

Erweiterung der Simulation - der Trödelfaktor

Aktuell herrscht eine ziemlich perfekte Welt in der Simulation. Aber wie sieht es in der Realität aus? Gibt es da nicht diese Sonntagsfahrer, die einfach nicht Gas geben, obwohl vor ihnen alles frei ist?! Und dann gibt es die Außendienstler, die dem Vordermann schon fast im Kofferraum hängen und dann doch wieder scharf bremsen müssen...

Deshalb wurde das NaSch-Modell in seiner Grundversion um einen sogenannten Trödelfaktor erweitert. *Jedes Auto reduziert demnach seine Geschwindigkeit mit einer vorgegebenen Wahrscheinlichkeit um eins.* Das randomisierte NaSch-Modell sieht damit wie folgt aus:



Aufgabe 1:

Erweitert eure Simulation um den oben beschriebenen Trödelfaktor. Dieser wird später im Applet selbst vor dem Start der Simulation eingestellt. Die Wahrscheinlichkeit, mit der ein Auto trödelt, wird im Attribut `troedelfaktor` gespeichert, das Werte zwischen 0 und 100 annehmen kann.. Wie könnt ihr diesen zufälligen Trödelfaktor realisieren?

Ergänzt außerdem wieder eine Ausgabe für die Simulation, wenn ein Auto zufällig trödelt, damit ihr beim Testen das Verhalten der Autos immer noch nachvollziehen könnt.

Wie erzeugt man in Java Zufallszahlen?

Eine Zufallszahl von 0 bis 100 erzeugt man in *Java* mit den zwei Befehlen

```
Random zufall = new Random();  
int zufallszahl = zufall.nextInt(101);
```

Die Zahl 101 selbst ist dabei nicht mehr enthalten.

Aufgabe 2:

Interpretiert den Trödelfaktor im Vergleich zum realen Straßenverkehr. Welche verschiedenen Situationen kann er simulieren, wenn ...

- a. ein Auto in der vorherigen Generation mit Geschwindigkeit 0 gefahren ist?

- b. ein Auto in der nächsten Generation bremsen muss?
