Monika Wojtowiec
Michael Gebhard
STephan Kambor

Projektdokumentation:

Der Turm von Hanoi

[als Java Applet]

Index Seite

1. Die Aufgabenstellung

Unsere Aufgabe bestand darin, ein Java – Applet zu entwickeln was die Lösung des Turms von Hanoi grafisch darstellt. Es sollte eine beliebige Anzahl von Scheiben einzugeben sein. Wir haben im Applet die Auswahl auf 5 Scheiben begrenzt, da sonst der Platz im Applet nicht ausreicht. Weiterhin ist zu beachten, dass die Zeit zum Anzeigen von Lösungen mit ansteigen der Scheibenanzahl extrem groß wird, dies wird im Kapitel *Die Lösung* noch genauer beschrieben. Unser Programm sollte über Eingabemöglichkeiten wie Tasten zum Starten und Stoppen der Animation, Tasten um einen Schritt vorwärts oder rückwärts zu gehen und eine letzte Taste um das Applet auf die Ausgangsposition zurückzusetzen verfügen.

2. Das Spiel

Hier gehen wir auf die Geschichte und das Spielprinzip ein.

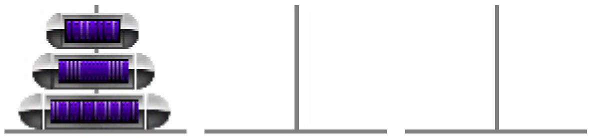
Das Spiel basiert auf einer Legende, welche vom französischem Mathematiker Eduard Lucas im Jahre 1883 erzählt wurde.

In einem Tempel in der indischen Stadt Benares lagen 64 kostbare Scheiben aus Diamant zu einem Turm aufgeschichtet. Jede Scheibe ist ein wenig kleiner als die Scheibe auf der sie ruht. Ein Priesterorden hat nun die Aufgabe erhalten den Turm unter Beachtung heiliger Regeln an eine andere Stelle im Tempel zu bewegen. Die Scheiben sind so kostbar, dass sie nur auf einem von drei Plätzen im Tempel liegen dürfen, dem Platz wo der Turm zu Beginn stand, dem wo er aufgebaut werden soll und einem weiteren Platz, der als Zwischenlager dient. Die Scheiben sind schwer und zerbrechlich, daher darf immer nur eine der Scheiben bewegt werden, niemals mehrere zur gleichen Zeit. Die letzte Regel besagt, dass zu keiner Zeit eine Scheibe auf einer kleineren Scheibe liegen darf. Wenn der Turm an einer Stelle abgebaut und an anderer Stelle wieder ganz aufgebaut wurde, wird der Tempel und mit ihm die ganze Welt zu Staub zerfallen.

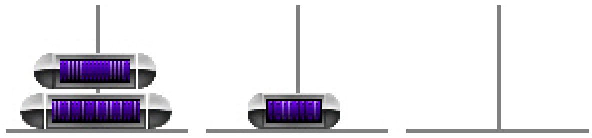
Ob es nun möglich ist diese Scheiben umzusetzen, und wenn es möglich ist, wie lange das Ganze dauert, wird in den folgenden Abschnitten geklärt.

3. Die Lösung

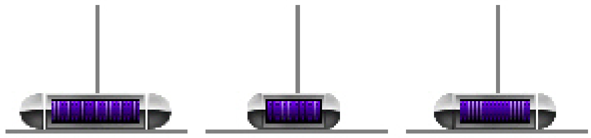
Es gilt also zu beweisen, dass es überhaupt möglich ist einen Turm von beliebig vielen Steinen nach den oben genannten Prinzipien umzusetzten. Als erstes schauen wir uns an, wie die das Spiel mit 3 Scheiben gelöst werden kann.



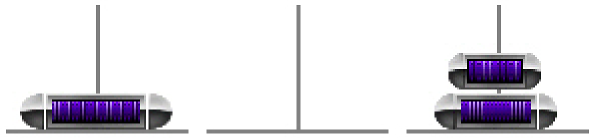
#0)
Als Ausgangssituation haben wir also 3 Scheiben die vom ersten Platz [links] auf den mittleren Platz [mitte] umgesetzt werden sollen. Der dritte und rechte Platz dient als Zwischenlager.



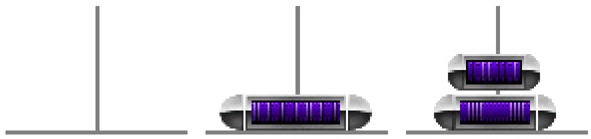
#1)
Die erste und kleinste Scheibe kommt auf Platz 2.



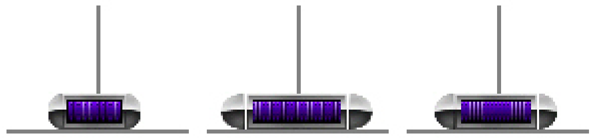
#2)
Die zweite (mittlere) Scheibe wird auf Platz 3 gesteckt.



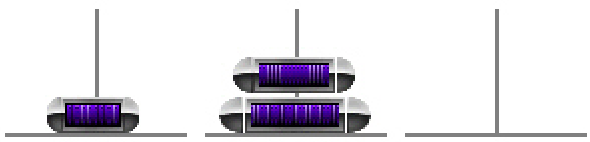
#3)
Nun wird die kleine Scheibe ebenfalls auf Platz 3 gesteckt, was erlaubt ist weil sie kleiner ist. (andersherum geht es nicht!)



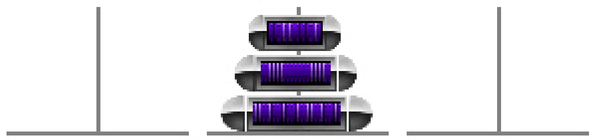
#4)
Nun kann die dritte (große) Scheibe auf den Platz 2 gesteckt werden, was für diese Scheibe nun auch schon die Endposition ist.



#5)
Die erste Scheibe kann jetzt auf Platz 1 gelegt werden.



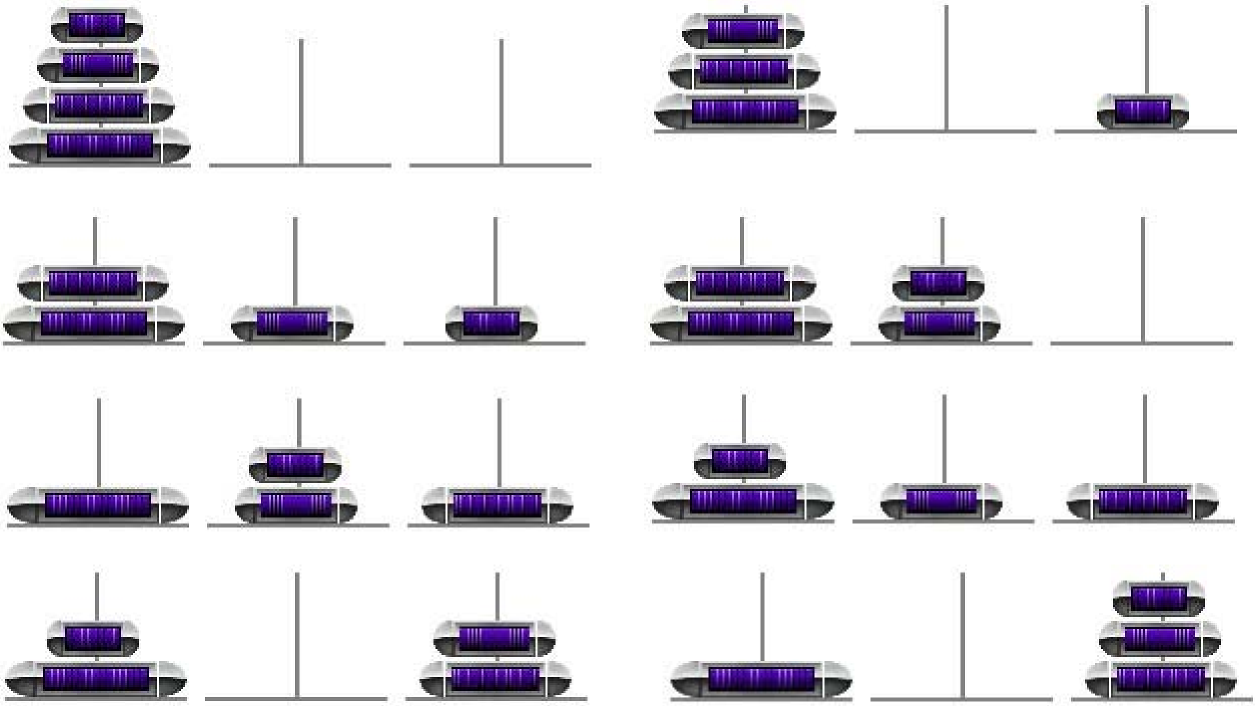
#6)
Nun die zweite mittelgroße Scheibe auf Platz 2 ...



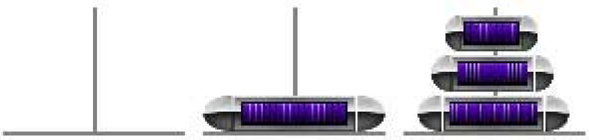
#7)
...und die kleine oben drauf. Fertig!

Es ist also möglich unter Beachtung der Regeln den dreischeibigen Turm von Platz 1 nach Platz 2 zu versetzen. Bei genauerem Hinsehen wäre es auch möglich gewesen, den Turm von Platz 1 auf Platz 3 umzubauen, oder von Platz 2 auf Platz 3 usw. Prinzipiell kann man sagen, der Turm ist von irgendeinem der drei Plätze auf irgendeinen der beiden anderen unter Beachtung der Regeln umbaubar. Ein schöner Satz wie ich finde.

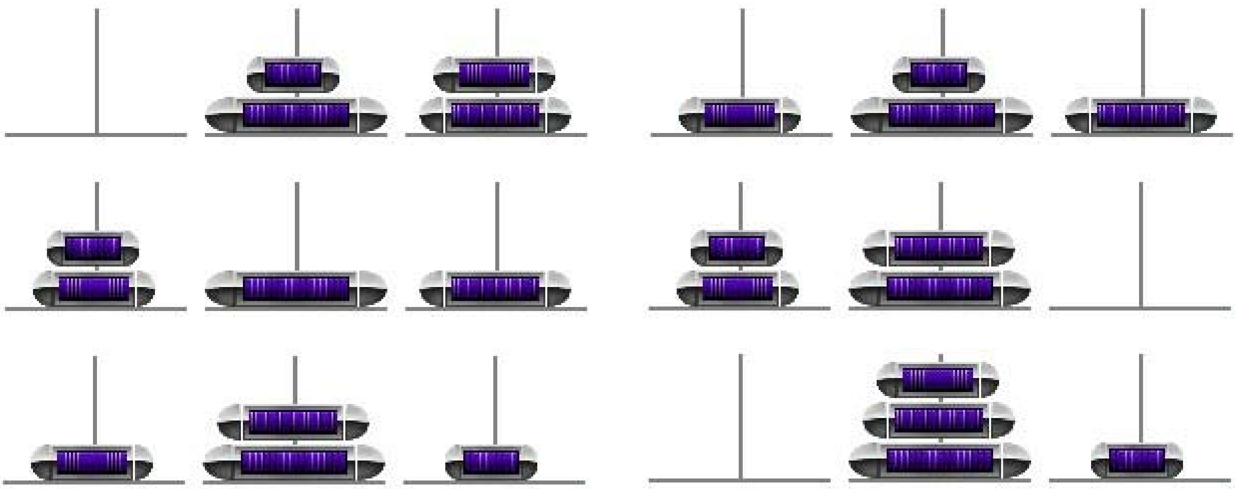
Schauen wir uns nun die Angelegenheit mit vier Scheiben an und betrachten den Turm als Turm mit drei Scheiben bei denen eine vierte untergeschoben wurde.

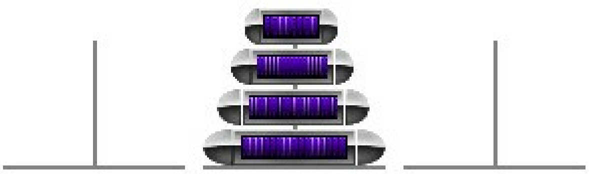


Was ist bis hierhin passiert? Wir haben genau das gemacht, was wir oben herausgefunden haben. Der obere Dreierturm wurde in sieben Schritten von Platz 1 auf Platz 3 umgesetzt.



Nun wurde die vierte Scheibe auf Platz 2 gelegt und wir müssen jetzt wieder nur das machen, was wir oben gesagt haben, den Dreierturm in weiteren sieben Schritten von Platz 3 auf Platz 2 umbauen und wir sind fertig.





Wir haben also in 7 + 1 + 7 = 15 Schritten einen Turm mit vier Scheiben umgesetzt. Auch hier können wir wieder sagen: Der Turm ist von irgendeinem der drei Plätze auf irgendeinen der beiden anderen unter der Beachtung der Regeln umbaubar.

Aufgrund der obigen Erkenntnisse können wir überlegen, wie ein Turm mit fünf Scheiben umgebaut werden kann. Der obere Viererturm wird in 15 Zügen von Platz 1 auf Platz 3 gesetzt, die fünfte Scheibe wird auf Platz 2 gesteckt und dann kann der Turm auf Platz 3 in weiteren 15 Zügen auf Platz 2 umgesetzt werden.

Es ist also möglich Türme mit n Scheiben von Platz 1 nach Platz 2 umzusetzen. Man kann also für einen Turm aus 1+n Scheiben sagen: Baue den Turm aus n Scheiben von Platz 1 nach Platz 3 um und lege die übriggebliebene Scheibe auf Platz 2, dann setze den Turm von Platz 3 auf Platz 2 um.

Es funktioniert also immer nach dem gleichen Prinzip. Wenn sich ein Dreierturm umsetzen lässt, lässt sich auch ein Viererturm umsetzen. Wenn sich ein Viererturm umsetzen lässt, lässt sich auch ein Fünferturm umsetzen und so weiter. Dieses Prinzip nennt man mathematisch ausgedückt: <u>Beweis durch vollständige Induktion</u>.

Anhand der obigen Schlussfolgerungen ist es nun auch möglich zu errechnen, wieviele Züge zum lösen eines bestimmten Turmes nötig sind.

Für 3 Scheiben 7 Züge. [$2^3 - 1$]
Für 4 Scheiben 7+1+7=15 Züge. [$2^4 - 1$]
Für 5 Scheiben 15+1+15=31 Züge. [$2^5 - 1$]

Daraus ergibt sich eine erstaunlich simple Formel:

$$Züge\ zur\ Lösung = 2^{Scheibenanzahl} - 1$$

Wir können nun einmal ausrechnen, wie viele Züge z.B. bei Scheibenanzahlen wie 10, 20, 30, 40, 50 oder 64 benötigt werden.

$2^{10} - 1 = 1023$
$2^{20} - 1 = 1.048.575$
$2^{30} - 1 = 1.073.741.823$
$2^{40} - 1 = 1.099.511.627.775$ -> hier sind es schon über eine Trillion Züge
$2^{50} - 1 = 1.125.899.906.842.623$ -> über Tausend Trillionen Züge
$2^{64} - 1 = 18.446.744.073.709.551.615$
-> 18 Millionen 446744 Trillionen 73 Billionen 709 Millionen 551615 Züge

Man kann als sehr deutlich sehen, dass die Mönche, selbst wenn sie keinen Fehler machen, diesen Turm nicht innerhalb eines Menschenlebens umsetzen können. Beim Umsetzen von einer Scheibe pro Sekunde bräuchte man grob gerundet 233.976.966.942 Jahre um einen Turm mit 64 Scheiben umzusetzen.

Wie man aber diese Lösung in Java umsetzt, wird im nächsten Abschnitt erklärt.

4. Die Umsetzung

Im Pogramm selbst wird dieses System so eingesetzt, dass die Lösungsreihenfolge durch einen rekursiven Algorithmus bestimmt und Schritt für Schritt abgearbeitet wird. Diese Reihenfolge wird in einen String gespeichert der in der Methode *Loesung* erzeugt. Die Abarbeitung erfolgt dann Schrittweise in der Methode *demo* die die Position des betroffenen Balken anpasst und die Variablen aktualisiert. Bei [Go] und [Back] wird *demo* einmal aufgerufen einen Schritt zu gehen, bei [Animation] werden mit Hilfe eines Threads die Schritte automatisch ausgeführt, und mit [Stop] wird der Thread stillgelegt. Die Steuerung des Threads erfolgt dabei über die boolean Variable *lauf* die mit [Animation] auf **true** gesetzt wird und in [Stop] und [Reset] auf **false**. [Reset] setzt die Position aller Balken wieder auf 0 (=erster Stift) und trägt bei der Lagervariable von Stift 0 die Balken ein.

Die eigentliche Umsetzung des Algorithmus geschieht in wenigen Programmzeilen:

```
String Loesung(int anzahl,String quelle,String lager,String ziel)
{
  String loesung=new String();
  loesung = quelle+ziel;
  if (anzahl>1)
    loesung = Loesung(anzahl-1,quelle,ziel,lager)+loesung+Loesung(anzahl-1,lager,quelle,ziel);
  return loesung;
}
```

Alles weitere zur Umsetzung ist im Quellcode einzusehen.

5. Die Browserkompatibilität

Das Applet funktioniert nur mit Browsern, die eine Java 2 Unterstützung bieten. Je nachdem wie sie installiert und konfiguriert wurden ist diese Funktion installiert oder nicht bzw. ein oder ausgeschalten. Entsprechende Hinweise findet man in den Einstellungen bzw. den Browser-howtos. Auf Anhieb funtioniert es nur mit dem IE, weil dort defaultmäßig alles eingeschalten ist (was ich als bedenklich einstufe). Bei allen andern Browsern (mit * gekennzeichnet) musste die Funktion eingeschalten bzw. nachinstalliert werden. Bei den neueren Mozilla Versionen und deren Ablegern (Netscape, OpenX) musste es nur eingeschalten werden, bei den anderen half ein nachinstallieren des j2re (Java 2 Runtime Environment) Paketes was kostenlos und plattformübergreifend auf java.sun.com geladen werden kann. Der Netscape 3.x unterstützt Java 2 überhaupt nicht.

Das Applet wurde mit folgenden Browsern und Betriebssystemen auf Funktion geprüft:

Betriebssystem	Browser	Funktion
Linux Kernel 2.4.18	Konqueror 3.0.0	ja*
Linux Kernel 2.4.18	Mozilla 1.2.1	ja*
Windows 2000 SP3	OpenOffice 1.0.1 / StarOffice 6.x	ja*
Windows 2000 SP3	Mozilla 1.1a	ja*
Windows 2000 SP3	Mozilla 1.2.1	ja*
Windows 2000 SP3	Netscape Navigator 3.03	nein (keine Schnittstelle)
Windows 2000 SP3	Netscape Navigator 6.2	ja*
Windows 2000 SP3	Internet Explorer 5.00.3502.1000	ja
Windows 2000 SP3/XP SP2	Internet Explorer 6.0.2800.1106	ja