

Visual Basic Hamster

Harald R. Haberstroh

1. Einleitung

Die Idee zum Hamstermodell ist ganz einfach: Sie als Programmierer müssen einen virtuellen Hamster in einer virtuellen Landschaft steuern.

Das Hamster-Modell wurde in einer einfachen Version zu Beginn der 80er Jahre in der GMD (Gesellschaft für Mathematik und Datenverarbeitung) entwickelt. Zielsprache war damals die imperative Programmiersprache ELAN.

Vorlage für das Hamster-Modell war dabei "Karel der Roboter". Das imperative Hamster-Modell wurde von Dietrich Boles an die Programmiersprache Java angepaßt.

Mich hat diese Idee begeistert und ich habe daher das Hamster-Modell für *Visual Basic* neu geschrieben.

Den *Visual Basic*-Hamster gibt es mittlerweile in der dritten Version. Die Ziel- und Programmiersprache ist *Visual Basic*. Der Hamster ist gänzlich in die Programmierumgebung von *Visual Basic* integriert. Dazu gibt es ein *ActiveX*-Control, welches automatisch aufgerufen wird, wenn das Hamsterprogramm gestartet wird.

1.1. Landschaft

Die Hamster-Welt ist eine gekachelte Ebene. Die Anzahl der Kacheln (Felder) ist nicht explizit vorgegeben.

Auf einzelnen Feldern können Körner liegen. Diese werden durch ein eigenes Symbol dargestellt, wobei aus der Grafik nicht ersichtlich ist wieviele Körner auf dem Feld liegen.

Auf einzelnen Feldern können Körner liegen. Diese werden durch ein eigenes Symbol dargestellt, wobei aus der Grafik nicht ersichtlich ist wieviele Körner auf dem Feld liegen.

1.2. Hamster

In einer Landschaft existiert immer genau ein Hamster, der auf einem der Felder steht. Dieses Feld darf keine Mauer enthalten. Das Feld, auf dem sich der Hamster

befindet, darf jedoch Körner enthalten.

Der Hamster kann auch Körner im Maul haben. Es ist aus der Grafik nicht ersichtlich, ob der Hamster Körner im Maul hat bzw. ob auf dem Feld noch Körner sind.

Der Hamster kann in vier verschiedene Richtungen schauen (Nord, Ost, Süd oder West).

1.3. Hamsteraufgaben

Ihnen werden Aufgaben gestellt, die Sie durch die Steuerung des Hamsters durch eine Landschaft zu lösen haben. Diese Aufgaben werden *Hamsteraufgaben* genannt und die entsprechenden Lösungsprogramme *Hamsterprogramme*.

Die Aufgaben werden zunächst in Form einer Hamsterlandschaft (eine einfache Textdatei) und einem dazugehörigen Text gestellt. Später werden keine bestimmten Landschaften mehr vorgegeben.

Eine Hamsterwelt kann einfach mit einem Texteditor erstellt werden. Dabei dürfen nur folgende Zeichen verwendet werden:

- '*' eine Mauer
- ' ' (Leerzeichen) ein leeres Feld
- Ziffern ('1', '2', ... '9') an dieser Stelle liegen ebensoviele Körner
- 'N', 'O', 'S', 'W' die Position des Hamsters mit Blickrichtung Norden, Osten, Süden bzw. Westen. Es darf nur eines dieser Symbole in einer Hamsterwelt vorkommen.

Folgendes Bild zeigt ein Beispiel einer Hamsterwelt:

Abbildung 1. Hamsterwelt



Diese Welt wird durch folgende Welt erzeugt:

```

*****
*               *
*11W           *1*
*11**          *1*
*11            *1*
*****

```

Folgende Aufgabenstellung ist zu lösen: der Hamster muss alle Körner einsammeln und dann stehen bleiben.

2. Grundlagen der Hamstersprache

2.1. Lexikalik

Der Zeichenvorrat (die Lexikalik) entspricht der der Programmiersprache *Visual Basic*. Im Prinzip sind das alle Zeichen, die Sie auf der Tastatur finden.

2.2. Syntax und Semantik

Hier werden zunächst die Grundbefehle definiert. Die Syntax hält sich an die der Sprache *Visual Basic* wobei zunächst nur ein Teil der Sprache verwendet wird.

Ein Hamster versteht folgende Grundbefehle:

vor

Der Hamster bewegt sich um ein Feld nach vor. Ist das Feld vor dem Hamster durch eine Mauer blockiert, wird das Hamsterprogramm durch einen *Laufzeitfehler* unterbrochen.

linksUm

Der Hamster dreht sich um 90 Grad nach links. Soll sich der Hamster nach rechts drehen, so muss man dies aus mehreren linksUm zusammensetzen.

nimm

Der Hamster nimmt ein Korn in sein Maul auf. Befindet sich auf dem Feld kein Korn, so gibt das einen *Laufzeitfehler*.

gib

Der Hamster legt ein Korn auf das Feld, auf dem er sich gerade befindet. Hat er kein Korn mehr im Maul, so gibt es einen *Laufzeitfehler*.

Der Hamster versteht zusätzlich noch drei Testbefehle:

vornFrei

Der Hamster prüft, ob vorne frei ist, d.h. ob sich keine Mauer auf dem Feld vor dem Hamster befindet (statt vornFrei ist auch vorneFrei erlaubt).

maulLeer

Der Hamster prüft, ob sein Maul leer ist (statt maulLeer darf auch backenLeer verwendet werden).

kornDa

Der Hamster prüft, ob sich (mindestens) ein Korn auf dem Feld liegt, auf dem sich der Hamster befindet.

Jeder Befehl muss in eine eigene Zeile geschrieben werden. Alles was hinter dem Zeichen ' steht, gilt als Kommentar und wird vom Hamster ignoriert. Kommentar dient als Zusatzinformation für den Programmierer.

Ein Hamsterprogramm beginnt mit einem Kommentar, in dem festgehalten wird, was die Aufgabenstellung ist und von wem das Programm stammt:

```
' File: demo.bas
',
' Zweck: Demonstration des Programmkopfes
',
' Autor: Harald Haberstroh/hp
',
' Algorithmus:
',
' Status: <draft> testen freigegeben
',
' History:
' 2001-08-25, Ersterstellung
',
```

Das eigentliche Programm beginnt mit

```
Option Explicit
Sub main()
```

und endet mit ¹

```
End Sub
```

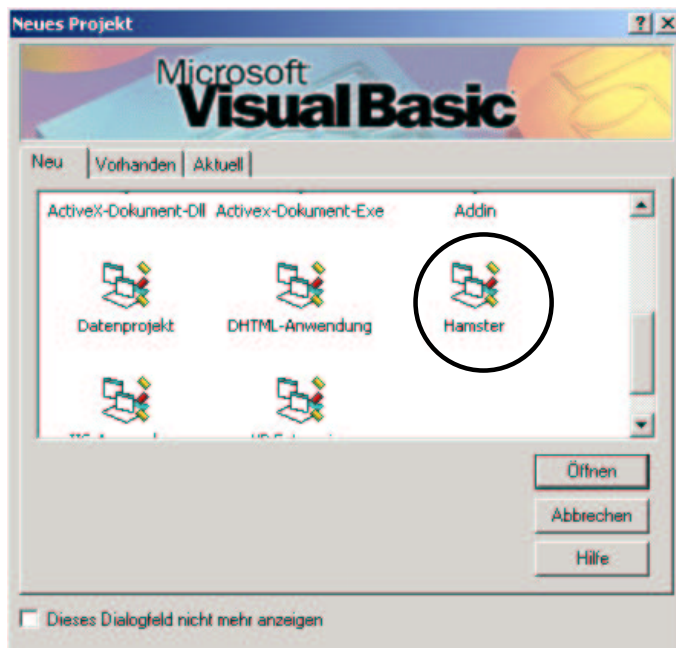
Alles weitere wird durch die Sprache *Visual Basic* definiert.

3. Verwenden des Hamstersimulators

3.1. Ein neues “Hamsterprojekt” anlegen

Starten Sie Visual Basic. Wählen Sie *Hamster* bei dem Dialog für die Auswahl eines Projekts (Dialog *Neues Projekt*):

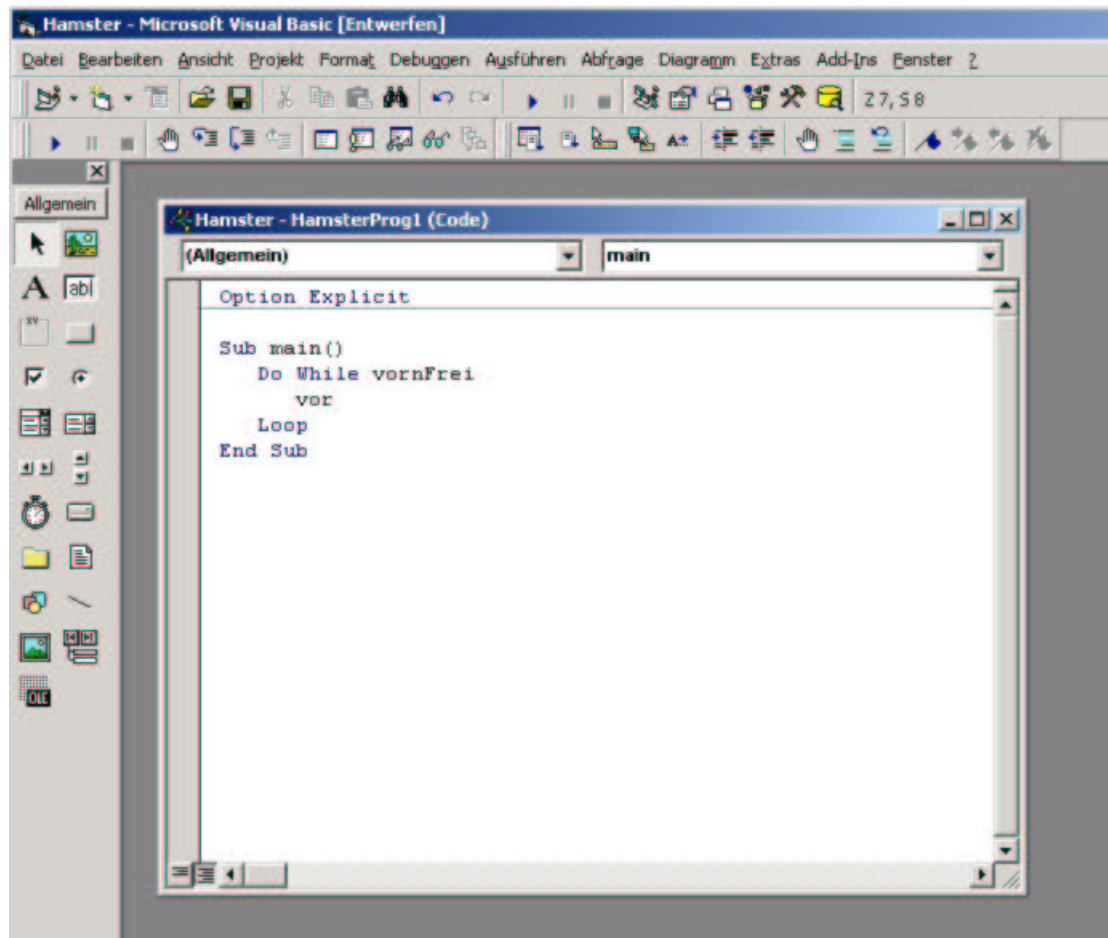
Abbildung 2. Hamsterprojekt anlegen



Damit wird ein neues Projekt mit einem Modul *Hamsterprogramm* angelegt. In dieses Modul kommen die *Hamsteranweisungen*.

Folgendes Bild zeigt ein Hamsterprogramm mit einer Schleife, die den Hamster bis zur nächsten Mauer laufen lässt:

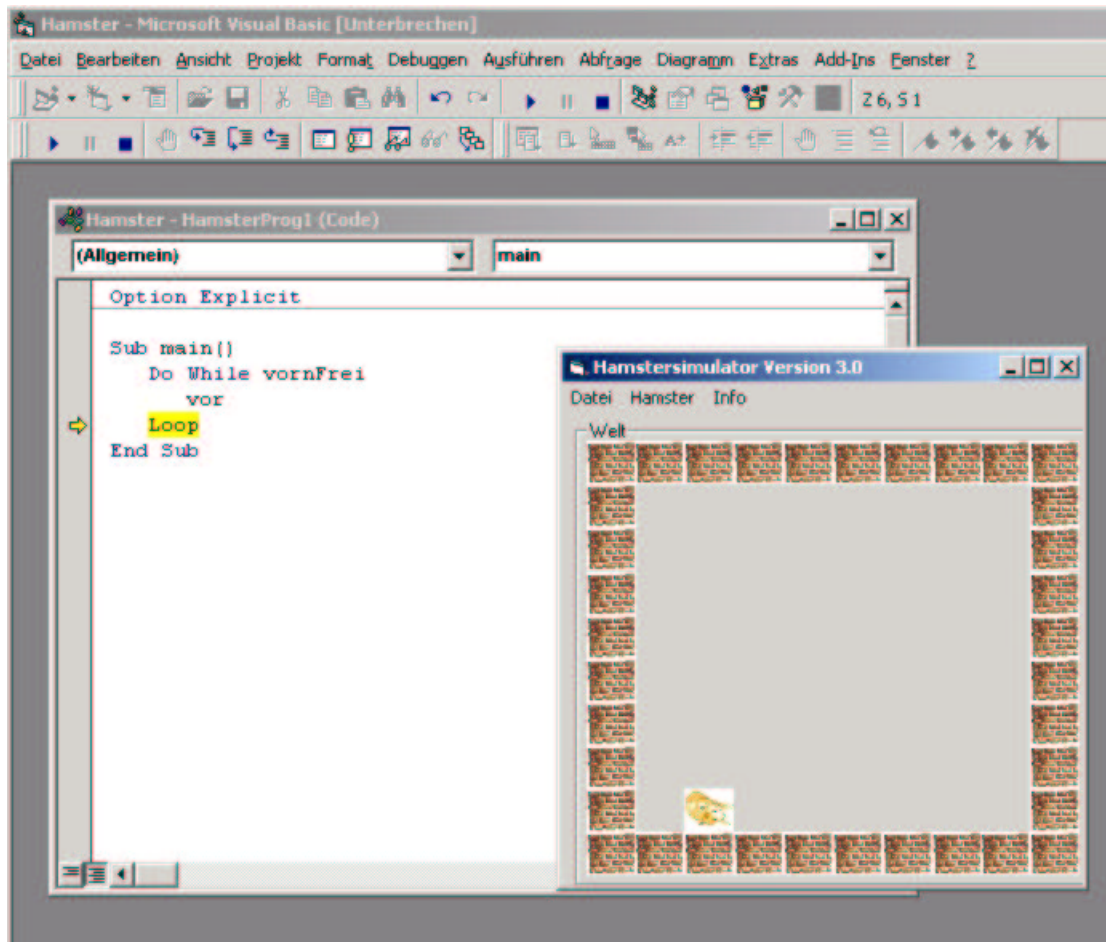
Abbildung 3. Einfaches Hamsterprogramm



3.2. Den Hamstersimulator starten

Durch das Starten des Hamsterprogrammes (*F5*) wird automatisch der Hamstersimulator gestartet:

Abbildung 4. Hamstersimulator starten

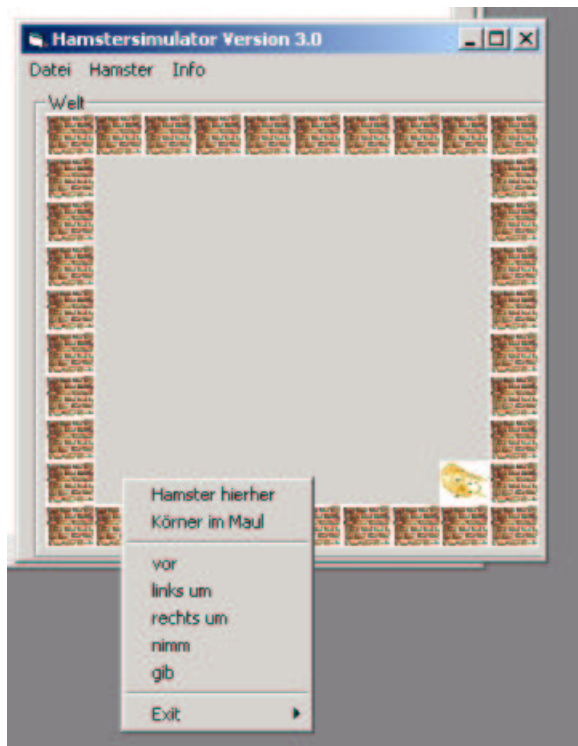


Der Simulator startet mit einer voreingestellten Welt (eine rechteckige Welt ohne Körner) oder mit der zuletzt gespeicherten Welt. Das Bild zeigt den Simulator und dahinter Visual Basic. Dort ist gerade die nächste Anweisung gelb hinterlegt (das Programm wurde angehalten).

3.3. Den Hamster interaktiv steuern

Der Hamster kann über ein Kontextmenü (rechte Maustaste) bewegt werden. Der Hamster kann zu aktueller Mausposition bewegt werden, dem Hamster können Körner gegeben werden und dem Hamster können direkt Befehle erteilt werden (hier gibt es auch ein *rechts um*, obwohl das von einem Hamsterprogramm aus nicht möglich ist).

Abbildung 5. Hamster interaktiv steuern



Noch schneller kann man den Hamster über die Tastatur steuern:

l

links um

r

rechts um

n

nimm

g

gib

v

vor

3.4. Hamsterwelt ändern/laden

Wollen Sie eine vorgegebene Hamsterwelt verwenden, dann müssen Sie das Hamsterprogramm einmal starten, sodass der Hamstersimulator gestartet wird. Das Simulatorfenster bleibt offen, bis es explizit geschlossen wird. Über *Datei->Welt* kann eine Hamsterwelt geladen werden ².

Die aktuell geladene Hamsterwelt kann verändert werden, in dem man bei gedrückter *Umschalttaste (Shift)* mit der rechten Maustaste in die Welt klickt:

Abbildung 6. Hamsterwelt bearbeiten



Dabei bedeutet

Neu

eine neue Hamsterwelt erzeugen. Dabei werden die Dimensionen der Welt in Zeilen und Spalten abgefragt.

Mauer

eine Mauer an der aktuelle (maus-) Position erzeugen

Körner

Körner an die aktuelle (Maus-) Position legen

Leer

ein leeres Feld an der aktuellen Position erzeugen

Damit kann jede beliebige Welt erzeugt werden.

3.5. Hamsterwelt speichern

Eine Hamsterwelt kann auch gespeichert werden. Diese Welt wird dann beim nächsten Start automatisch geladen. Die Welt wird im *Temp*-Verzeichnis (*C:\Temp*) unter dem Namen *Hamsterwelt.txt* gespeichert.

Speichern kann man indem man bei gedrückter *Strg-Taste (Ctrl)* mit der rechten Maustaste in die Welt klickt:

Abbildung 7. Hamsterwelt speichern/laden



Speichern

bedeutet das Speichern in die Datei *C:\Temp\Hamsterwelt.txt*

Laden

bedeutet das Laden der Datei *C:\Temp\Hamsterwelt.txt*

4. Hamsterwelten erstellen

Sie können neue Hamsterwelten erstellen, indem Sie mit dem Hamstersimulator eine Welt, wie in Hamsterwelt ändern beschrieben, erstellen und dann speichern (Hamsterwelt speichern). Die Datei `C:\Temp\Hamsterwelt.txt` können Sie dann kopieren oder umbenennen.³

Hamsterwelten können aber auch mit einem Texteditor erstellt werden. Dabei gibt es zwei verschiedene Dateiformate. Das einfachere Format, welches von den früheren Versionen des Hamstersimulators benötigt wurde, ist schon in Hamsteraufgaben beschrieben worden. Dieses Format hat aber den Nachteil, dass es keine Möglichkeit gibt, schon in der Aufgabenstellung festzulegen, wieviele Körner der Hamster im Maul haben soll. Außerdem kann der Hamster zu Beginn nur auf einem leeren Feld stehen. Deshalb wurde das Format erweitert:

```
<welt>
*****
* 1 2 3 4*
*          *
*      *W  *
* 3 ***  *
*  ***** *
*****
</welt>
<hamster>
<koerner>3</koerner>
<spalte>6</spalte>
<zeile>2</zeile>
<richtung>0</richtung>
</hamster>
```

Zwischen den “Tags” `<welt>` und `</welt>` befindet sich eine Welt im herkömmlichen Format. Die Hamsterposition (im Beispiel Osten, in Zeile 4 und Spalte 8) kann weggelassen werden, da sie ohnehin weiter unten angegeben wird.

Zwischen den “Tags” `<hamster>` und `</hamster>` werden dann die Informationen zum Hamster angegeben:

`<koerner></koerner>`

zwischen diesen Tags wird die Anzahl der Körner im Maul angegeben (ganze Zahl).

`<spalte></spalte>`

zwischen diesen Tags wird die Anfangsspalte des Hamsters angegeben (ganze Zahl).

<zeile></zeile>

die Anfangszeile des Hamsters (ganze Zahl).

<richtung></richtung>

damit kann die Richtung des Hamsters angegeben werden:

N

Norden (Hamster schaut nach oben)

O

Osten

S

Süden

W

Westen

Bei obigem Beispiel befindet sich der Hamster in der linken unteren Ecke, schaut nach rechts und hat 3 Körner im Maul (diese Angaben “überschreiben” die Angaben aus der “Welt”).

5. Hamstermodell installieren

Entpacken Sie *HamsterSetup.zip* in ein beliebiges Verzeichnis, z.B. *C:\Temp*. Starten Sie anschließend *setup.exe*. Wenn Sie den Installationspfad nicht ändern, dann wird der Hamstersimulator in das Verzeichnis *C:\Programme\HamsterAX* installiert.

Nach der Installation müssen Sie die Vorlagen für Hamsterprogramme (*Hamster.vbp* und *HamsterProg1.bas*) in das Template-Verzeichnis von Visual Basic kopieren (bei Visual Basic 6.0 ist das meistens *C:\Programme\Microsoft Visual Studio\Vb98\Template\Projects*). Dadurch können beim Start von Visual Basic neue Hamsterprogramme erstellt werden.

Fußnoten

1. Tatsächlich endet nur das *Hauptprogramm* mit dieser Anweisung. Das Programm kann aus mehreren Teilen (Unterprogrammen) bestehen, die nach

dem “End Sub” stehen.

2. nicht in der Demo-Version
3. Das Speichern ist in der Demo-Version nicht möglich.