

# Selbst-organisierende, adaptive Systeme (WS 19/20)

Übungsblatt 01 (Bearbeitung bis: 22.10.2019, 12:00 Uhr (mittags))

## Selbst-organisierende Systeme und NetLogo

### Aufgabe 1 (*Einrichten des Git Repositories*)

Zur Verwaltung der Quellcodes setzen wir GitHub ein.

- a) Melden Sie sich bitte auf <https://gitlab.isse.de> mit Ihrer RZ-Kennung und PW an und melden sich bei uns (E-Mail oder pers.), sodass wir Ihnen die nötigen Zugriffsrechte geben können. Sie finden Ihr Repository danach unter <https://gitlab.isse.de/lehre/soas/studenten/2019-WiSe/gruppex>, wobei x für Ihre Gruppennummer steht.
- b) Legen Sie Ihr Repository gemäß den Vorschlägen bei “Command line instructions” an. Beispielsweise:

```
git clone git@gitlab.isse.de:lehre/soas/studenten/2019-WiSe/gruppe01.git
cd gruppe01
touch README.md
git add README.md
```

- c) Legen Sie nun einen Ordner **Blatt-01** an und darin einen Ordner **Aufgabe-01**. Implementieren Sie darin Ihr erstes adaptives System in einer Programmiersprache Ihrer Wahl: Sie implementieren eine einfache Feedbackschleife in Form eines Modells eines Thermostats.
  - Als Eingabe dient eine Temperatur-Messreihe (also ein **Array** von **Floats**, z.B. `outsideTemp = new double[] {10.0, 25.0, 30.0};`) sowie minimale und maximale Temperatur (z.B. `double tMin = 15.0, tMax = 20.0;`) und initiale Temperatur (z.B. `current = 18.0;`).
  - Als Steueraktionen stehen Ihnen *kühlen* (-5.0 Grad), *heizen* (+5.0 Grad) oder *passiv* (0.0 Grad) zur Verfügung.
  - Berechnen Sie den Temperaturverlauf, indem Sie für jeden Zeitschritt die Mischlufttemperatur (Mittel zwischen Außentemperatur und Raumtemperatur) berechnen und abhängig davon die geeignete Kontrollaktion durchführen. Achten Sie darauf, dass die Raumtemperatur stets im “Korridor” zwischen `tMin` und `tMax` verbleibt.
  - Geben Sie den Temperaturverlauf aus.
- d) Fügen Sie die Projektordner zu Ihrem Repository hinzu und pushen dieses.

Alle Ihre Aufgaben werden mit Hilfe des Git-Repositories verwaltet.

Weitere Informationen zu Git und Versionsverwaltung finden Sie unter den folgenden Links:

- a) **git - the simple guide:** <http://rogerdudler.github.io/git-guide/>
- b) **GitHub Training:** <https://training.github.com/kit/>
- c) **GitHub for Windows:** <https://desktop.github.com/>

### Aufgabe 2 (*Modellierung selbst-organisierender Systeme*)

In der Vorlesung haben Sie bereits Beispiele für selbst-organisierende Systeme kennen gelernt. Viele dieser Systeme bestehen aus drei grundlegenden Komponenten: einzelnen Entitäten oder *Agenten*, die sich in einer gemeinsamen *Umwelt* befinden und deren Verhalten durch *Interaktionen* untereinander oder mit der Umwelt gesteuert wird. Diese Aufteilung eignet sich in vielen Fällen für eine Modellierung von selbst-organisierenden Systemen oder Systemen mit emergenten Eigenschaften.

- a) Beschreiben Sie mit eigenen Worten unter Angabe Ihrer Quellen<sup>1</sup> **zwei** selbst-organisierende und/oder emergente Systeme, die nicht als Beispiel in der Vorlesung genannt wurden. Gehen Sie insbesondere auf die Eigenschaften ein, an denen der Aspekt der Selbst-Organisation und/oder der Emergenz sichtbar wird. Verdeutlichen Sie außerdem, welche Definition Sie zugrunde legen.
- b) Beschreiben Sie dann, welche Agenten in diesen Systemen eine Rolle spielen, welchen Einfluss die Umwelt auf das System hat und welche Interaktionen zwischen den Agenten bzw. zwischen den Agenten und der Umwelt stattfindet.

### Aufgabe 3 (*NetLogo*)

Im weiteren Verlauf der Übung werden Sie eigene selbst-organisierende Systeme mit Hilfe von NetLogo modellieren und simulieren. Um sich mit der Simulationsumgebung und der Sprache NetLogo vertraut zu machen, installieren Sie NetLogo, dass Sie auf <http://ccl.northwestern.edu/netlogo/> für verschiedene Plattformen herunterladen können. Alternativ können Sie auf die vorinstallierte Version auf den Praktikumsrechnern zugreifen.

Unter <http://ccl.northwestern.edu/netlogo/docs/> finden Sie die Dokumentation von NetLogo. Lesen Sie sich die Einleitung durch und gehen Sie dann das dreiteilige Tutorial durch. Am Ende des Tutorials sollten Sie drei einfache Modelle kennengelernt haben und auch bereits eigenen Code für das Ökosystem-Modell geschrieben haben. Weitere Informationen zur Programmierung mit NetLogo finden Sie im „Programming Guide“. Eine Übersicht über die eingebauten NetLogo-Befehle gibt das „NetLogo Dictionary“.

Im Rahmen des 3. Tutorials haben Sie ein Modell erstellt, das ein Ökosystem aus Pflanzenfressern und nachwachsendem Futter beschreibt. In diesem Modell stellt sich nach einer Weile ein Gleichgewicht zwischen Turtles und Gras her, das relativ stabil über die Laufzeit des Systems bleibt.

- a) Erweitern Sie das Modell, indem Sie einen neuen Button **disease** mit einer entsprechenden Prozedur einführen. Die Krankheit befällt alle Turtles mit einer Energie von weniger als 30 und führt sofort zum Tode. Was passiert, wenn Sie während der Simulation wiederholt den Button betätigen? Was passiert, wenn die Krankheit aggressiver ist und bereits bei einer Energie von weniger als 40 zum Tode führt?
- b) Integrieren Sie einen Slider, mit dessen Hilfe Sie die Energie, bei der die Krankheit zum Tode führt, einstellen können. Eventuell sollten Sie die Laufzeitbeschränkung des Modells auf 500 Ticks aufheben.
- c) Führen Sie einen weiteren Button **grow** ein, der die Regenzeit simuliert, indem 50% der abgegrasteten Patches spontan nachwachsen. Welche Auswirkung hat das spontane Nahrungsangebot auf die Population? Wenn Wachstum und Krankheit zeitlich zusammenliegen, welche Effekte lassen sich beobachten?

---

<sup>1</sup>Als Quellen können Sie auch Internetseiten angeben.