

# Satelliten-Netz

## Lösung des 1. Studierendenwettbewerb der GI

Yves Radunz      Volker Grabsch

20. Dezember 2005

### Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
1.1	Allgemeines zum Programm . . . . .	2
1.2	Verwendetes Modell . . . . .	2
<b>2</b>	<b>Klassen</b>	<b>3</b>
2.1	Vektor . . . . .	3
2.2	Satellit . . . . .	3
2.3	SatNet . . . . .	4
2.4	Graph . . . . .	5
2.5	MainRenderer . . . . .	5
<b>3</b>	<b>Probleme/Grenzen</b>	<b>5</b>
<b>4</b>	<b>Erweiterungsmöglichkeiten</b>	<b>6</b>

# 1 Einleitung

## 1.1 Allgemeines zum Programm

Unser Programm ist freie Software<sup>1</sup>. Projekt-Homepage:

<https://dev.njh6.de/wiki/index.php/SatNet-VY>

Das Programm, bzw. die zu diesem Programm gehörigen Klassen, wurden in Java implementiert. Zur visuellen Darstellung der Satelliten, ihres Kommunikationsnetzes und des Planeten wird OpenGL verwendet. Als Schnittstelle zwischen Java und OpenGL dient uns JOGL<sup>2</sup>.

Programm-Aufruf:

```
java -classpath SatNet-VY.zip Main
```

Das Programm gibt beim Start eine kurze Erklärung zur Bedienung auf der Standardausgabe aus:

F2, S	anhalten / weiter
F3, D	Erde durchsichtig (an / aus)
F4, *	neuen Satelliten auf einer zufällig einfügen
F5, +	neuen Satelliten optimal einfügen
F6, -	letzten Satelliten entfernen
Esc, Q	beenden
klicken, ziehen	Szene drehen
Doppelklick	Erde durchsichtig [an/aus]

Während des Ablaufs des Programms werden die gesuchten Parameter, z.B. die minimale, durchschnittliche, maximale und aktuelle Netzlänge, ebenfalls auf der Standardausgabe angezeigt.

## 1.2 Verwendetes Modell

Die wichtigsten Eigenschaften eines Satelliten sind seine Flugbahn und Fluggeschwindigkeit. Die Flugbahn wird durch die Aufgabenstellung dahingehend eingeschränkt, dass sie kreisförmig ist und alle Satelliten die gleiche Flughöhe haben. Die Flughöhe der Satelliten wird beim Erstellen des Satellitennetzes einmal festgelegt und kann danach nicht mehr verändert werden. Daher kann die Flughöhe der Satelliten beim Hinzufügen eines neuen Satelliten automatisch festgelegt werden. Außerdem wird hierdurch verhindert, dass durch Bedienungsfehler des Programms der Fall eintritt, dass die Satelliten eines Netzes in unterschiedlichen Flughöhen fliegen. Auf Grund der physikalischen Gesetze bezüglich Schwerkraft und Fliehkraft wird die Geschwindigkeit der Satelliten durch ihre Flughöhe und die Masse der Erde festgelegt. Für die von uns verwendete Darstellung der Satelliten wird jedoch die Geschwindigkeit nicht direkt gespeichert, sondern lediglich die Umlaufzeit der Satelliten, welche ebenfalls nur von der Planetenmasse und Flughöhe abhängig ist. Der Radius des Planeten wird nur benötigt, um beim Erstellen des Kommunikationsnetzes zu verhindern, dass die

<sup>1</sup><http://www.gnu.org/philosophy/free-sw.de.html>

<sup>2</sup><https://jogl.dev.java.net/>

Satelliten versuchen, durch den Planeten hindurch miteinander zu kommunizieren. Angesichts der Tatsache, dass die Satelliten im Vergleich zum Planeten viel zu klein sind, werden Satelliten ein wenig größer als in der Realität (mit einem Radius von 200km) dargestellt. Den Abstand, ab dem Kollisionsgefahr besteht, nennen wir „Alarmschranke“. In dem Programm beträgt die Alarmschranke das 10fache des Satellitenradius’.

## 2 Klassen

### 2.1 Vektor

Diese Klasse stellt alle Methoden zur Verfügung, um die Berechnung von Position, Abstand und Bewegung der Satelliten zu vereinfachen. Unter anderem werden hier Methoden zur Berechnung des Skalarprodukts (Winkelberechnung), Betrag eines Vektors (nötig zur Abstandsberechnung), Vektorprodukt (Berechnung der Flugbahn) und Skalierung eines Vektors (z.B. Flughöhe) bereitgestellt.

### 2.2 Satellit

Die Umlaufbahn der Satelliten wird beim Erstellen des Satelliten durch zwei Punkte festgelegt, die er (in dieser Reihenfolge) überfliegen soll. Satelliteneigenschaften wie die Umlaufzeit müssen nur einmal berechnet werden und bleiben dann für den Rest des Programmablaufes konstant. Daher können sie als feste Werte für jeden Satelliten gespeichert werden. Ebenso ist die Alarmschranke eines Netzes für jeden Satelliten gleich. Weil theoretisch auch mehrere Satellitennetze gleichzeitig (in unterschiedlichen Flughöhen) existieren können, muss die Flughöhe, Umlaufzeit etc. für jedes Netz separat gespeichert werden. Die Satelliten sollen jedoch auch in der Lage sein, ihre Position zu einem bestimmten Zeitpunkt zu berechnen, ohne immer wieder auf das Netz, zu dem sie gehören, zugreifen zu müssen. Weil hierdurch für jeden Satelliten bestimmte Eigenschaften (z.B. die Flugbahn, Flughöhe, Umlaufzeit) gespeichert werden müssen, empfiehlt es sich, diese als Objekte einer Klasse (Satellit) darzustellen.

Da sich die Satelliten in einem dreidimensionalen Raum befinden, müssen viele Berechnungen von Aufenthalts-Orten (der Satelliten) und deren Abstände durchgeführt werden, die sich durch Vektoroperationen relativ einfach darstellen lassen. Um die Position eines Satelliten zu einem Zeitpunkt  $t$  bestimmen zu können, werden zwei zueinander rechtwinklige Vektoren als Eigenschaften dieses Satelliten gespeichert, welche die Ebene, in der die Umlaufbahn des Satelliten liegt, aufspannen. Der erste Vektor gibt den Punkt (auf dem Planeten) an, der von dem Satelliten zum Zeitpunkt  $t=0$  überflogen wird. Der zweite Vektor gibt den Punkt auf dem Planeten an, der nach dem ersten Viertel der Umlaufzeit vom Satelliten überflogen wird. Jeder Satellit kann nun durch die Methode

```
Vektor getPosition(double zeit)
```

einen Vektor ausgeben, der angibt, wo er sich zum Zeitpunkt `zeit` befindet. Damit ist es auch möglich, den Abstand zu einem anderen Satelliten

```
double dist(Satellit sat, double zeit)
```

zu einem bestimmten Zeitpunkt zu bestimmen und dessen Minimum mit der Alarmschranke

```
boolean kollgef(Satellit sat)
```

zu vergleichen.

### 2.3 SatNet

Ein komplettes Netz von Satelliten wird durch die Klasse SatNet definiert. Jedes Objekt dieser Klasse verwendet eine Auflistung der in dem Netz enthaltenen Satelliten, um die Eigenschaften (Kommunikationsnetz usw.) des Satellitennetzes zu bestimmen. Das kürzeste Kommunikationsnetz wird in zwei Schritten gesucht:

Zuerst wird ein Graph erstellt, der alle Verbindungen zwischen zwei Satelliten enthält, die nicht durch den Planeten führen. Hierfür muss lediglich bestimmt werden, ob die Gerade, welche die Verbindung zweier Satelliten als Strecke enthält, einen Abstand vom Planetenmittelpunkt hat, der kleiner oder gleich dem Radius des Planeten ist. Der Einfachheit halber wird keine (dünne) Schicht als Atmosphäre beachtet, welche unter Umständen eine direkte Kommunikation zwischen den Satelliten erschweren könnte.

Im zweiten Schritt wird ein Kommunikationsnetz aufgebaut, welches nur die Verbindungen aus diesem Graphen enthält.

Die Klasse SatNet dient auch zur Abfrage der minimalen, durchschnittlichen und maximalen Netzlänge. Diese werden schrittweise angenähert.

Außerdem stellt diese Klasse die Methoden bereit, mit denen Satelliten hinzugefügt und entfernt werden können. Die Methode `newsat` dient dabei entweder zum Hinzufügen eines Satelliten in einer bestimmten Umlaufbahn, oder zum Hinzufügen eines Satelliten, dessen Bahn so gewählt ist, dass die Alarmschranke möglichst hoch gewählt werden kann. Der erste Satellit wird auf eine Startbahn gesetzt, die bei der Ausgangslage des Planeten eine nicht allzu langweilige Bewegung des Satelliten in der Visualisierung verspricht. Der zweite Satellit wird auf die gleiche Umlaufbahn gesetzt, wobei er jedoch so positioniert wird, dass er sich immer exakt gegenüber dem ersten Satelliten befindet. Die Suche nach der Bahn der darauf folgenden Satelliten erfolgt in zwei Schritten, wobei zunächst der Startpunkt und dann der zweite Punkt, der zur Definition der Richtung benötigt wird, bestimmt wird.

Der Startpunkt wird so gewählt, dass er sich möglichst weit von allen anderen Satelliten-Startpunkten entfernt befindet. Der zweite Punkt wird so gewählt, dass die Bahn des Satelliten so liegt, dass der Satellit während seines Fluges möglichst weit von den anderen Satelliten entfernt ist. In beiden Schritten werden lediglich einige Möglichkeiten durchprobiert. Je mehr Satelliten sich bereits im Netz befinden, umso mehr mögliche Werte für den Startpunkt und die Richtung werden überprüft.

Um die Berechnung im Schritt 2 zu vereinfachen, wird ein Satellit für kurze Zeit hinzugefügt und danach dessen Methoden verwendet, um den minimalen Abstand zu den anderen Satelliten zu bestimmen. Danach wird er wieder gelöscht, um die nächste mögliche Bahn zu testen.

Zu beachten ist außerdem, dass nur einige Satellitenpositionen getestet werden. Es wird im Allgemeinen nicht die absolut beste Umlaufbahn gefunden.

Die minimale, durchschnittliche und maximale Netzlänge wird nur mit einer bestimmten Qualität berechnet. Diese Qualität wird, wenn sie nicht angegeben

wird, auf

$$\max\left(\frac{10000}{n}, 100\right)$$

gesetzt. ( $n$  bezeichnet hierbei die Anzahl der Satelliten.) Diese Längen werden zum ersten Mal (und alle gleichzeitig) berechnet, wenn eine dieser Längen abgefragt wird. Danach werden sie nur neu berechnet, wenn sich zwischenzeitlich die Anzahl der Satelliten, oder die Qualität der Längenmessung geändert hat. Die Qualität gibt an, wie viele Messungen durchgeführt werden. Die Messungen der Längen erfolgen für bestimmte Zeitpunkte, die gleichmäßig über die gesamte Umlaufperiode verteilt sind. Die Anzahl dieser Zeitpunkte ist gleich der Qualität der Messung.

## 2.4 Graph

Diese Klasse stellt das Kommunikationsnetz eines Satellitennetzes dar. Sie dient zur Verwaltung des Netzes, um die Satellitenverbindungen visuell darzustellen, und zum Erstellen des kürzesten Kommunikationsnetzes aus einer gegebenen Menge von Verbindungen, die nicht durch den Planeten führen. Die Methode

```
void optimize(int anzahl, Vector sat, double zeit)
```

übernimmt dies unter Verwendung einer abgewandelten Form des Kruskal-Algorithmus<sup>7</sup> zur Bestimmung eines minimal spannenden Teilbaumes. Am Ende erhält man ein Kommunikationsnetz, das genau  $n-1$  Verbindungen bei  $n$  Satelliten enthält.

## 2.5 MainRenderer

Diese Klasse dient zur Darstellung des Planeten, der Satelliten und des Kommunikationsnetzes. Die Satelliten sind hell-rot, der Planet hell-blau und das Netz ist weiß. Zur Orientierung wird die Nord-Süd-Achse des Planeten hell-grün eingezeichnet. Wenn die Gefahr besteht, dass zwei (oder mehrere) Satelliten zu einem bestimmten Zeitpunkt kollidieren, d.h. wenn ihr Abstand geringer als die Alarmschranke ist, dann werden diese Satelliten zu diesem Zeitpunkt knall-rot dargestellt.

Des Weiteren übernimmt diese Klasse auch die Ausgabe der minimalen, durchschnittlichen und maximalen Netzlänge, sowie die Meldung eventueller Fehler bzw. Warnungen („Kein Netz“ und „Kollisionsgefahr“).

## 3 Probleme/Grenzen

Beim Erstellen eines neuen Satelliten müssen zwei Punkte angegeben werden, die vom Satelliten überflogen werden sollen. Dies hat den Vorteil, dass nicht versehentlich eine falsche Flughöhe eingegeben werden kann, wie es zum Beispiel bei einer expliziten Angabe der kartesischen Koordinaten möglich wäre. Wenn jedoch zweimal der selbe Punkt, oder zwei diametral gegenüberliegende Punkte, angegeben werden, kann diesen Werten keine eindeutige Flugbahn zugeordnet werden. Das Ergebnis der Berechnungen, die in dem Programm verwendet werden, wäre eine nicht-kreisförmige Bahn, die ein Hin- und Herpendeln auf einer Geraden durch den Mittelpunkt des Planeten darstellt.

Eine Lösung wäre die Angabe eines Punktes und einer Richtung, in die der Satellit fliegt. Dies ist jedoch schwieriger, weil hier eine eindeutige allgemeine Festlegung einer Richtung nötig ist. Eine Lösung wie z.B.  $0^\circ$  entspricht „nach Norden“ ist nicht immer eindeutig, da an den Polen des Planeten keine eindeutige Nord-Richtung festgelegt ist.

Ein anderes Problem ist die Tatsache, dass nicht immer ein vollständiges Kommunikationsnetz existieren muss. Zum Beispiel können zwei Satelliten sich immer auf zwei unterschiedlichen Seiten des Planeten befinden, wodurch eine direkte Kommunikation zwischen diesen Satelliten nie möglich ist. Analog kann auch bei mehreren Satelliten der Fall eintreten, dass der Graph in mehrere Zusammenhangskomponenten zerfällt.

In diesem Fall liefert der hier verwendete Algorithmus nur eine dieser Zusammenhangskomponenten, also ein Netz, das weniger als  $n-1$  Verbindungen bei  $n$  Satelliten enthält. Das Programm reagiert hierauf, als ob gar kein Netz existieren würde: Es gibt die Meldung aus, dass kein Netz existiert. Demnach wird auch die Gesamtlänge dieses Teilnetzes als 0 betrachtet. Die Angaben der minimalen, durchschnittlichen und maximalen Netzlänge werden hierdurch ebenfalls beeinflusst, wenn zeitweilig kein vollständiges Kommunikationsnetz existiert.

## 4 Erweiterungsmöglichkeiten

Mögliche Erweiterungen (die teilweise schon von den hier verwendeten Klassen unterstützt werden) sind zum Beispiel:

1. Man kann Netze für andere Planeten als die Erde simulieren, oder auch zum Beispiel ein Satellitennetz für die Sonne. Hierfür müssen lediglich andere Werte für Masse und Radius festgelegt werden.
2. Eine Atmosphäre kann die Kommunikation der Satelliten auf direktem Wege beeinträchtigen. Hierfür kann man zusätzlich zu dem Planetenradius noch die Dicke der Atmosphäre betrachten, um Kommunikationsverbindungen zu behandeln, die vielleicht nur zeitweise und in Abhängigkeit vom Wetter etc. möglich sind.
3. Manchmal können die Satelliten so ungünstig stehen, dass ein Satellit beim Aufbau des kürzesten Kommunikationsnetzes als zentraler Knoten verwendet wird. Dieser könnte dann unter Umständen überlastet werden. Man kann also eine weitere Einschränkung, wie zum Beispiel nur eine maximale Anzahl an Kommunikationsverbindungen mit einem Satelliten, festlegen.
4. Wenn ein Satellit ausfällt, oder überlastet wird, ist es gut, wenn eine Ausweichverbindung über andere Satelliten zur Verfügung steht. Das Kommunikationsnetz könnte also auch dahingehend verändert werden, dass (egal, welcher Satellit ausfällt) immer eine zweite Verbindung zu jedem anderen Satelliten zur Verfügung steht.
5. Physikalisch sind auch nicht-kreisförmige Bahnen möglich, sowie Netze, die Satelliten in unterschiedlichen Flughöhen enthalten. Letzte müssen jedoch nicht unbedingt periodisch sein, was die Bestimmung der Kollisionsgefahr erschwert. Denkbar wäre jedoch eine Überprüfung, ob die Satelliten innerhalb einer gewissen Zeitspanne kollidieren können.