgaben:

- Remember-me: Der Bot reagiert auf ein Kommando der Form “remind me at 13:45: go eat in Mensa” und um 13:45 schreibt der Bot den Absender mit der Nachricht “go eat in Mensa” an.
- Wikipedia-suche: Der Bot reagiert auf ein Kommando der Form “wiki universität bielefeld” und sucht auf Wikipedia nach “universität bielefeld”, nimmt den ersten Treffer und antwortet mit dem ersten Absatz des Artikels
- Kalender-Plugin: Das ekVV stellt den persönlichen Kalender im caldav-Format zur Verfügung. Laden sie diesen zum Start/auf Wunsch ein und senden sie vor jedem Event eine Nachricht an den jeweiligen User.
- Verteiltes-Rechnen-Plugin: Wenn sie das Projekt “verteilttes Rechnen” gemacht haben, dann sollen sie auch in der Lage sein über ihren Bot mit dem jeweiligen Server zu reden (und so Aufträge absenden, abholen etc.).
- Webprojekt-Plugin: Wenn das Webprojekt ebenfalls gemacht wurde, dann soll dieser Bot in der Lage sein private Nachrichten an den jeweiligen User weiterzuleiten.

Bepunktung

- Aufsetzen des Bots: 15 Punkte (div. Libraries)
- Remember-Me-Aufgabe: 5 Punkte (Core)

Optionale Zusätze:

- Wikipedia-Suche-Aufgabe: 10 Punkte (Core) + 5 Punkte (div. Libraries) bei Verwendung von hexpat-lens zum Antwort-Parsen
- Kalender-Plugin:
 - Parser für ekvv-Kalender: 10 Punkte (Parser) oder 10 Punkte (div. Libraries)
 - Implementation Erinnerungsfunktion: 5 Punkte (Core)
- Verteiltes Rechnen-Plugin:
 - Bot baut eine Verbindung zum Projekt “Verteiltes Rechnen” auf und leitet Inzut/Output durch: 5 Punkte (Parser) + 5 Punkte (Core)
- “Webprojekt”-Plugin:
 - Notifications vom Forum per Jabber verstehen und an den angegeben User weiterleiten: 5 Punkte (Core)

Webprojekt (Summe: 75 Punkte)

Erstellen sie ein simples Forum. Die Registrierung kann entweder über Email/Passwort oder auch per Plugins, wie “OAuth” (Google, Facebook, Github, Twitter, Twitch, ...) ermöglicht werden.

Das Admin-Panel soll umfassen: - Hinzufügen/Entfernen von Foren - Moderations-Fähigkeiten - Rechteverwaltung (Benennen von “Moderatoren”, einschränken der Rechte auf Unterforen)

Bepunktung

- Minimal-Forum: 25 Punkte (div. Libraries) + 5 Punkte (Core)
- Registrierung: 2 Punkte (div. Libraries)

Optionale Zusätze:

- Gute responsive Visualisierung mittels Bootstrap (o.ä.): 20 Punkte (Visualisierung)
- Upload von Avataren und deren Darstellung: 3 Punkte (div. Libraries)
- Admin-Panel: 10 Punkte (Core)
- Notifications per E-Mail: 5 Punkte (Core)
- Notifications per Jabber: 5 Punkte (div. Libraries)

Abgabedatum: 28.8.2016

Projekte werden vorzugsweise als Projekt auf Github oder über euer **Remote-Folder** abgegeben. Wir helfen in den Tutorien auch bei der Benutzung von git bzw. bei der Rechtevergabe im Remote-Verzeichnis.