6. Übungsblatt

Ausgabe: 24.01.05

Abgabe: 08/09.02.05

13 Haskell on the Road

20 Punkte

In dieser Aufgabe geht es darum, den in der Vorlesung vorgestellten Haskell-Fahrsimulator zu einem markttauglichen Rennspiel weiterzuentwickeln. Mit den Erlösen des Spiels (die Markteinführung ist für den 10.02.2005 geplant, der Stand auf der CeBIT ist schon gemietet) soll die Frühpension des Veranstaltungspersonals finanziert werden.

1. Zuerst erweitern wir den Fahrsimulator um eine Gangschaltung.

Der Zustand des Autos wird um den eingelegten Gang erweitert. Die neue Geschwindigkeit ergibt sich jetzt (sofern nicht gebremst wird) nicht wie vorher einfach durch $v' = v + a * \Delta t$, sondern jetzt hat jeder Gang eine jeweilige Mindestgeschwindigkeit, ab der in diesem Gang beschleunigt wird, und eine Höchstgeschwindigkeit, bis zu der in diesem Gang beschleunigt wird.

Der Gangwechsel erfolgt durch die Mausknöpfe (links für den nächstniedrigeren, rechts für den nächsthöheren Gang). Der eingelegte Gang soll auf dem Bildschirm angezeigt werden. (4 Punkte)

2. Im zweiten Schritt erweitern wir den Fahrsimulator um Hindernisse.

Dabei soll es *stationäre* Hindernisse geben, die eine am Anfang entweder vorgegebene oder zufällig berechnete Position beibehalten, und *mobile* Hindernisse (zum Beispiel andere Autos), die sich bewegen. Kollidiert das Auto mit einem Hindernis, ist das Spiel vorbei, der Benutzer drückt eine Taste, und das Spiel startet von vorne.

Die Hindernisse werden weitere Komponenten des globalen Zustands. Mobile Hindernisse werden genau wie das Auto bei jedem Zustandsübergang bewegt. Ferner muss bei jedem Zustandsübergang geprüft werden, ob eine Kollision vorliegt. (8 Punkte)

3. Im dritten Schritt fügen wir das Rennelement hinzu.

Entwerfen Sie dazu eine Rennstrecke, d.h. entweder einen von stationären Hindernissen eingegrenzten Parcours, oder eine Folge von Landmarken, die passiert werden müssen (z.B. Tore, durch die gefahren werden muss). Beim Start des Parcours wird eine Uhr gestartet, und beim Passieren des Ziels die Zeit genommen. Die Uhrzeit soll in der Statuszeile am unteren Bildschirmrand angezeigt werden. (8 Punkte)

Hinweis: Für die Zeitnahme ist das Standardmodul `Time` mit den Funktionen `getClockTime` und `diffClockTimes` und dem Datentyp `TimeDiff` hilfreich.

4. Ansonsten sind Ihrem Einfallsreichtum keine Grenzen gesetzt– von Bestzeitenlisten über Mehrspielerversionen und einem realistischeren Kurvenverhalten sind noch viele Erweiterungen möglich. *(bis zu 10 Bonuspunkten)*