

Christian Maurer

eGriffel

v. 17. September 2011

Dr. Christian Maurer
Keithstr. 16
10787 Berlin

<http://murus.org/>

Die Quelltexte des Modula-2 UML Sums sind mit größter Sorgfalt entwickelt und werden laufend gepflegt. Kein Programmsystem dürfte jedoch jemals frei von Fehlern sein; deshalb ist auch nicht damit zu rechnen, daß *dieses* System fehlerfrei ist: Es darf nur benutzt werden „wie es ist“.

Die Modula-2- und die Java-Quelltexte von Murus sind freie Software. Sie können sie unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation veröffentlicht, weitergeben und/oder modifizieren, entweder gemäß Version 3 der Lizenz oder (nach Ihrer Wahl) jeder späteren Version. Ihre Veröffentlichung erfolgt in der Hoffnung, daß sie Ihnen von Nutzen sein könnten – aber *ohne irgendeine Garantie*, auch ohne die implizite Garantie der *Marktreife* oder der *Verwendbarkeit für einen bestimmten Zweck*.

Der Originaltext der GPL ist im weltweiten Netz unter der Adresse www.gnu.org/licenses/gpl.html zu finden; eine deutsche Übersetzung unter www.gnu.de/documents/gpl-3.0-de.html.

Die Quelltexte von Murus sind nur zu Lehrzwecken konstruiert und haben rein akademischen Wert. Ihre Verwendung in Programmen könnte zu *Schäden* führen, z. B. zur Inbrandsetzung von Rechnern, zur Entgleisung von Eisenbahnzügen, zum GAU in Atomkraftwerken oder zum Absturz des Mondes ...

Satz: Autor mit $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\mathcal{E}\mathcal{X}$

Entia non sunt multiplicanda praeter necessitatem;
frustra fit per plura, quod fieri potest per pauciora.

Johannes Clauberg (1622 - 1665)
zugeschrieben *William von Occam (1285 - 1349)*

Vorwort

Beim **Griff1** handelte es sich ursprünglich um einen Testmodul für Folgen von Objekten variabler Größe (**VFolgen** aus dem Modul $\text{Ula}_{\text{Mod}}^{\text{Ula}} \text{Univer}^{\text{Sum}}$), wobei später für die Objekte – aus Ärger über ein m. E. völlig blödsinniges Beispiel aus dem ersten Kapitel von BARNES-KÖLLING: Java lernen mit BlueJ – ein abstrakter Datentyp **Figuren2** eingesetzt wurde.

Mittlerweile hat sich gezeigt, daß das System ganz gut geeignet ist, Erfahrungen damit zu sammeln, den klassischen Tafelanschrieb durch Rechner- und Beamereinsatz zu ergänzen (möglicherweise – in ferner Zukunft irgendwann einmal – zu ersetzen).

Ein Vorteil dieses – gegenüber dem System **E-Chalk** von Prof. Dr. R. Rojas von der Freien Universität Berlin – winzigen Systems ist die extreme didaktische Reduktion des Konzepts qua „Schlankheit“: Seine objektbasierte Architektur ist absolut überschau- und deshalb auch lehrbar; das Hauptprogramm – die Ereignissteuerung – umfaßt nur wenige hundert Quelltextzeilen.

Christian Maurer

Inhaltsverzeichnis

ALLGEMEINES

Arbeitsverzeichnis	1
--------------------------	---

EINFÜHRUNG

Leistungsmerkmale des \mathbb{E} Griff1	2
Nachteile	2
Vorteile	3
Die Figuren des \mathbb{E} Griff1	4
Die Operationen auf den \mathbb{E} Tafln	7

DAS PROGRAMM

Ergonomische Überlegungen	8
Die Bedienung des \mathbb{E} Griff1	10
Start des Programms, Farbauswahl	10
Erzeugung neuer Figuren	11
Figuren ändern, färben, markieren, kopieren	14
Figuren verschieben, löschen, zurückholen	15
\mathbb{E} Tafln laden, speichern, ausdrucken	16

ANHANG

Bemerkungen zur Architektur	17
Optionen für Erweiterungen	18

ALLGEMEINES

eGriff1 ist als Bestandteil des `ModulserverUlaSumms` nach dessen Installation oder Aktualisierung ebenfalls installiert.

Voraussetzung für die Installation des `ModulserverUlaSumms` ist die Installation von **Mocka**, dem Modula-2-Compiler der GMD (Hinweise dazu finden sich unter lwb.mi.fu-berlin.de/inf/mocka).

Arbeitsverzeichnis

Das Verzeichnis, in dem ggf. Daten abgelegt werden, ist dasjenige, aus dem heraus **eG** aufgerufen wird.

EINFÜHRUNG

Leistungsmerkmale des $\mathbb{E}_{\text{Griff}}$

Der Einsatz des $\mathbb{E}_{\text{Griff}}$ kann – unter gewissen Bedingungen – das Schreiben und Zeichnen an einer Tafel ersetzen.

Er leistet im Grundsatz das gleiche, darüberhinaus aber viel mehr: Eine $\mathbb{E}_{\text{Tafel}}$ – der simulierte „Tafelanschrieb“ –

- *kann sehr sauber gestaltet werden,*
- *ist reproduzierbar,*
- *änder- und erweiterbar sowie*
- *ausdruckbar.*

Die Vor- und Nachteile gegenüber dem Schreiben und Zeichnen an einer Tafel sind im Einzelfall natürlich – in Abhängigkeit vom Einsatzzweck – sorgfältig gegeneinander abzuwägen; dabei sind beliebige Mischformen denkbar.

Unabhängig von didaktisch-methodischen Überlegungen seien hier objektive Nach- und Vorteile des Einsatzes des $\mathbb{E}_{\text{Griff}}$ genannt:

Nachteile

- Mit *Kreide* kann jeder umgehen; die Bedienung eines *Programms* dagegen muß erlernt sein.
- Auch die Schülerinnen müssen die Bedienung lernen, wenn sie das Programm nutzen wollen oder sollen.
- Freihandzeichnungen oder -texte leiden bei der Erzeugung durch eine Maus auf der $\mathbb{E}_{\text{Tafel}}$ in der Regel mehr unter „zittriger Hand“ als beim Tafelanschrieb.
- Der Einsatz ist *von Geräten* (Rechner, Beamer) *abhängig*, die vor der Benutzung eingerichtet werden müssen.
- Es entstehen ggf. *zusätzliche Kosten* für die Geräteausstattung, oder es muß ein entsprechend ausgestatteter Lehrraum aufgesucht werden.
- Auch die *Folgekosten* sind nicht vernachlässigbar (Ersatzlampen für Beamer sind z. B. erheblich teurer als – auch farbige – Kreide).
- Das Programm setzt einen Rechner (oder Netzzugriff auf einen Server) unter *Linux* voraus (ob das *wirklich* ein Nachteil ist ...).

Vorteile

- Das Programm ist sehr *einfach zu bedienen*. Seine Steuerung ist leicht änderbar und seine ergonomischen Schwächen oder Fehler leicht korrigierbar.
- Der gesamte Quelltext steht unter GPL; damit ist die Anpaßbarkeit an individuelle Bedürfnisse gesichert.
- Es ist mit wenigen „Mausklicks“ möglich,
 - saubere Figuren, z. B. Dreiecke, Vierecke, Rechtecke, Kreise, Ellipsen und elegante Kurven zu zeichnen,
 - Text sauber zu schreiben und
 - Bilder einzubinden,
- Tafelwischen ist *auch* (im Zweifel: „Drecks-“)Arbeit.
- Der zusätzliche Kostenaufwand kann abgedeckt werden, wenn eine allgemein verfügbare *Mobilstation* (Rechner und Beamer auf einem Rollwagen oder in der Tasche) eingesetzt wird, die von mehreren Personen genutzt wird.
- Es ist damit zu rechnen, daß das Programm in absehbarer Zeit auch unter bestimmten Bedingungen unter anderen fensterorientierten Betriebssystemen läuft, wenn der lokale Rechner über eine Netzanbindung an einen Server unter Linux verfügt.
- Benutzerin oder Benutzer ist beim Einsatz *dem Auditorium zugewandt* (und fängt sich z. B. keine Papierkügelchen von hinten ein).
- Die \mathbb{E} Tafeln können – auch an einem anderen – Rechner in Ruhe optimal *vor- und nachbereitet* und „*durchgestylt*“ werden.
- Die \mathbb{E} Tafeln sind beliebig *wiederverwendbar*. („So sah unsere Tafel vorgestern aus.“)
- Die langfristige *Pflege* (d. h. die Änderung auf der Grundlage von Erfahrungen, die Anpassung an andere Lehrsituationen und die Weiterentwicklung) der \mathbb{E} Tafeln ist jederzeit möglich.
- Die \mathbb{E} Tafeln können *ausgedruckt* und folglich *vielfältigt* werden.

Die Figuren des $\mathcal{E}_{\text{Griff}}$

Der $\mathcal{E}_{\text{Griff}}$ verwaltet folgende Arten von Figuren:

- *Punktfolgen*,
- *Streckenzüge*,
- *Polygone*,
- *Kurven*,
- *Geraden*,
- *Rechtecke*,
- *Kreise*,
- *Ellipsen*,
- *Texte* und
- *Bilder*.

Im folgenden werden diese Figuren einzeln erläutert. Vorab jedoch ein paar Worte zum

Farbkonzept

Alle Figuren können in verschiedenen Farben dargestellt werden (zur Zeit sind gut 100 Farben auswählbar).

Die *Hintergrundfarbe* des Bildschirms ist standardmäßig schwarz, kann aber „auf Knopfdruck“ auf weiß umgestellt werden

Punktfolgen

Es handelt sich um *Folgen einzelner Punkte* („Pixel“ auf dem Bildschirm), von denen je zwei aufeinanderfolgende, wenn sie als Pixel nicht benachbart sind, durch eine Strecke verbunden sind.

Die Punkte werden durch die Bewegung eines Zeigeeinstruments (zur Zeit nur der Maus; in Zukunft vielleicht auch eines Graphiktablets) selbständig (d. h. bei der Bewegung vom Programm) erzeugt, bis die Eingabe von Benutzerin beendet wird (die Reihenfolge der Punkte ist durch die zeitliche Abfolge bei der Erzeugung gegeben).

Punktfolgen realisieren damit „Freihandzeichnungen“ (die natürlich auch Texte darstellen können).

Streckenzüge

Streckenzüge sind Folgen von Strecken, bei denen der Endpunkt jeweils einer Strecke mit dem Anfangspunkt der nächsten Strecke zusammenfällt, sofern noch eine folgt (die Reihenfolge der Strecken durch die zeitliche Abfolge der Setzung ihrer Endpunkte gegeben).

Im Prinzip sind Streckenzüge also das gleiche wie Punktfolgen; der Unterschied besteht darin, daß die Randpunkte der Strecken von Benutzerin einzeln gesetzt statt durch die Mausbewegung erzeugt werden.

Polygone

Polygone sind „geschlossene“ Streckenzüge: Der Anfangspunkt ihrer ersten Strecke fällt mit dem Endpunkt ihrer letzten zusammen.

Damit gehören *Dreiecke*, *Vierecke*, *Fünfecke* usw. zu den Figuren.

Kurven

Kurven sind durch Bezier-Polynome definiert, die durch – von Benutzerin einzeln gesetzte – Punktfolgen, begrenzt durch eine gewisse Maximalzahl, definiert sind.

Eine Kurve hat den ersten und letzten Punkt der Folge als Randpunkte; die Punkte dazwischen sind die „Stützpunkte“ des Polynoms, dessen Grad n um 1 niedriger als die Anzahl der Punkte ist:

$$z(t) = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i z_i \text{ für } 0 \leq t \leq 1 \text{ und } z_0, z_1, \dots, z_n \in \mathbb{C}$$

Geraden

Wegen der „Endlichkeit“ der Größe des Bildschirms sind *Geraden* durch zwei Punkte gegebene *Strecken*, deren Endpunkte am Rand des Bildschirms liegen.

Rechtecke

Zu den Bildschirmkanten parallele *Rechtecke*, gegeben durch zwei Punkte, der linken oberen und der rechten unteren Ecke.

Kreise

Kreise sind durch ihren Mittelpunkt und ihren Radius gegeben.

Ellipsen

Nur *Ellipsen* mit *achsenparallelen* Halbachsen, die durch ihren Mittelpunkt und die Längen ihrer Halbachsen gegeben sind.

Texte

Alphanumerische Texte, d. h. auch *Zahlen* (z. B. im einfachsten Fall Ziffernfolgen). Sie können verschiedene *Schriftgrößen* haben.

Bilder

Bilder sind Graphikdateien im ppm-Format. Damit sie vom eGriff benutzt werden können, müssen sie natürlich genügend klein sein.

Ihre Bearbeitung ist mit den Routinen aus dem netpbm-Paket möglich; z. B. ihre Skalierung mit `pamscale` und die Umwandlung zwischen diesem Format und dem jpeg-Format mit `pnmtjpeg` bzw. `jpegtopnm`.

Gefüllte Figuren

Rechtecke, *Kreise*, *Ellipsen* und *kreuzungsfreie konvexe* (unter *X* auch beliebige Polygone) können in dem Sinne *gefüllt* sein, daß alle Pixel im Inneren der Figur auf die Farbe (des Randes) der Figur gesetzt sind.

 eTafln

Mögliche *Option für eine Erweiterung*:

Es war einmal ein Vater, der hatte sieben Kinder; und diese sieben Kinder, die sagten „Vater, erzähl’ uns ’mal ’ne Geschichte.“

Und der der Vater erzählte: „Es war einmal ein Vater ... “

Die Operationen auf den $\mathbb{E}_{\text{Tafln}}$

Mit dem $\mathbb{E}_{\text{Griff}}$ lassen sich auf den $\mathbb{E}_{\text{Tafln}}$ – den mit dem $\mathbb{E}_{\text{Griff}}$ auf dem Bildschirm simulierten „Tafelbildern“ – im einzelnen folgende Operationen durchführen:

- Einzelne Figuren
 - *erzeugen,*
 - *ändern,*
 - *färben,*
 - *verschieben,*
 - *löschen* sowie
 - *markieren und entmarkieren*
- alle markierten Figuren
 - *kopieren,*
 - *verschieben,*
 - *löschen,*
 - *und entmarkieren,*
- ganze $\mathbb{E}_{\text{Tafln}}$
 - *markieren,*
 - *löschen,*
 - *dazuladen* und
 - *speichern*
- sowie alle Erzeugungen und Löschungen auf einer \mathbb{E}_{Taf}
 - *rückgängig machen.*

Neue Figuren werden in der aktuellen Art, der aktuellen Farbe und – falls es ein Text ist – in der aktuellen Schriftgröße erzeugt, die anfangs Standardwerte haben und mit bestimmten Kommandos jederzeit geändert werden können.

DAS PROGRAMM

Ergonomische Überlegungen

Der \mathcal{E} Griffel ist nach ergonomischen Gesichtspunkten konzipiert, Das Programm ist darauf abgestimmt, daß es – abgesehen von Texteingaben – mit der einen Hand möglichst bewegungsfrei über der Tastatur und mit der anderen Hand auf der Maus bedient werden kann.

Optimal gilt das allerdings nur, wenn der \mathcal{E} Griffel mit einer 3-Tasten-Maus und in einer tty-Konsole betrieben wird. Der im Abschnitt über den *Konsolenbetrieb* auf S. 2 angegebene Vorteil gilt für den \mathcal{E} Griffel in besonderem Maße:

Nur *so* steht der gesamte Bildschirm für eine \mathcal{E} Tafel zur Verfügung! Auch wer öfter *Bilder* auf \mathcal{E} Tafeln darstellen will, sollte aus Effizienzgründen dem Konsolenbetrieb den Vorzug geben.

Unter diesen Bedingungen kommt der \mathcal{E} Griffel im wesentlichen mit wenigen Tasten

- Tabulatortaste \leftarrow ,
- Steuerungstaste **Strg**, Metataste **Alt**
- Leertaste

aus, die dicht nebeneinanderliegen und daher nach kurzer Gewöhnung „blind“ benutzbar sind, sodaß die Konzentration auf die \mathcal{E} Tafel nicht durch den ständigen Blickwechsel zwischen Bildschirm, Tastatur und Maus unterbrochen wird.

Im Konsolenbetrieb wirkt die Kontrolltaste **Strg** grundsätzlich wie die Umschalttaste \uparrow , da **Strg** „blind“ leichter als \uparrow erreichbar ist.

Die Fenstermanager der gängigen graphischen Oberflächen fangen dagegen bestimmte Tastenkombinationen mit der Kontrolltaste **Strg** und der Metataste **Alt** ab und verwenden sie zur Manipulation von Fenstern; eine 2-Tasten-Maus hat keine *mittlere* Taste.

Deshalb müssen in diesen Fällen alternative Tasten

- Entfernungstaste **Entf**,
- Funktionstasten **F5** und **F6**

verwendet werden, was die obigen Überlegungen etwas konterkariert.

Schließlich sei – aus naheliegenden Gründen – davon abgeraten, in einer graphischen Oberfläche das Fenster, in dem sich die \mathcal{E} Tafel beim

Programmstart öffnet, durch „Verziehen der Fensterrahmen“ mit der Maus in seiner Größe zu verändern!

Hilfe kurzgefaßt

Der Hilfe-Bildschirm erscheint auf Druck der Funktionstaste F1 und verschwindet beim Bewegen der Maus oder Drücken irgendeiner Taste. Er zeigt die folgenden knappen Hinweise zur Bedienung:

für neue Figuren Art (Farbe) auswählen: (Umschalt- +) Leer- oder Rolltaste
 Schriftgröße auswählen: Alt- + Leer- oder Rolltaste
 neue Figur erzeugen: linke Maustaste;
 Streckenzüge, Polygone und Kurven
 mit rechter Maustaste abschließen,
 Texteingabe mit Eingabetaste beenden
 Figur ändern: Alt- + linke Maustaste; die 'Punkte'
 mit rechter Maustaste verschieben,
 mit linker Maustaste abschließen
 Figur färben: Umschalt-, Alt- + linke Maustaste
 (alle) Figur(en) markieren: (Umschalt- +) F5-Taste oder
 Umschalt- + (Alt- +) mittl. Maustaste
 (alle) Figur(en) entmarkieren: (Umschalt- +) F6-Taste oder
 Umschalt- + (Alt- +) mittl. Maustaste
 markierte Figuren kopieren: Umschalt- + linke Maustaste
 (markierte) Figur(en) verschieben: (Umschalt- +) rechte Maustaste
 (markierte) Figur(en) löschen: (Umschalt- +) Entfernungstaste oder
 (Umschalt-,) Alt- + rechte Maustaste
 letzte gelöschte Figur zurückholen: Rücktaste (<-)
 letzte erzeugte Figur löschen: Umschalt- + Rücktaste (<-)
 letzte gelöschte mark. Fig. zurückholen: Alt- + Rücktaste (<-)
 Hintergrundfarbe umschalten (auswählen): (Umschalt- +) F4-Taste
 Figuren aus eTafl holen und markieren: F7-Taste
 markierte Figuren in eTafl ablegen: F8-Taste
 eTafl ausdrucken: Drucktaste
 eGriffl beenden: Schlußtaste (Esc)

Steuerungstaste (Strg) wirkt wie Umschalttaste

Die Bedienung des $\mathbb{E}_{\text{Griff1}}$

Der Hilfe-Bildschirm stellt nur eine Kurzübersicht dar.

Er hat lediglich den Zweck, Benutzerin schnell – d. h. mit einem Tastendruck – ein vergessenes Kommando nachsehen zu lassen, ohne erst in einer Bedienungsanleitung blättern zu müssen.

Die Bedienung des $\mathbb{E}_{\text{Griff1}}$ wird im folgenden detailliert erläutert.

Start des Programms

Der $\mathbb{E}_{\text{Griff1}}$ wird mit dem Kommando `eG` gestartet, dem der Name der \mathbb{E}_{Taf1} , abgelegt in einer Datei dieses Namens, an den das Suffix `.egr` angehängt ist, als Parameter mitgegeben werden kann. Falls das der Fall ist, erscheint dieser Name – andernfalls der provisorische Name `temp` – im Feld des \mathbb{E}_{Taf1} -Namens links oben auf dem Bildschirm; er kann editiert werden.

Wenn die Eingabe mit der Eingabetaste \blacktriangleleft abgeschlossen wird, wird die letzte \mathbb{E}_{Taf1} geladen, die unter diesem Namen gespeichert ist, falls es einen solchen gibt; andernfalls ist die \mathbb{E}_{Taf1} jetzt leer.

Statt einer Eingabe kann aber auch mit der Funktionstaste F2 in ein Menü verzweigt werden, in dem unter allen gespeicherten \mathbb{E}_{Taf1} n mit den Pfeiltasten gewählt und die gewünschte mit der Eingabetaste übernommen wird.

Wenn kein Name eingegeben oder die Eingabe mit der Schlußtaste Esc abgeschlossen wird, wird das Programm sofort wieder beendet.

Auswahl der Hintergrundfarbe der \mathbb{E}_{Taf1}

Mit der Funktionstaste F4 wird zwischen dem – standardmäßigen – *schwarzen* Hintergrund der \mathbb{E}_{Taf1} und einem *hellweißen* umgeschaltet.

Wird diese Taste zusammen mit der Umschalttaste \uparrow gedrückt, erscheint in Höhe des Mauszeigers ein farbiger Streifen, aus dem per Klick mit der linken Maustaste die *Hintergrundfarbe* ausgewählt werden kann.

Die Auswahl wird durch einen Klick mit der linken Maustaste außerhalb des Streifens abgebrochen; in diesem Fall bleibt die Hintergrundfarbe erhalten.

Auswahl von Art und Farbe neuer Figuren

Die Betätigung der Leertaste oder der Taste A bewirkt, daß an der Stelle des Mauszeigers ein Menü erscheint, aus dem mit der Maus die *aktuelle Art* der Figuren ausgewählt werden kann.

Die aktuelle Art kann auch mit den Pfeiltasten \uparrow und \downarrow und der Anfangs- und Endetaste Pos1 und Ende ausgewählt und mit der Eingabetaste \leftarrow bestätigt werden.

Die Auswahl wird mit der Schlußtaste Esc abgebrochen; dann bleibt die alte aktuelle Art erhalten.

Neue Figuren werden – bis zur Auswahl einer anderen aktuellen Art - in dieser Art gezeichnet.

Wenn die Leertaste zusammen mit der Umschalttaste \uparrow gedrückt wird, kann – wie bei der Auswahl der Hintergrundfarbe – die *aktuelle Farbe* ausgewählt werden, in der neue Figuren erzeugt werden.

Neue Figuren erzeugen

Punktfolgen

Mit einem Druck auf die linke Maustaste wird die Erzeugung einer neuen Figur an der Stelle des Mauszeigers begonnen.

Ist die aktuelle Art eine *Punktfolge*, wird – der Mausbewegung folgend – solange „gezeichnet“, bis die Maustaste losgelassen wird. Sollte die maximal mögliche Anzahl der Punkte der Folge vorher schon erreicht sein, wird die Zeichnung automatisch beendet.

Je schneller die Maus bewegt wird, desto deutlicher wird der Effekt sichtbar, daß die einzelnen – von der Maus erkannten – Punkte jeweils durch Strecken verbunden werden: Die Figur wird etwas „eckiger“.

Strecken und Streckenzüge

Handelt es sich bei der aktuellen Art um *Strecke(n)* (*Strecken* bzw. *Streckenzüge*), beginnt die Figur beim ersten Mausklick.


Weitere Klicks mit der linken Maustaste an anderen Stellen setzen die nächste Strecke; Bewegungen zwischen den Mausklicks führen die jeweils letzte Strecke bis zum nächsten Mausklick mit.

Ein Streckenzug wird durch einen Klick mit der rechten Maustaste beendet.

Eine *einfache Strecke* wird also mit der folgenden Kommando-
folge erzeugt: Bewegung der Maus zum Anfangspunkt – Klick mit
der linken Maustaste – Bewegung der Maus zum Endpunkt – Klick
mit der rechten Maustaste.

Polygone

Polygone werden ganz ähnlich erzeugt; der einzige Unterschied ist,
daß ab dem zweiten Mausklick die aktuelle Mausposition automatisch
mit der Anfangsposition verbunden wird.

Wenn ein Polygon konvex ist (diese Einschränkung gilt nur für
den Betrieb des  in einer Konsole), wird es in der gleichen Farbe
wie sein Rand *gefüllt*, falls der abschließende Klick mit der rechten
Maustaste zusammen mit der Umschalttaste \uparrow erfolgt.

Kurven

Kurven werden im Grunde wie Streckenzüge erzeugt: Klick mit
der linken Maustaste; Fortsetzung durch Mausbewegung und weitere
Klicks mit der linken Maustaste; die Fixierung erfolgt mit einem Klick
der rechten Maustaste.

Wenn bei der Fortsetzung die maximal mögliche Zahl der Stütz-
punkte erreicht wird, wird die Erzeugung automatisch beendet.

Anders als bei Streckenzügen paßt sich während der Erzeugung
ständig die gesamte Kurve den gesetzten Stützpunkten an; es erfordert
etwas Übung und Erfahrung, bis Benutzerin „den Bogen heraushat“.

Geraden

Ein Punkt einer *Geraden* wird mit einem Druck auf die linke Maus-
taste gesetzt, wobei die Gerade als Horizontale erscheint; die Bewe-
gung der Maus führt den zweiten Punkt und damit die Gerade mit,
bis sie durch Loslassen der Maustaste fixiert sind.

Rechtecke

Bei der Erzeugung von *Rechtecken* wird ähnlich wie bei Geraden
verfahren: Ein Druck mit der linken Maustaste setzt eine Ecke des

Rechtecks; solange die Maus bewegt wird, wird auf ihrer Position die diagonal gegenüberliegende Ecke und damit das Rechteck mitgeführt, bis das Rechteck durch Loslassen der Maustaste fixiert wird.

Wird während des Loslassens der Maustaste die Kontrolltaste **Strg** gedrückt, wird es in der gleichen Farbe wie sein Rand *gefüllt*.

Kreise

Kreise werden nach dem gleichen Prinzip erzeugt: Ein Druck mit der linken Maustaste setzt den Mittelpunkt; die Bewegung der Maus führt zur Mitführung eines Kreises durch die Mausposition; sowie die Maustaste losgelassen wird, ist der Kreis fixiert.

Kreise können wie Rechtecke *gefüllt* werden.


Ellipsen

Ellipsen werden wie *Kreise* erzeugt; der einzige Unterschied ist, daß die Mausbewegung eine Ecke des umschreibenden Rechtecks setzt.

Ellipsen können wie Rechtecke *gefüllt* werden.


Texte

Ist die aktuelle Art ein *Text*, erscheint nach einem Klick mit der linken Maustaste an der Stelle des Mauszeigers ein blinkender Cursor. Der Text wird mit der Tastatur eingegeben, wobei er – wie mit einem üblichen Editor – komfortabel verändert werden kann (wie im Kapitel über den Feldeditor von das Modell^{Ula}_{Uli}^{Sum} beschrieben).

Wenn die Eingabe leer ist oder mit einer anderen Taste als der Eingabetaste  beendet wird, ist die Erzeugung abgebrochen.

Bilder

Ist die aktuelle Art ein *Bild*, erscheint – wie bei einem Text – nach einem Klick mit der linken Maustaste ein blinkender Cursor, womit die Aufforderung verbunden ist, den Namen des Bildes einzugeben.

Nach Abschluß der Eingabe mit der Eingabetaste  erscheint das in der entsprechenden Datei, deren Dateiname der Name des Bildes mit angehängtem Suffix `.ppm` ist, enthaltene Bild mit der Mausposition als linker oberer Ecke – ggf. so weit verschoben, daß es auf die

$\mathcal{E}_{\text{Taf1}}$ paßt – sofern eine solche Datei vorhanden ist und das Bild nicht größer als die $\mathcal{E}_{\text{Taf1}}$ ist.

Bei leerer Eingabe oder Abschluß mit der Schlußtaste Esc wird abgebrochen.

Figuren ändern

Sitzt der Mauszeiger auf einer Figur, führt ein Klick mit der linken Maustaste zusammen mit der Metataste Alt dazu, daß die Punkte, die die Figur charakterisieren, sichtbar werden.

Sie können einzeln mit der rechten Maustaste „gefaßt“ und verschoben werden; die Figur paßt sich dabei an. Die Änderung wird durch einen Klick mit der linken Maustaste abgeschlossen.

Figuren färben

Wenn die linke Maustaste zusammen mit der Kontroll- und der Metataste Strg und Alt gedrückt wird, wird die Figur unter der Maus in der *aktuellen Farbe* eingefärbt.

Wenn die Maus dabei bewegt wird, werden alle Figuren, über die die Maus „fährt“, eingefärbt.

Figuren markieren und entmarkieren

Eine unter dem Mauszeiger sitzende Figur wird mit einem Klick der mittleren Maustaste oder mit der Funktionstaste F5 markiert; in Verbindung mit der Umschalttaste \uparrow wird die Markierung entfernt.

Unabhängig von der Position der Maus werden *alle Figuren* beim zusätzlichen Drücken der Metataste Alt markiert bzw. entmarkiert.

Die markierten Figuren erscheinen während eines Klicks auf die mittlere Maustaste in ihren Komplementärfarben.

Für den Fall, daß nur eine 2-Tasten-Maus vorhanden ist, wird mit den Funktionstasten F5 markiert und F6 entmarkiert; zusammen mit der Umschalttaste \uparrow wirkt das auf *alle Figuren* in der $\mathcal{E}_{\text{Taf1}}$.

Figuren kopieren

Wenn eine oder mehrere Figuren *markiert* sind, bewirkt ein Klick mit der linken Maustaste bei gleichzeitigem Drücken der Kontrolltaste

Strg, daß sie an die Stelle des Mauszeigers kopiert werden, sofern der Platz im Bild dafür reicht.

Figuren verschieben

Mit der rechten Maustaste können einzelne Figuren – markiert oder nicht – „angefaßt“ und bei gedrückter Maustaste mit der Bewegung der Maus verschoben werden.

Wird dabei die Umschalttaste \uparrow gedrückt, werden *alle markierten* Figuren verschoben, falls mit dem Mauszeiger eine von ihnen angefaßt ist, soweit das im Rahmen der E_{Taf1} möglich ist.

Figuren löschen

Sitzt der Mauszeiger auf einer Figur, wird sie durch Drücken der Entfernungstaste **Entf** gelöscht. Wenn die Maus dabei bewegt wird, werden alle von der Maus überstrichenen Figuren gelöscht.

Im Konsolenbetrieb funktioniert das auch mit einem Klick mit der rechten Maustaste, wenn dabei die Metataste **Alt** gedrückt ist.

Wenn zusätzlich die Umschalttaste \uparrow gedrückt wird, werden – unabhängig von der Mausposition – *alle markierten Figuren* gelöscht.

Gelöschte Figuren zurückholen

Mit der Rücktaste \leftarrow wird die jeweils letzte gelöschte Figur wieder restauriert; falls sie vorher markiert war, ist sie das jetzt nicht mehr.

Zusammen mit der Metataste **Alt** werden alle Figuren aus der letzten Löschung *aller markierten Figuren* wieder restauriert.

Letzte erzeugte Figuren löschen

Wenn die Rücktaste \leftarrow zusammen mit der Umschalttaste \uparrow gedrückt wird, wird die jeweils letzte erzeugte Figur (unwiderrufflich) gelöscht.

Figuren aus einer $\mathcal{E}_{\text{Taf1}}$ laden

Mit einem Druck auf die Funktionstaste F7 werden alle Figuren einer bestimmten $\mathcal{E}_{\text{Taf1}}$ in die aktuelle hineinkopiert, wobei genau diese Figuren markiert werden. Dazu tut sich das Feld für den $\mathcal{E}_{\text{Taf1}}$ -Namen auf, in dem der Name der zu ladenden $\mathcal{E}_{\text{Taf1}}$ eingegeben wird.

Die Eingabe wird mit der Eingabetaste \leftarrow abgeschlossen; beim Abschluß mit der Schlußtaste Esc wird der Vorgang abgebrochen.

Figuren in einer $\mathcal{E}_{\text{Taf1}}$ speichern

Durch Betätigung der Funktionstaste F8 können alle markierten Figuren, sofern es welche gibt, in einer $\mathcal{E}_{\text{Taf1}}$ abgelegt werden; dazu wird dessen Name im Feld für den $\mathcal{E}_{\text{Taf1}}$ -Namen eingegeben (eine unter diesem Namen vorhandene $\mathcal{E}_{\text{Taf1}}$ wird überschrieben).

$\mathcal{E}_{\text{Taf1}}$ ausdrucken

Auf Betätigung der Drucktaste Druck wird der Inhalt der $\mathcal{E}_{\text{Taf1}}$ – mit weißem Hintergrund – ausgedruckt (sofern CUPS installiert ist).

Beendigung des Programms

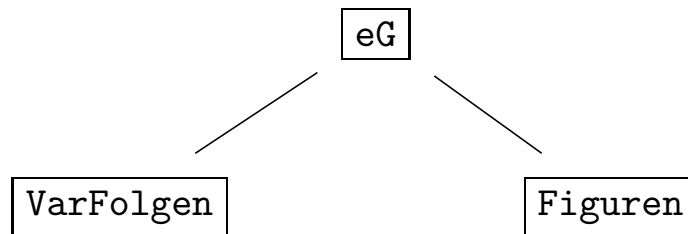
Der $\mathcal{E}_{\text{Griff1}}$ wird mit der Schlußtaste Esc beendet.

Falls die $\mathcal{E}_{\text{Taf1}}$ nun unter einem neuen Namen gespeichert werden soll, kann das im Namensfeld korrigiert werden; ansonsten bleibt die aktuelle $\mathcal{E}_{\text{Taf1}}$ unter dem Namen erhalten, der ihr beim Start des Programms gegeben wurde.

ANHANG

Bemerkungen zur Architektur des \mathcal{E} Griff1

Die Architektur des \mathcal{E} Griff1 ist trivial:



Das „Hauptprogramm“ `eG.mod` ist eine Ereignissteuerung in Form einer *Ereignisschleife*, in der zwei abstrakte Datentypen zusammengebunden werden:

- `VarFolgen` – *Folgen von Objekten variabler Größe* – wobei
- `Figuren` – die beschriebenen *Figuren* – für die Objekte eingesetzt wurden.

Optionen für Erweiterungen

Im folgenden einige Ideen:

- mehr Farben
- unterschiedliche Strichstärken
- jpg- und andere Formate im Programm verarbeiten: `libnetpbm`
- nicht pixelweise ausdrucken, sondern die Figuren mit Hilfe von `Postscript` darstellen
- Zahlen als Figuren: Rechenoperationen einbinden
- Terme als Figuren: Auswertungen einbinden
- das interne Format (in Pixeln) auf (lange) reelle Zahlen umstellen
- den `Griff` mehrbenutzerfähig machen
- ... ? ...