

## 5. Übungsblatt

**Ausgabe:** 25.06.15

**Abgabe:** 14.07.15

Auf diesem finalen Aufgabenblatt bekommt unsere Robotersimulation den letzten Feinschliff, um die internationale Computerspielpresse vollends zu entzücken, und die Taschen potenzieller Risikokapitalgeber weit zu öffnen.

### 5.1 *Einer für alle - alle für einen!*

15 Punkte

Der letzte Schritt zum kommerziellen Erfolg liegt natürlich in ausgefeilten Mehrspielermodi und mehr Action! Erweitern sie daher ihre Implementierung von Übungsblatt 4 um Teams und Wasserpistolen (der Wunsch der Marketingabteilung nach „Lasern, Disruptoren und Neutronenbomben“ wurde aus ethischen Gründen verworfen).

*Wasserpistolen* können die empfindliche Elektronik der Roboter stören: Jeder Roboter verfügt über einen Wassertank aus dem er Sprühstöße in eine beliebige Richtung abgeben kann. Wird ein Roboter getroffen, fällt er eine zufällige Zeitspanne aus, bis die Elektronik wieder getrocknet ist. Selbstverständlich ist die Reichweite der Wasserpistolen auf ein paar Felder begrenzt. Solange ein Roboter wehrlos da steht, können sich andere Roboter an seinen Goldvorräten bedienen.

*Roboterteams* können sowohl aus menschlichen als auch aus computergesteuerten Mitspielern bestehen. Roboter in einem Team sollen Zugriff auf die Liste ihrer Mitspieler haben und ihnen Nachrichten schicken können. Für menschliche Teams ist ein Chatfenster in der Weboberfläche verfügbar, für computergesteuerte Teams bedeutet das, dass die Roboter-Aktoren beliebige Nachrichten untereinander austauschen können.

Erweitern sie außerdem das Verhalten der autonomen Roboter so, dass sie sich untereinander absprechen, um mehr Gold zu sammeln als die anderen Teams.

Um die Leistung der Teams bewerten zu können, sollen die Roboter eines Teams eine gemeinsame Basis haben, wo sie das geladene Gold zwischendurch immer wieder abladen können. Das Team mit dem größten Haufen Gold in der Basis gewinnt.

Die letzten 5 der 15 Punkte dieser Aufgabe gibt es für die Lösung mindestens einer der folgenden Aufgaben:

1. Geometrisch korrekte Darstellung des Schattenwurfs im Radar in der Webschnittstelle;
2. Roboter können sich gegenseitig Dilithium übertragen und damit liegengebliebene Teammitglieder wiederbeleben (inkl. entsprechendes autonomes Verhalten);
3. Vereinheitlichtes Interface und Wettkampf der autonomen Roboter mit einer oder mehreren anderen Übungsgruppen.

## 5.2 Nützliche Hülsenfrüchte

5 Punkte

Die Patientenkartei von Dr. Okulus gerät langsam aus den Fugen, daher entscheidet er sich schweren Herzens, sich nicht weiter der modernen Technik verschließen, und beauftragt uns seine Patientenkartei von nun an digital zu verwalten. Seine Sicherheitsbedenken können wir aber erst nach einem langen Beratungsgespräch aus der Welt räumen: Als wir ihm von *Linsen* erzählen willigt er ein — „so sind meine Daten sicher“, ist er zufrieden.

Ein Eintrag in der Patientendatenbank besteht aus folgenden Daten:

- Versicherungsstatus (Gesetzlich/Privat)
- Krankenversicherungsnummer
- Name, Vorname
- Geburtsdatum (bestehend aus Jahr, Monat, Tag)
- Adresse
  - Straße, Hausnummer
  - Postleitzahl
  - Ort
- Liste von Telefonnummern (jeweils kategorisiert als privat, mobil oder geschäftlich)
- Liste der Diagnosen
- Nächster Termin (Typ wie Geburtsdatum)

Implementieren sie die Patientendatenbank sowie drei verschiedene Schnittstellen mit Hilfe von Linsen:

- Der *Patient* kann nach Angabe von KV-Nummer und Geburtsdatum bequem seine Patientenakte einsehen. Um die Patienten nicht zu verunsichern sollen sie jedoch nicht die Liste ihrer Diagnosen einsehen können. Der Patient soll seine Kontaktdaten (Adresse/Telefonnummern) aktualisieren können.
- Die *Sprechstundenhilfe* kann mit Hilfe der Gesundheitskarte (Krankenversicherungsnummer) eine Patientenakte aus der Datenbank laden. Sie soll die Möglichkeit haben den nächsten Termin zu ändern.
- Der *überweisende Hausarzt* kann die Liste der Diagnosen einsehen und verändern.

Das Interface der Datenbank selbst besteht aus lediglich zwei Methoden:

```
def query[T] (lens: Lens [Database, T]): T
def updated[T] (lens: Lens [Database, T], value: T): Database
```

Der Inhalt der Datenbank ist also nur mit Linsen zugreifbar. Die Linsen bekommt man über die jeweiligen Schnittstellen für Patienten, Sprechstundenhilfen und Hausärzte. Alle Updates sollen mit Linsen realisiert werden. Es steht Ihnen frei, ob sie die Linsenimplementierung aus der Vorlesung oder die Linsen der *shapeless* Bücherei verwenden. Die *scalastyle*-Definition von Übungsblatt 1 soll hier ohne Warnungen durchlaufen.

*Hinweis:* Die aktuelle Version der *shapeless* Bücherei können Sie mit folgender Zeile in ein *sbt*-Projekt einbinden:

```
libraryDependencies += "com.chuusai" %% "shapeless" % "2.2.3"
```