

Einführung zur Aufgabengruppe 4

- **vogle – A very ordinary graphics learning enviroment**
- **Beispiel zu vogle**
- **Kommunikation mit anderen Rechnern**
 - **Datentransfer**
 - **Archivieren**
 - **Dialogtransfer**

vogle – A very ordinary graphics learning environment

vogle ist eine Bibliothek für Funktionen zum Darstellen von Grafikobjekten in 2 und 3 Dimensionen. Sie ermöglicht das Zeichnen von Geraden, Kreisen, Kurven, Kreisbögen, Polygone und auch Texten unabhängig von der jeweiligen Gerätekonfiguration. Es bestehen weiterhin die Möglichkeiten, Font festzulegen, Farben zu ändern, Füllfarben anzugeben und anderes mehr.

Es gibt eine Schnittstelle zu C, Fortran und Pascal.

Die Bibliothek ist frei verfügbar auf „The Eric H. Echidna Memorial Home Page“ unter <http://www.wumpus.com.au/eric/>.

Im Verzeichnis <http://www.informatik.uni-leipzig.de/cgip/lehre/ss03/vogle/> finden Sie die gepackte **vogle** - Bibliothek **vogleansi.tar.gz**, im Unterverzeichnis **docs/** die Dokumentation, im Unterverzeichnis **examples/** zahlreiche Beispiele und Installationshinweise zu **vogle** unter Windows.

In den IfI-Pools sind die Bibliothek unter `/usr/local/lib/libvogle.a` und die Headerdatei unter `/usr/local/include/vogle.h` verfügbar.

Vorgabe der Fenstergröße und Fensterlage

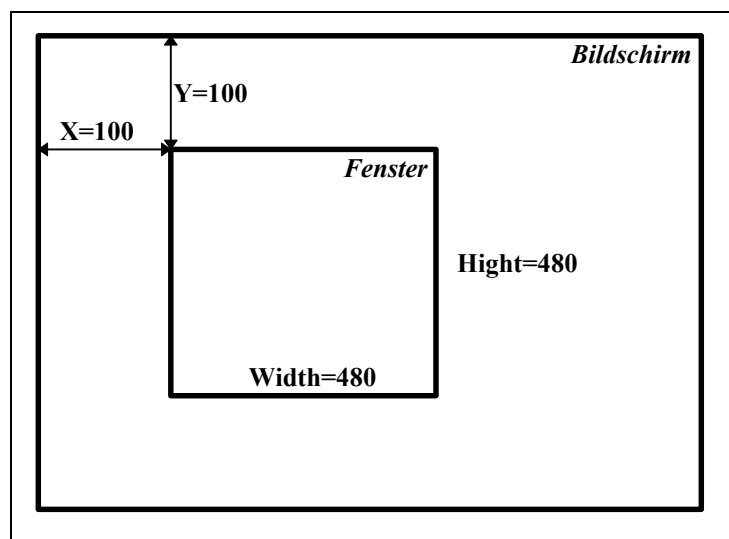
```
void prefsiz( int PrefWidth, int PrefHeight);  
/* Einstellen der bevorzugten Fenstergröße */
```

```
void prefpos( int PrefX, int PrefY);  
/* Einstellen der bevorzugten Fensterposition */
```

```
/* Bsp.: */  
/* prefsiz( 480, 480); */  
/* prefpos( 100, 100); */
```

Öffnen des vogle-Fensters

```
void vinit( char *DeviceName);  
/* Initialisieren eines vogle-Devices. */  
/* Hier lautet der DeviceName "X11": X11-Fenster */
```



Einstellen der Darstellungsfarbe

```
void color( int aColor); /* Farbe einstellen ( WHITE, BLACK, RED, GREEN, ...)*/*
```

Löschen des Fensters

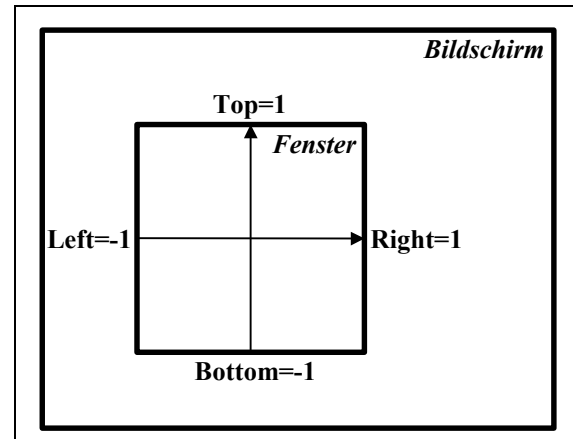
```
void clear(); /* Ausgabebereich löschen */
```

Schließen des vogle-Fensters

```
void vexit();  
/* Schließen des vogle-Devices */
```

Festlegen eines ViewPorts (Ausgabebereich)

```
void viewport( float Left, float Right, float Bottom, float Top)
/* Festlegen der Eckkoordinaten eines Viewports */
/* (Ausgabebereich); */
/* Die Koordinaten sind Screenkoordinaten, dabei hat das */
/* maximal eingeschriebene Quadrat die Eckkoordinaten */
/* (-1,1),(1,1),(-1,-1),(1,-1) */
/* Bsp.: viewport( -1, 1, -1, 1); */
```

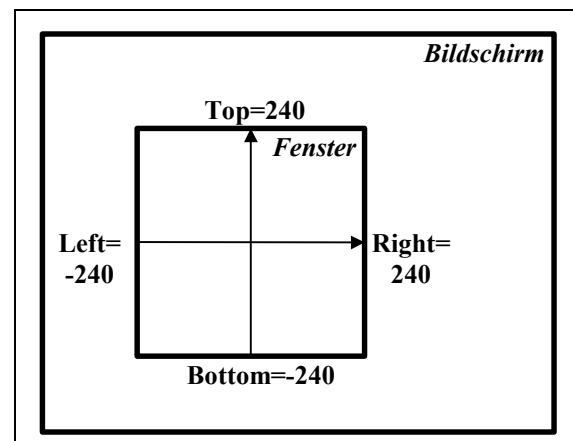


Abschneiden einer Ausgabe außerhalb des ViewPort

```
void clipping( int OnOff); /* Clipping ein/aus - schalten */
```

Festlegen eines Koordinatensystems für den ViewPort

```
void ortho2( float Left, float Right, float Bottom, float Top)
/* Festlegen des Koordinatensystems eines Viewports */
/* (Ausgabebereich), 2-dimensional-orthogonal durch */
/* Angabe der Eckkoordinaten in Weltkoordinaten */
/* - also der Koordinaten, in denen sich */
/* das Geschehen abspielt, welches man darstellen will */
/* Bsp.: ortho2( -240, 240, -240, 240); */
```



Ausgabe von Graphik und Text

```
void move2( float X, float Y);
/* Graphikposition auf ( X, Y) setzen (Weltkoordinaten) */
```

```
void draw2( float X, float Y);
/* Linie von der aktuellen Graphikposition nach (X,Y) ziehen, */
/* Graphikposition auf ( X, Y) setzen (Weltkoordinaten) */
```

```
void rect( float XA, float YA, float XB, float YB);
/* Zeichnen eines Rechtecks mit linker oberer Ecke ( XA, YA) */
/* und rechter unterer Ecke ( XB, YB) (Weltkoordinaten) */
```

```
void textsize( float cWidth, float cHeight);
/* Breite und Höhe eines Textzeichens setzen (Weltkoordinaten) */
```

```
void drawstr( char *aString);
/* Ausgabe eines Strings an der aktuellen Graphikposition, */
/* die linke untere Ecke des dargestellten Strings wird auf */
/* die aktuelle Graphikposition positioniert (Weltkoordinaten) */
```

Beispiel: Simple (Simple.c, s.h., s.c, makefile)

Demo: Kurven

Monika Meiler
Programmier-Praktikum

Zeichnen eines Dreiecks
Kreise, Ellipsen und Astroide

Aufgabengruppe 4

Kommunikation mit anderen Rechnern

Bei dem Datenaustausch zwischen zwei Rechnern unterscheidet man zwischen dem **Datentransfer** und dem **Dialogtransfer**.

Datentransfer

Um sich Dateien von entfernten Rechnern zu holen, muss man wissen, auf welchem Rechner diese liegen, Zugriff auf diesen haben und den Dateitransfer durchführen können.

ftp	file transfer protocol	Datentransfer
------------	-------------------------------	----------------------

ftp userv1.informatik.uni-leipzig.de

Nach dem Kommando **ftp** kann der Name oder die Adresse des entfernten Rechners folgen, an dem man ein Login hat und arbeiten möchte.

Befehl	Beschreibung
open <i>Hostname</i>	Öffnen einer Verbindung
help	Kommandoübersicht
ls, dir	Verzeichnisinhalt des entfernten Rechners
lls, ldir	Verzeichnisinhalt des lokalen Rechners (implementationsabhängig)
pwd	Aktueller Pfad des entfernten Rechners
cd <i>Verzeichnisname</i>	Verzeichniswechsel im entfernten Rechner
lcd <i>Verzeichnisname</i>	Verzeichniswechsel im lokalen Rechner (implementationsabhängig)
ascii	Einstellen des Übertragungsmodus für ASCII
binary	Einstellen des Binärübertragungsmodus (ist schneller)
get	Empfangen einer Datei
mget	Empfangen mehrerer Dateien (Wildcards erlaubt)
put	Senden einer Datei
mput	Senden mehrerer Dateien (Wildcards erlaubt)
close	Beendet die Hostverbindung
bye, quit	Beenden der ftp – Sitzung

anonymous-ftp:

Mit einem Gastzugang von Benutzern, die kein Login auf diesem Rechner haben, kann dieser sich in einem begrenzten Teil des Dateisystems mittels ftp-Kommandos bewegen und auch Dateien transferieren (Archive mit freier Software).

ftp ftp.rz.uni-leipzig.de

login: **anonymous** oder auch **ftp**

Password: *e-mail-Adresse* (!!! Niemals das Paßwort eingeben !!!)

ftp > **help**

Kommandoübersicht

Dieser Zugang wird häufig für den Austausch von Public-Domain-Software oder -Daten genutzt. Hierzu gibt es weltweit sogenannte **anonymous ftp-Server**, auf denen verschiedenste Daten abgerufen werden können (hier: **ftp.rz.uni-leipzig.de**)

Internet- Browser:

ftp://ftp.rz.uni-leipzig.de

Login und Passwort werden vom Browser übermittelt, Daten lassen sich über Browserfunktionen auf den lokalen Rechner speichern.

Archivieren

Beim Datentransfer großer Datensätze ist es günstiger, die Dateien vor dem Versenden zu komprimieren. Man stellt ein Archiv zusammen, welches später wieder entarchiviert werden kann.

tar	tap archiver	Dateien archivieren und Archive bearbeiten
------------	---------------------	---

Archivieren:

tar cvf TvH.tar *

Das Archiv wird in der Datei **TvH.tar** abgelegt.

Archive lesen:

tar tvf TvH.tar

Der Inhalt des gesamten Archivs wird ausgegeben.

Entarchivieren:

tar xvf TvH.tar TvH.c

Liest die Datei **TvH.c** aus dem Archiv **TvH.tar** in das aktuelle Verzeichnis ein.

tar xvf TvH.tar

Liest das gesamte Archiv in das aktuelle Verzeichnis ein.

Option	Beschreibung
c	Archivieren, Datensicherung
x	Entarchivieren
t	Inhaltsverzeichnis eines Archivs wird ausgegeben
v	Übertragungsprotokoll auf der Standardausgabe
f name.tar	Archiv <i>name.tar</i> befindet sich im aktuellen Verzeichnis
f -	Archiv befindet sich auf der Standardausgabe

(cd /usr/mueller ; tar cf - .) | (cd /usr/muellerHans ; tar xf -)

Das Dateiverzeichnis **/usr/mueller** wird nach **/usr/muellerHans** kopiert.

Dialogtransfer

Um in einer interaktive Sitzung auf einen fernen Unix-Rechner über Internet zuzugreifen, verwendet man auf dem lokalen Rechner eines der folgenden Kommandos:

telnet		Dialogtransfer
---------------	--	-----------------------

telnet *userv1.informatik.uni-leipzig.de*

Nach dem Kommando **telnet** kann der Name oder die Adresse des fernen Rechners folgen, an dem man ein Login hat und arbeiten möchte.

Mit **telnet** werden Verbindungen zwischen zwei Unix-Maschinen oder einer nicht Unix- und einer Unix-Maschine aufgebaut. Jedes Zeichen, einschließlich Absender und Adresse, wird in ein Datenpaket verpackt. Für weite Verbindungen kann auf Zeilenmodus umgeschaltet werden.

Befehl	Beschreibung
open <i>Hostname</i>	Öffnen einer Verbindung
quit	Beenden der Verbindung
help	Kommandoübersicht
status	Anzeigen der Rechnerverbindung
mode line	Zeilenmodus
mode character	Zeichenmodus

rlogin	remote login	Dialogtransfer
---------------	---------------------	-----------------------

rlogin *userv1.informatik.uni-leipzig.de*

Nach dem Kommando **rlogin** folgt der Name oder die Adresse des fernen Rechners, an dem man ein Login hat und arbeiten möchte.

rlogin funktioniert nur zwischen Unix-Maschinen. Das Kommando verschickt jedes Zeichen einzeln in einem Paket und ist deshalb für entfernte Verbindungen nicht geeignet.

Befehl	Beschreibung
open <i>Hostname</i>	Öffnen einer Verbindung
exit	Dialog beenden
help	Kommandoübersicht

Auf dem entfernten Rechner kann man mit diesen Kommandos zunächst im Textmodus zugreifen. Möchte man im **Unix-Unix-Dialog** eine X-Anwendung starten, um von einem fernen Rechner auf einen nahen Rechner z.B. eine Grafik zu exportieren, so sind die folgende Schritte auszuführen:

1. Der lokale Rechner muß dem entfernten Rechner explizit erlauben, Fenster auf dem lokalen Bildschirm zu öffnen:
lokale Rechner > **xhost *userv1.informatik.uni-leipzig.de*** (ferner Rechner)
2. Dem fernen Rechner wird mitgeteilt, wo die X-Anwendung dargestellt werden soll (abhängig von der jeweilig verwendeten Shell, z. B. bash-Shell und ksh-Shell):
ferner Rechner > **export DISPLAY=*hppool01.rz.uni-leipzig.de*:0**
(lokaler Rechner, 0 ... Maschine hängt nur an einem Bildschirm und dieser wird exportiert.)
3. Starten der X-Anwendung auf dem entfernten Rechner.