



7363 - Web-basierte Anwendungen: **09: Wettkampf-Simulation**

Modellierung
Empfohlene Parameter



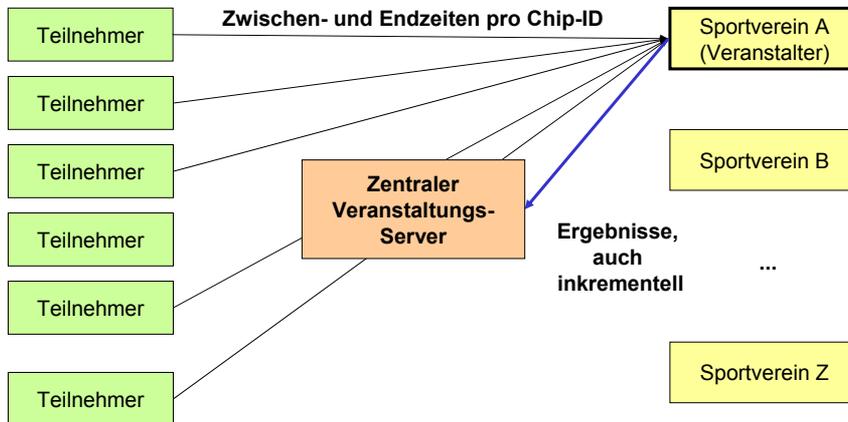
Ziele



- Übungen
 - Keine Übungen mehr, sondern Projektimplementierung!
- Ziele
 - Vermittlung physikalischer Grundlagen für die Implementierung der Wettkampf-Simulation
 - Parameter-Empfehlungen



- Das Szenario: Wettkampf



- Starterfeld: Jeder Starter mit Eigenschaften

- x Aktuelle Position in Meter
- v Aktuelle Geschwindigkeit in m/s
- chipID
- startNr
- weitere Attribute ...

- Simulation einer Zeitscheibe Δt :

Für $t = 0, \Delta t, 2 * \Delta t, \dots t_{\max}$:

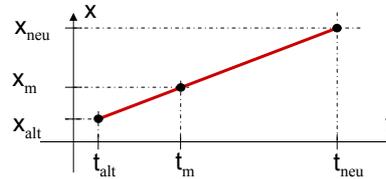
- Für alle Starter:
 - $x \rightarrow x + v * \Delta t$
 - $v \rightarrow v + \Delta v(\dots)$ # Zu Δv siehe unten
 - Zwischenzeit(en) ermitteln
- $t += \Delta t$
- sleep Δt_{sim} # z.B. 30 sec für simulierte 5 Minuten



- Zwischenzeitmessung: Einfach durch lineare Interpolation

Sei x_m ein Messpunkt, etwa $x_m = 20000.0$ (km-Marke 20)

- if $x_{alt} \leq x_m$ and $x_{neu} > x_m$
 $t_m = t_{alt} + (x_m - x_{alt}) / v$
- (chipID, t_m , x_m) melden



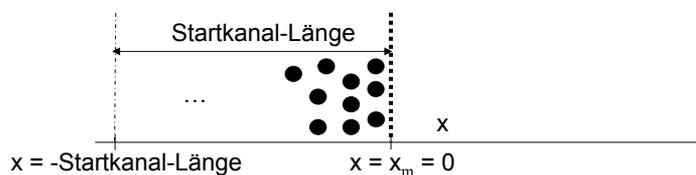
- Hinweise

- Innerhalb einer Zeitscheibe können auch mehrere x_m durchlaufen werden, etwa km 20 und 21.1 (Halbmarathon) innerhalb 5 Min. Berücksichtigen Sie diesen Fall!
- Damit nicht für jeden Starter für jede Zeitscheibe jede Messmarke getestet werden muss:
 - Sortieren Sie die Messmarken nach ihrer Entfernung
 - Führen Sie pro Starter einen Index, welche Messmarke als nächste erwartet wird, und testen Sie erst ab dieser Marke.
 - Keine weiteren Tests, wenn Index > letzte Marke (im Ziel!)



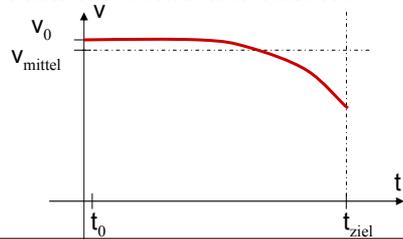
- Netto- und Bruttozeiten

- Initialisieren Sie die Starterpositionen x auf Zufallswerte aus einem Intervall [-Startkanal-Länge, 0.0]
- Startschuss = $t_0 := 0$
- Erster Messpunkt = Startlinie
 - Für dieses m gilt: $x_m = 0$
- Gemessen & gemeldet werden nur Bruttozeiten
 - Nettozeiten: $t_{m, net} = t_m - t_{x=0}$





- Variation der Geschwindigkeiten
 - Variante 1: Individuell verschiedenes, aber konstantes v
 - $\Delta v = 0$. Langweilig!
 - Variante 2: v variiert in jeder Zeitscheibe um Zufallswert
 - $\Delta v = \Delta v_{\max} * (2 * \text{rand} - 1)$ # rand = Zufallszahl aus $[0, 1]$
 - Einfach, aber unrealistisch
 - Variante 3: v erst konstant, dann immer schneller fallend
 - z.B. per Parabel modelliert
 - $\Delta v = \Delta v(t)$ oder $\Delta v = \Delta v(x)$
 - Realistischer
 - Für Ambitionierte!



- Modellparameter (Empfehlungen)
 - **Anzahl Starter:** **ca. 1000**
 - Nur wenige müssen namentlich gemeldet sein,
 - aber für alle sollten Startnr. und Chip-ID bekannt sein
 - **Länge des Startkanals:** **100 ... 500 m**
 - Geschwindigkeiten
 - $v_{\min} = \text{ca. } 8 \text{ km/h}$, $v_{\max} = \text{ca. } 18 \text{ km/h}$, $\Delta v_{\max} = 1 \text{ km/h}$
 - Erinnerung: Durch 3.6 teilen ergibt m/s
 - Zeiten
 - $\Delta t = 5 \text{ min} = 300 \text{ s}$, $t_{\max} = 5 \text{ h}$ (entspricht ca. $42.195 \text{ km} / v_{\min}$)
 - $\Delta t_{\text{sim}} = 30 \text{ s} = \Delta t / 10$.
 - Ein 5-Stunden-Wettkampf wird also in 30 min. simuliert. Das reicht für unsere Zwecke während der Abnahme.
 - Zum Testen empfohlen: $\Delta t_{\text{sim}} = 5 \text{ s}$



- *On-line Demo*
 - Ruby-Programm laufsиму.rb:
 - Diskussion des Quellcodes
 - Demo-Lauf

- Hinweis für Ihre Projektabnahme
 - Nach Demonstration von Vereinsregistrierung, ..., Teilnehmermeldungen sollten Sie einen Wettkampf simulieren.
 - Während der Wettkampf läuft, werden Zwischenzeiten an den zentralen Server geleitet.
 - Zwischenergebnisse werden dann während des noch laufenden Wettkampfs (per Browser) abgerufen!
 - Sinn: Verfolgung des Rennverlaufs !