

# Einführung in die Informatik

## Übung zu Modellierung und Simulation

8. Dezember, 2011

Diese Übung muss bis Mittwoch 21. Dezember um 14 Uhr auf Ilias abgegeben werden. Beachten Sie jeweils genau die Angaben zum Material, das abgegeben werden soll. Die Übung kann in Zweiergruppen bearbeitet werden. Vergessen Sie nicht, Ihre Namen und Matrikelnummern in der Abgabe zu vermerken.

### Praktische Aufgaben

In dieser Übung arbeiten Sie mit Greenfoot, einem Werkzeug, um spielerisch das Programmieren mit Java zu erlernen. Wir verwenden Greenfoot in dieser Übung, um mit Simulationen zu experimentieren. Sie können Greenfoot kostenlos von <http://www.greenfoot.org> beziehen. Beachten Sie, dass als Voraussetzung auch ein Java JDK (Java Development Kit) installiert sein muss. Ein Link dazu finden Sie auch auf der Greenfoot Webseite.

Greenfoot bietet unter <http://www.greenfoot.org/doc> umfangreiche Dokumentation an. Um sich mit den Grundkonzepten bekannt zu machen, empfiehlt es sich, zuallererst ein Tutorial auf dieser Webseite zu bearbeiten. Grundsätzlich verwendet Greenfoot Java als Programmiersprache.

Für die folgenden Aufgaben stellen wir auf Ilias jeweils ein Greenfoot Projekt als Grundgerüst für die Lösung zur Verfügung. Wir empfehlen, dass Sie sich mit Hilfe dieser Beispiele einarbeiten. Zögern Sie nicht, Fragen und Probleme auf dem Ilias Forum und in den Poolstunden zu diskutieren, insbesondere wenn Sie Fragen zur Java Programmierung haben.

#### 1. Zwei-dimensionale Wärmeleitung

In dieser Aufgabe geht es darum, die zwei-dimensionale Wärmegleichung zu simulieren. Verwenden Sie als Ausgangslage das Greenfoot Projekt im Verzeichnis *Heat*. Der vorgegebene Programmcode implementiert eine vereinfachte Simulation, welche durch Mausklick auf den *Run* Button unten im Greenfoot Fenster gestartet werden kann.

Die Simulation basiert auf einem einzigen Objekt *heater*. Sie können den Java Code dieses Objekts ändern, indem Sie auf der rechten Seite des Greenfoot Fensters auf den Kasten *Heater* doppelklicken. Beachten Sie, dass Sie nach Veränderungen des Codes jeweils neu compilieren müssen, um die Änderungen zu aktivieren. Klicken Sie dazu auf den *Compile* Button unten rechts im Greenfoot Fenster.

Studieren Sie zunächst den vorgegebenen Programmcode. Das *Heater* Objekt speichert ein zwei-dimensionales Feld, welches die Wärmeverteilung auf einem Gitter darstellt. In je-

dem Simulationsschritt wird von Greenfoot automatisch die Methode *act* aufgerufen. Die Veränderung der Temperaturen des Feldes soll dann in der Methode *diffuseTemperature* berechnet werden, die von *act* aufgerufen wird. Schliesslich erfolgt die Darstellung des Gitters in der Methode *displayTemperature*, die am Schluss jedes Schritts abgearbeitet wird.

## Aufgaben

Modifizieren Sie die Methode *diffuseTemperature* so, dass die Entwicklung der Wärmeverteilung mit dem finiten Differenzenverfahren wie in der Vorlesung berechnet wird.

Beachten Sie, dass die Formel zur Berechnung jedes Wertes immer auf alle vier Nachbarn zugreift. Das bedeutet, dass in einer naiven Implementation an den Rändern auf nicht existierende Daten zugegriffen wird. Um dieses Problem zu lösen, müssen Sie zusätzliche *Randbedingungen* formulieren. Implementieren Sie die folgenden zwei Varianten:

- *Periodische Randbedingungen*: Man stellt sich vor, das zwei-dimensionale Gitter wiederhole sich periodisch. Anschaulich heisst das, dass das nächste Element links vom linken Rand dem Element links vom rechten Rand entspricht. Angenommen die Indizes laufen von 0 bis  $n - 1$ , dann wird also der Index  $-1$  auf den Index  $n - 1$  abgebildet. Umgekehrt entspricht der Index  $n$ , der ausserhalb des Feldes liegt, dem Index 0. Beachten Sie, dass die periodische Berechnung der Indizes mit Hilfe von Modulo-Operationen ohne Fallunterscheidungen erreicht werden kann.
- *Dirichlet Randbedingungen*: Man nimmt an, die Nachbarelemente ausserhalb des Bereichs hätten einen festen Wert. Wählen Sie den Wert null für Ihre Experimente.

Experimentieren Sie mit beiden Varianten, die Randbedingungen zu implementieren. Welche Unterschiede beobachten Sie?

Untersuchen Sie weiter den Einfluss des Parameters  $r = k/h^2$ , der in der Diskretisierung mit finiten Differenzen vorkommt. Beschreiben Sie Ihre Beobachtungen.

## Abgabe

Abzugeben sind je ein Greenfoot Scenario für die Implementation der zwei verschiedenen Randbedingungen. Ausserdem sollen die Beobachtungen in Worten in einer Textdatei aufgeschrieben werden.

## 2. Schafherde

In dieser Aufgabe simulieren Sie eine weidende Schafherde. Verwenden Sie als Ausgangslage das Greenfoot Projekt im Verzeichnis *Sheep*. Die zwei wichtigsten Objekte der vorgegebenen Implementation sind die Schafe *Sheep* und Grasflächen *Grass*. Studieren Sie das Verhalten dieser Objekte, indem Sie auf die entsprechenden Buttons rechts im Greenfoot Fenster doppelklicken.

## Aufgaben

Die vorgegebene Simulation ist “unrealistisch”, weil die Schafe nur Grass fressen, sich aber weder vermehren noch sterben. Erweitern Sie das Verhalten der Schafe folgendermassen:

- Jedes Schaf speichert eine Variable *full*, die anzeigt wie stark es mit Gras vollgefressen ist. Die Variable soll bei der Erzeugung eines Schafs auf einen bestimmten Wert gesetzt werden. Der Wert wird in jedem Schritt erhöht, wenn das Schaf Gras findet. Er wird in jedem Schritt um einen gewissen Futterverbrauch verkleinert, wenn kein Futter gefunden wird.
- Wenn ein Schaf genügend Nahrungsreserven hat, das heisst *full* überschreitet einen gewissen Schwellwert, dann pflanzt es sich fort. Sie können dazu mit dem Java Schlüsselwort *new* eine neue Instanz eines Schafes erzeugen. Gleichzeitig soll der Wert *full* des “Mutterschafs” wieder unter den Schwellwert reduziert werden.
- Wenn ein Schaf über längere Zeit kein Futter findet, dann wird sein *full* Wert unter null sinken. In diesem Fall stirbt das Schaf. Sie löschen die Instanz eines Schafs indem Sie es mittels des Befehls *getWorld().removeObject(this)* aus der Simulation entfernen.

Die folgende Liste enthält eine Übersicht über die Simulationsparameter, die das Verhalten der Schafe bestimmen:

- Der Zuwachs von *full*, wenn ein Stück Gras gefunden wird.
- Die Reduktion von *full*, wenn kein Gras gefunden wird.
- Der Schwellwert von *full*, der eine Geburt auslöst.
- Der Wert von *full* der Mutter nach der Geburt.
- Der Wert von *full* des Neugeborenen bei der Geburt.

Beachten Sie, dass der vorgegebene Code die Anzahl Schafe und Grassflächen in jedem Schritt in eine Datei “sheep\_data.txt” speichert. Beachten Sie, dass die Datei jeweils erst auf Disk geschrieben wird, wenn Sie die Simulation stoppen indem Sie auf den *Pause* Button unten im Greenfoot Fenster klicken.

Experimentieren Sie mit verschiedenen Werten für die Simulationsparameter. Analysieren Sie jeweils die gesammelten Daten (Anzahl Schafe und Grasflächen zu jedem Zeitschritt), die in der Datei *sheep\_data.txt* gespeichert werden, indem Sie die Daten als Kurven plotten. Dazu können Sie z.B. Microsoft Excel oder ein Programm Ihrer Wahl verwenden. Beschreiben Sie Ihre Beobachtungen in ein paar Worten.

## Abgabe

Abzugeben ist das Greenfoot Scenario für die erweiterte Simulation der Schafherde. Die analysierten Daten in Form von Kurven müssen als Dateien im PDF Format abgegeben werden. Für alle Kurven soll ersichtlich sein, mit welchen Simulationsparametern sie erzeugt wurden. Ausserdem sollen die Beobachtungen in Worten in einer Textdatei aufgeschrieben werden.

### **Bonusaufgabe**

Erweitern Sie die Schafsimulation um zusätzliche Effekte Ihrer Wahl. Sie könnten weitere Tierarten, wie zum Beispiel Wölfe, einführen, und Sie könnten weitere Verhaltensweisen implementieren. Sie könnten auch Benutzerinteraktion ermöglichen, oder die Simulation optisch verbessern, indem Sie eigene Sprites erstellen. Geben Sie einen kurzen Text ab, der Ihre Erweiterungen beschreibt, sowie das entsprechende Greenfoot Projekt. Wir werden die besten Abgaben in der letzten Vorlesungsstunde vorstellen und mit einem kleinen Preis belohnen!