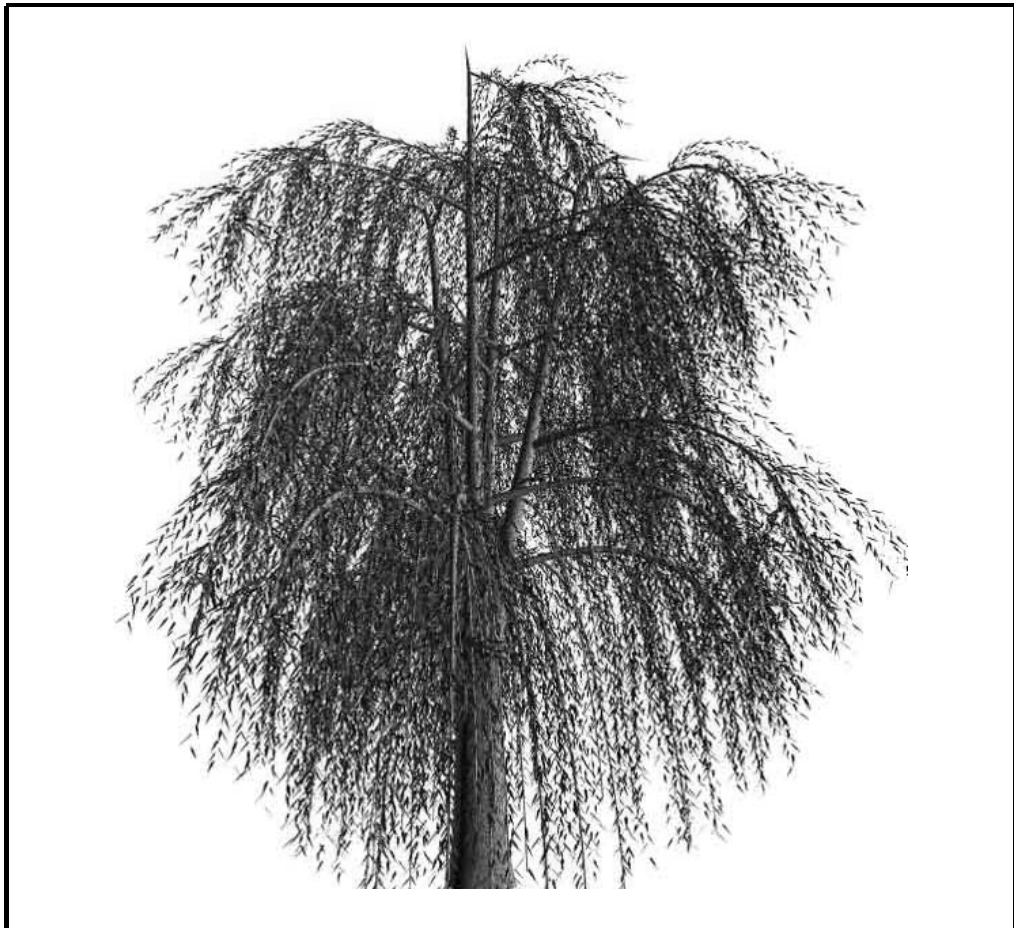




# NORSIGD INFO

Nummer 2 1996



NORSK SAMARBEID INNEN GRAFISK DATABEHANDLING

ISSN 0803-8317

## Aktivitetskalender

*Hva skjer når og hvor?*

---

<b>Oktober 1996</b>	
7-8	<b>IMPLICIT SURFACES '96: The Second Eurographics Workshop on Implicit Surfaces</b> , Eindhoven, Netherlands. wsinkvo@win.tue.nl.
14-18	<b>Theory and Practice of Geometric Modelling</b> , Blaubeuren, Tyskland. blaubeurenII@uni-tuebingen.de.
28-29	<b>Symposium on Volume Visualization</b> , San Francisco, CA, USA. <a href="http://www.erc.msstate.edu/conferences/vis96/volviz/volviz.html">http://www.erc.msstate.edu/conferences/vis96/volviz/volviz.html</a> .
<b>November 1996</b>	
6-8	<b>UIST'96, 9th Annual Symposium on User Interface Software and Technology</b> , Seattle, WA, USA. <a href="http://www.acm.org/uist/">http://www.acm.org/uist/</a> .
13-15	<b>Discrete Geometry For Computer Imagery</b> , Ecole Normale Supérieure de Lyon, France. <a href="http://www.ens-lyon.fr/LIP/DGCI96">http://www.ens-lyon.fr/LIP/DGCI96</a> .
14	<b>Grafikkformater og utveksling av grafiske data</b> , Veritassenteret Høvik, Norge. rre@dnv.no.
18-19	<b>3D-Bildanalyse und -synthese</b> , Workshop, Erlangen, Tyskland. mueller@ls7.informatik.uni-dortmund.de.
<b>Desember 1996</b>	
15-19	<b>COMPUGRAPHICS'96 5th International Conference on Computational Graphics and Visualization Techniques</b> , Paris, Frankrike. chpsanto@beta.ist.utl.pt
<b>Februar 1997</b>	
10-14	<b>The Fifth International Conference in Central Europe on Computer Graphics and Visualization'97</b> , Würzburg, Tyskland. cg97@informatik.uni-wuerzburg.de.
<b>Mars 1997</b>	
20-21	<b>CG'97 - 13th European Workshop on Computational Geometry</b> , Plzen - Bory, Tsjekkia. <a href="http://yoyo.zcu.cz/~skala/wscg97.html">http://yoyo.zcu.cz/~skala/wscg97.html</a> .
<b>Juni 1997</b>	
9-11	<b>10th Scandinavian Conference on Image Analysis</b> , Lappeenranta, Finland.
<b>August 1997</b>	
3-8	<b>SIGGRAPH 97: The 24rd International Conference on Computer Graphics and Interactive Techniques</b> , Los Angeles, California., <a href="http://www.siggraph.org/s97/">http://www.siggraph.org/s97/</a> .
<b>September 1997</b>	
4-8	<b>Eurographics'97</b> , Budapest, Ungarn. <a href="http://www.sztaki.hu/conferences/eg97/">http://www.sztaki.hu/conferences/eg97/</a>

---



### Om forsiden

Forsiden viser et piletre (©Bernd Lintermann) og er generert av **xfrog** som er ett interaktivt (*shareware*) modellerings program for planter o.l. **xfrog** kombinerer regel basert model generering med andre teknikker slik som *free form deformation* og *global and local constraints*. Dette muliggjør rask og fleksibel generering av planter.

**xfrog** er utviklet av *Bernd Lintermann* ved *Zentrum für Kunst und Medientechnologie Karlsruhe*.

Vil du vite mere, ta en titt på <http://www.greenworks.de/>.

## Hilsen fra styret

Høvik, september 1996

Kjære medlemmer,

Tema for denne utgaven av NORSIGD Info er JAVA. NORSIGD's fagansvarlig, Wolfgang Leister, har oversatt en tysk artikkel som gir en innføring i bruken av grafikk i Java. Dessuten har han søkt etter Java på nettet og gir sine kommentater til dette.

Styret planlegger et seminar den 14. november som skal handle om *Grafikkformater og utveksling av grafisk data*. Egen innbydelse vil bli sendt ut senere, da noen foredragsholdere ennå ikke har bekreftet deltagelsen.

Har du ønsker om temaer som du vil at vi skal skrive om i NORSIGD info eller arrangere seminar om, ta kontakt med styret. Oversikt over styret finner du på siste omslagsside.

Hilsen,

Reidar Rekdal



### NORSIGD Info

– medlemsblad for NORSIGD

Utgitt av: NORSIGD  
 Ansvarlig: Wolfgang Leister  
 Metronor AS  
 Postboks 238  
 1360 Nesbru

ISSN: 0803-8317

Utgivelser: 1996: 1/6 1/9 1/12

Annonsepriser: Helseid kr 5 000  
 Halvsid kr 2 500

Layout: Reidar Rekdal  
 L<sup>A</sup>T<sub>E</sub>X<sub>2</sub> $\epsilon$

Ettertrykk tillatt med kildeangivelse

## Innhold

Aktivitetsskalender .....	2
Internetthjørnet .....	4
Grafikk og Java .....	5
GPGS-F-hjørnet .....	10
Vil du vite mere on GPGS? .....	10

## Internethjørnet

Wolfgang Leister, Metronor AS

---

**Vi fortsetter med aktuelle temaer på internett. Det meste i denne utgaven handler om verdenen rundt Java. Et annet tema er standarder og utveksling av data.**

Java er et aktuelt tema på internett som i dataverdenen generelt. Opprinnelig ble Java utviklet hos SUN som et objektorientert programmeringsspråk ikke ulikt C++ for å styre all slags apparater fra fjernkontroller til mobiltelefoner. Men dette slo feil i første omgang, da ingen ville ha dette. Hovedpersonene under utviklingen var Bill Joy and James Gosling.

Senere ble idéen tatt opp igjen da man forsøkte å få mer dynamikk i det ellers så statiske og hypertext-baserte world wide web. Java med sin interpretative kode ga en plattform som kunne kjøres på forskjellige maskinarkitekturer i forbindelse med egnete web-browsere. Dermed er det nå mulig å skrive programmer som kan lastes ned fra nettet for kjøring på den lokale maskinen. I fremtiden vil det være mulig å få større applikasjoner levert på denne måten: tekstbehandlere, regneark, skatteprogrammer og spill.

På nettet finnes det mye informasjon om Java. Som inngangsport inn i Java-riket anbefales det <http://www.java.no>. Selv om endel sider kunne trenge en aktualisering er dette en grei måte å bli kjent med Java. Det finnes referanser til dokumentasjon (white paper), språkdefinisjon og et kurs! Et rikholdig bibliotek med eksempler og kildekode samt muligheten å laste ned kompilatorer og runtimebibliotek runder av tilbudet.

En annen kilde av informasjon er <http://java.sun.com>, hvor det også finnes informasjon om lisensiering og produkter. Derimot var et besøk på news-gruppen *comp.lang.java* heller skuffende. Etter en stund fant jeg ut at gruppen ble delt inn i åtte forskjellige grupper.

Er du interessert i eksempler som du kan kjøre på din maskin lønner det seg å ta et besøk på <http://home.sn.no/~espeset/>. Mange eksempler venter der.



Java programmer kan skrives på to forskjellige måter: som applikasjon og som applet. Forskjellen mellom disse to er at applikasjoner er frittstående programmer som kjøres i runtime-

omgivelsen. Java applets derimot brukes innenfor en Web browser.

Java programmer kompiles til en portabel kode, som interpreteres på den utførende datamaskinen. Denne koden er den samme for alle arkitekturer. Spesifikasjonen til denne abstrakte maskinen finnes på <http://www.java.no/docs/vmspec/vmspec-1.html>. Java programmer kan kjøres på veldig mange ulike arkitekturer og operativsystemer, da bare interpreteren er nødvendig.

Det finnes mange pakker for Java. Grafikk, brukersnitt, nettverksaksess mm finnes i egne biblioteker. En oversikt finnes på <http://www.java.no/docs/api/packages.html>.

På grunn av manglende operativsystemstøtte kan Java ikke kjøres på Windows 3.x. I disse tilfellene kan **JavaScript** brukes. JavaScript er en lett-versjon av Java, som programmeres inline i html-sider. Informasjon om JavaScript finner du på <http://depinfo.u-bourgogne.fr/bruno/java/script.html>.



Problemer med utveksling av data møter vi iblant når vår yndlingstekstbehandler ikke kan lese det viktige dokumentet som vi fikk på diskett, eller et illustrasjonsbilde ikke kan klippes og limes inn i en tekst uten konverteringer. Her kan bruken av standarder hjelpe til.

Det finnes mange forskjellige standarder for forskjellige datatyper og protokoller. Direktoratet XIII i EU organisasjonen tilrettelagt en oversikt som finnes på <http://www2.echo.lu/oi/en/index.html>. Her kan du velge blant mange forskjellige temaer, som også inbefatter endel grafikk.

Men også multi medier, lyd, tekstbehandlere, internettprotokoller, ... finnes her. For ISO standardene finnes det endel linker og henvisninger, mens standardene ikke finnes på nettet, da disse må bestilles fra standardiseringsorganisasjonene.

## Grafikk og Java

*Clemens Knörzer, Universität Karlsruhe*<sup>1</sup>

Java er et objektorientert programmeringsspråk, som på mange måter ligner C++. Utover dette introduserer Java også en ny filosofi om runtime-omgivelsen: Java programmer kompiles ikke til systemavhengig maskinkode, men til koden for en virtuell maskin. Derfor kan kompilerte Java programmer kjøres uforandret på mange arkitekturer ved hjelp av en interpreter. Med dette konseptet kan programutvikling skje for mange arkitekturer uten å måtte tenke på de mange forskjellige egenhetene for operativsystemene og prosessorene. Dette gir også den muligheten å kunne tilby programvare i det heterogene internet. Vi tar en nærmere titt på mulighetene for grafikk og brukerinteraksjon i Java.



Det finnes to typer av Java-programmer: Java applikasjoner og Java applets. **Java applikasjoner** er selvstendige programmer (innenfor Java runtime omgivelsen; noen nyere operativsystemversjoner kjører applikasjonene også direkte.). Derimot kan **Java applets** bare kjøres i omgivelsen til et annet (Java-)program, som har kontroll over appleten. Slike applets kan også lastes ned dynamisk, f.eks. fra internet.

### AWT pakken

Java tilbyr som standard over 200 predefinerte objekt-klasser som er delt inn i åtte pakker. For å lage grafiske brukersnitt brukes det *Abstract Windowing Toolkit* i pakkene *java.awt*, *java.awt.image* og *java.awt.peer*. I denne artikkelen skal vi i første linje snakke om pakken *java.awt*. Denne pakken omfatter totalt 44 objekt klasser (se bilde 1 og 2), hvorav de fleste klasser brukes for å definere dialoger og layout.

De fleste dialogkomponenter som brukes for å lage et grafisk brukersnitt finnes i pakken. Dialogkomponenter er hierarkisk organisert iht plassering av vinduene.

**Knapper:** Med klassen *button* lages det knapper, hvor aksjoner kan gjennomføres ved å trykke på knappen.

**Tekst:** Objekter av klassen *Label* viser tekst, som ikke kan forandres. Editerbar tekst fåes i klassene *TextField* (tekst som består av en linje) og *TextArea* (tekst som består av opp til flere linjer).

**Utvalg fra liste:** Klassen *List* består av flere linjer, og det kan enten velges nøyaktig ett ele-

ment (1:n) eller flere elementer (m:n) avhengig av moden. Det finnes også muligheten å vise en linje, og listen for å velge et element blir synlig som popup når man klikker på den.

**Utvalg:** Ved klikk på *toggle buttons* slås ett valg av eller på. I tillegg finnes det *radio buttons* hvor bare et valg i en liste kan være aktiv. Begge disse variantene tilhører klassen *checkbox*. For radio buttons må knappene være registrert i en felles *checkboxGroup*.

**Tegneflate:** Klassen *canvas* tilbyr en tom tegneflate.

**Meny:** Med klassene *MenuBar*, *Menu* og *CheckboxMenuItem* lages det enkle meny valg.

**Scrollbars:** Klassen *Scrollbar* benyttes for å implementere definere tallverdier grafisk.

### Beholdere

Beholdere brukes for å sette flere dialogobjekter sammen. Følgende klasser står til disposisjon:

**Rammer:** Klassen *Frame* implementerer en ramme rundt et vindu. Java applikasjonenes hovedvindu er et slikt rammeobjekt. Frames har inget foreldrevindu. De kan sammenlignes med *ApplicationShell* i X.

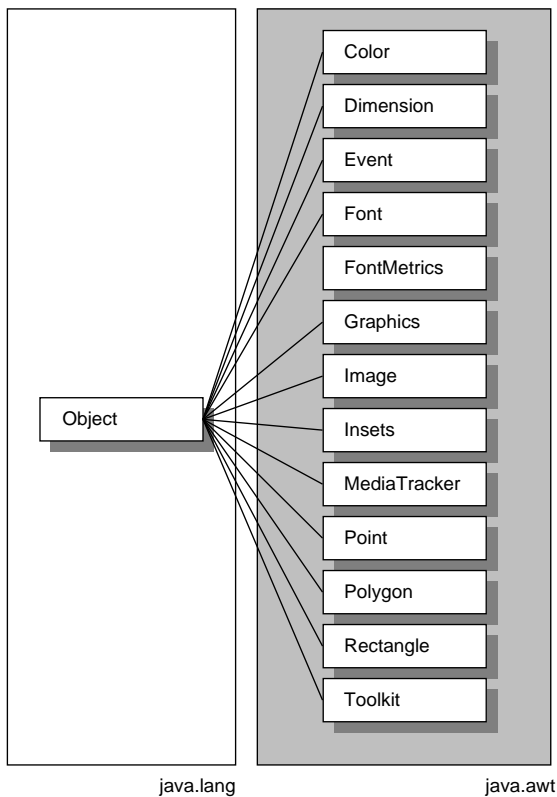
**Dialoger:** Dette er vinduer med rammer, men har ikke nødvendigvis en titellinje. Dialoger brukes som popup vinduer. For å implementere modale vinduer trenger de et foreldrevindu. Modale vinduer endrer sin status etter foreldrevinduet.

<sup>1</sup>Oversatt av Wolfgang Leister. Forfatteren kan nås på e-post <mailto:knoerzer@ira.uka.de>.

**FileDialog:** Spesiell dialog for å velge ut en fil.

**Window:** Vindu uten ramme, som bl a kan benyttes som små popup-vinduer. Window-objekter kan også bli større enn foreldrevinduet.

**Panel:** Et panel er en beholder for andre vinduer. Den eier ikke et eget (synlig) vindu. Dens oppgave er å plassere og gruppere andre dialogkomponenter. En Panel kan sammenlignes med en FormWidget i X.



**BorderLayout:** En beholder med *BorderLayout* kan ha opp til fem dialogkomponenter: en i midten og en på hver side.

**GridLayout:** Dialogkomponentene arrangeres i et fast raster. Hver av raster-cellene har lik bredde og høyde.

**GridBagLayout:** Dette er den mest omfattende layout manageren, der dialogkomponentene blir arrangert i en meget variabel tabell. Hver av kolonnene kan være av forskjellig bredde og hver linje kan ha forskjellig høyde. Komponentene kan innta flere kolonner og linjer, og innenfor en linje være sentrert eller justert til høyre eller venstre. På denne måten er det mulig å realisere valgfrie plasseringer.

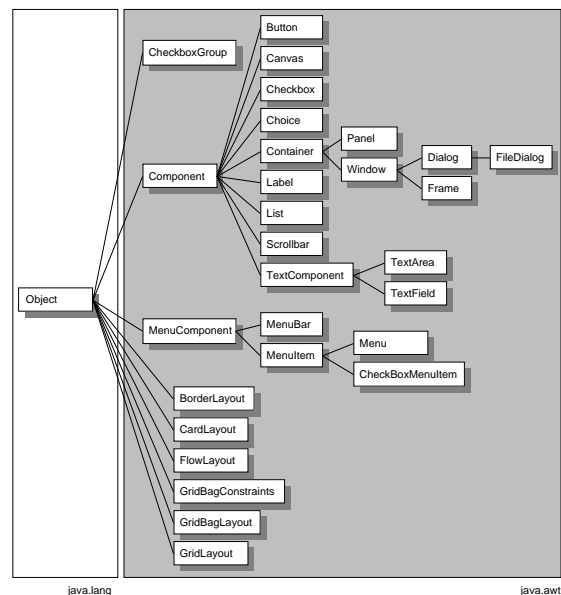
**CardLayout:** I denne litt uvanlige manageren blir de forskjellige komponentene arrangert opp på hverandre, slik at de dekker hverandre til. På denne måten kan man svitsje mellom dialogkomponenter.

**Egne managere:** Dersom disse mulighetene ikke skulle være tilstrekkelig er det mulig å programmere en layout manager selv.

## Dialogkomponenter

Dialogkomponenter eier ikke et eget vindu, mens de arrangerer avhengige dialogkomponenter. Dette styres av *layout manageren*, som tilordnes enhver beholder. AWT har fem forskjellige layout managere:

**FlowLayout:** Dialogkomponenter arrangeres som ord i et tekstbehandler: komponentene plasseres fra venstre til høyre ved siden av hverandre. Blir den høyre marginen nådd fortsettes plasseringen i neste linjen. Som i en tekstbehandler kan linjene justeres til høyre eller venstre side, eller sentrert.

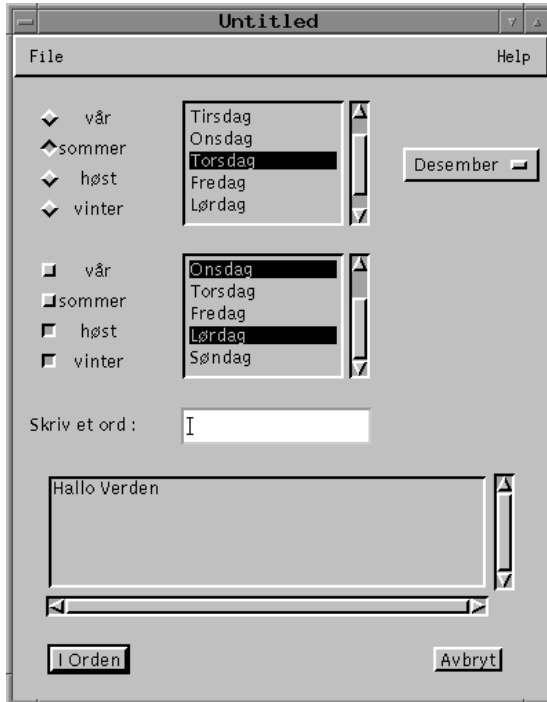


## Event-Handling

Ved en brukerinteraksjon på brukersnittet genereres det såkalte *events*. Det finnes flere typer events: Action-, Window-, Keyboard-, mus-,

scrolling-, list- og fokus-events. Alle komponenter har en metode *handleEvent()*, som tar seg av å bearbeide eventene.

Events blir sendt først til denne dialogkomponenten som har utløst dette og metoden *handleEvent()* blir kalt. Hvis den ikke kan bearbeides fra denne komponenten, så sendes meldingen videre til den overordnede beholderen. Det anbefales å bearbeide events så nær den utløsende dialogkomponenten som mulig for å unngå store *switch-case-strukturer* ved å gjenkjenne events.



I Java er det enkelt å lage programmer med

flere tråder (multi-threaded), i motsetning til å programmere et slikt brukersnitt i X/Motif.

## Enkle grafikk operasjoner

For å trylle frem grafikk og diagrammer på skjermen, brukes det primitive grafikk-operasjoner som sirkel, firkant, linje, osv. Disse er implementert i en klasse *Graphics*. En instans av denne klassen fåes ved en metode *getGraphics()* som defineres for alle dialogkomponenter. Klassen *Graphics* implementerer metoder som: *drawLine()*, *drawOval()*, *drawPolygon()*, *drawRoundRect()*, *drawString()*, *drawImage()*, *fillArc*, osv. Desverre finnes det bare linjer som er 1 piksel bred, og fyllfunksjoner har ingen mønster eller skravering. Da dette ble bemerket av mange Java-programmerere, vil dette sikkert bli endret i en av de neste Java versjonene.

## Hastighet

Det sies at Java programmer er omtrent ti til førti ganger så sakte som vanlig kompilerte programmer. Dette stemmer vel for veldig beregningsintensive programmer, men er ikke så relevant for et grafisk brukersnitt. Ved oppbygging av dialog-vinduer finnes det ingen synlige forskjeller, som ville være forstyrrende.

For å vise hastigheten for disse operasjonene, finnes det en side på internett som implementerer grafikk-operasjoner. Ved å rette en Java-Web-Browser på <http://i31www.ira.uka.de/~jsteck/> blir et grafikk-felt synlig, som brukes for å gjennomføre en benchmark på den aktuelle datamaskinen.

## Et eksempel

For å vise litt av mulighetene i Java presenterer vi et lite eksempel. Programmet viser et smilende / ikke smilende ansikt avhengig av hvordan knappen *happy* er satt. Det har blitt lagt vekt til å vise to muligheter for å bearbeide events: med en action-routine i dialog- eller programvinduet (som vist for checkbox-event) eller i dialogobjektet (som i *QuitButton*). For å oppnå mest objektorienterte programmer anbefales det å koble rutinen til objektet og lage nye klasser for alle vinduer.

```
import java.applet.*;
import java.awt.*;
import java.lang.*;

public class smile extends Applet {
    Dimension csize;
    SmilyCanvas sm;
    Checkbox happy;
```

```
public smile(){ // Konstruktor
    // Applet arver fra Panel
    setLayout(new BorderLayout());
    csize = new Dimension(100,100);
}

public smile(int w, int h){
    this();
    resize(w,h);
    csize = new Dimension(w,h);
}

public void init(){
    Panel panel;
    sm = new SmilyCanvas(csize);
    sm.setState(true);
    panel = new Panel();
    panel.setLayout(new GridLayout(1,0));
    happy = new Checkbox("glad");
    happy.setState(true);
    panel.add (happy);
    add ("Center", sm);
    add ("East", panel);
}

// Ved alle hendelser kalles metoden handleEvent(Event).
// Dersom denne ikke blir overskrevet kalles action(Event,Object).
// En slik hendelse finnes for Button, Checkbox, MenuItem, TextField, ...
public boolean action(Event evt, Object what){
    if (evt.target == happy)
        sm.setState(evt.arg.toString().equals("true"));
    return true;
}
// Event er ikke behandlet. Send videre til overordnet Panel / Window
return false;

// main brukes bare for Standalone programmer.
public static void main(String args[]){
    Applet a = new smile(100,100);
    Frame f = new Frame("Smily");
    f.add("Center", a);
    f.add("South", new QuitButton("God morgen da!"));
    a.init();
    f.pack();
    f.show();
    a.start();
}
} // end smile

// ===== SmilyCanvas =====
class SmilyCanvas extends Canvas {
    Dimension prefSize;
    boolean state;

    public SmilyCanvas(Dimension size){ // Konstruktor
        setForeground(Color.blue);
        setBackground(Color.yellow);
        prefSize = size;
        state = false;
    }
}
```



```

    }

    public Dimension preferredSize() { return prefSize; }
    public Dimension minimumSize() { return prefSize; }
    public void setState(boolean st) {
        state = st;
        repaint();
    }

    // brukes for redraw.
    public void paint(Graphics g){
        int c_x, c_y, diam;
        int eyes_x, eyes_y, eyes_diam;
        int mouth_diam;

        c_x = Math.round(size().width/2);
        c_y = Math.round(size().height/2);
        diam = (int)Math.round(Math.min(c_x,c_y)*0.8);
        eyes_x = (int)Math.round(diam*0.4);
        eyes_y = (int)Math.round(diam*0.4);
        eyes_diam = (int)Math.round(diam*0.1);
        mouth_diam = (int)Math.round(diam*0.5);

        g.fillOval(c_x-diam, c_y-diam, 2*diam, 2*diam);
        g.setColor(Color.yellow);
        g.fillOval(c_x-eyes_x-eyes_diam, c_y-eyes_y-eyes_diam,
            2*eyes_diam, 2*eyes_diam);
        g.fillOval(c_x+eyes_x-eyes_diam, c_y-eyes_y-eyes_diam,
            2*eyes_diam, 2*eyes_diam);
        if (state){
            g.drawArc(c_x-mouth_diam, c_y-mouth_diam,
                2*mouth_diam, 2*mouth_diam, 190, 160);
        } else {
            g.drawArc(c_x-mouth_diam, c_y,
                2*mouth_diam, 2*mouth_diam, 10, 160);
        }
    }
} // end SmilyCanvas

// ===== QuitButton =====
class QuitButton extends Button {
    public QuitButton (String label) { super(label); }

    // avslutt en applikasjon
    public boolean action(Event evt, Object what) {
        System.exit(0);
        return true;
    }
} // end QuitButton

```

## GPGS hjørnet

Magnar Granhaug, Sintef Tele og Data  
Reidar Rekdal, DNV SESAM

---

**Pattern i Windows versjonen:** En begrenset implementasjon av pattern/hatch er nå tilgjengelig i Windows versjonen (august 96). Implementasjonen begrenser seg foreløpig til SOPOL av type 2.

**Backingstore i Windows versjonen:** En form for *backingstore* ved hjelp av *memory bitmaps* er implementert i Windows versjonen sin skjermdriver (august 96). Grafikk som har vært skjult av andre vinduer eller minimalisert vil automatisk bli tegnet om når vinduet blir synlig igjen. Man kan velge å slå av "backingstorefunksjonaliteten.

**Ny funksjon i X-driveren:** En ny funksjon i X-driveren, GPGSdisplay, gjør det mulig å sende X display pekeren fra et X program til GPGS. Dette er nødvendig hvis både X applikasjonen og GPGS skal kunne håndtere input fra X vinduet.

**Automatisk bufferallokering:** Det er nå (september 96) mulig å overlate bufferallokeringen til GPGS. Fordelen er at GPGS allokere plass etterhvert som den trenger mere plass. Brukeren kan styre allokeringseenhet og maksimum plass som skal allokere. Utvidelsen er eksternt finansiert.

**Utvidet funksjonalitet ved pick input:** Ved bruk av rutinene REQHIT og SMPHIT vil GPGS-F søke gjennom segmentene i prioritert

rekkefølge, og returnere informasjon om første element som ligger innenfor det definerte treffområde.

Det har tidligere ikke vært mulig å få informasjon om det var flere elementer som ble truffet, noe som ofte vil være tilfelle ved litt komplekse bilder.

Denne muligheten har nå blitt lagt inn i GPGS-F, ved at det er definert en ny rutine NXTHIT. Denne rutinen kan kalles en eller flere ganger etter REQHIT/SMPHIT, og vil fortsette søkingen fra forrige treff.

Størrelsen på treff-området som brukes ved pick input kan defineres av applikasjonen, men fasongen har tidligere vært begrenset til å være et rektangel. En ny rutine er nå implementert for å definere treff-området som et vilkårlig polygon. I tillegg kan applikasjonen velge om treff skal returneres kun for elementer som ligger helt innenfor det definerte polygonet, eller som tidligere, også for elementer som ligger bare delvis innenfor.

GPGS-F har manglet muligheter til å gi en visuell tilbakemelding om hva som ble truffet ved pick input. Dette er nå mulig ved at det er definert en rutine HGHPIC som tegner et angitt segment i en spesiell highlight-modus. Denne modusen er definert slik at hvis segmentet tegnes en gang til, skal man få tilbake det opprinnelige bildet. Det er opp til den enkelte driver hvordan denne highlight-modusen skal implementeres, i X11 driveren brukes XOR modus.

---

## Vil du vite mer om GPGS?

Ønsker du å vite mere om grafikkpakken GPGS, ta kontakt med NORSIGDs sekretær eller en av de versjonsansvarlige:

**Unix og VMS :**  
Sintef Tele og Data  
7034 Trondheim  
v/ Magnar Granhaug  
Telefon: 73 59 29 63 (direkte)

**MS-DOS og Windows :**  
ViaNova AS  
Postboks 53  
1312 Slependen  
v/Stein Slaatsveen  
Telefon: 67 56 46 00 (sentralbord)

## Hva er NORSIGD?

**NORSIGD** – Norsk samarbeid innen grafisk databehandling – ble stiftet 10. januar 1974. NORSIGD er en ikke-kommersiell forening med formål å *fremme bruken av, øke interessen for, og øke kunnskapen om grafisk databehandling i Norge.*

Foreningen er åpen for alle enkeltpersoner, bedrifter og institusjoner som har interesse for grafisk datbehandling. NORSIGD har per januar 1996 65 institusjons- og 27 personlige medlemmer. Medlemskontingenten er 1.000 kr per år for institusjoner. Institusjonsmedlemmene er stemmeberettiget på foreningens årsmøte, og kan derigjennom påvirke bruken av foreningens midler.

Personlig medlemskap koster 250 kr per år. Personlige medlemmer får tilsendt medlemsbladet *NORSIGD Info*, og er berettiget til redusert kontingent ved medlemskap i vår europeiske samarbeidsorganisasjon *Eurographics*.

Alle medlemmer får tilsendt medlemsbladet *NORSIGD Info* 3-4 ganger per år.

### Interesseområder

NORSIGD er et forum for alle som er opptatt av grafiske brukergrensesnitt, uavhengig av om basisen er *The X window System*, *Microsoft Windows* eller andre systemer. NORSIGD arrangerer møter og seminarer, formidler informasjon fra internasjonale fora og distribuerer fritt tilgjengelig programvare. I tillegg formidles kontakt mellom brukere og kommersielle programvareleverandører.

NORSIGD har lang tradisjon for å støtte opp om bruk av datagrafikk. Foreningen bidrar til spredning av informasjon ved å arrangere møter, seminarer og kurs for brukere og systemutviklere.

### GPGS

GPGS er en 2D- og 3D grafisk subrutinepakke. GPGS er maskin- og utstyrsuavhengig. Det vil si at et program

utviklet for én maskin med f.eks. bruk av plotter, kan flyttes til en annen maskin hvor plotteren er erstattet av en grafisk skjerm uten endringer i de grafiske rutinekalene. Det er definert grensesnitt for bruk av GPGS fra FORTRAN og C.

Det finnes versjoner av GPGS for en rekke forskjellige datamaskiner, fra stormaskiner til Unix arbeidsstasjoner og PC. GPGS har drivere for over 50 forskjellige typer utsyr (plottere, skjermer o.l.). Data kan utveksles mot CGM metafil.

GPGS eies av NORSIGD, og leies ut til foreningens medlemmer.

### Eurographics

Eurographics ble grunnlagt i 1981 og har medlemmer over hele verden. Organisasjonen utgir et av verdens fremste fagtidsskrifter innen grafisk databehandling, *Computer Graphics Forum*. *Forum* sendes medlemmene annen hver måned. Eurographics konferansen arrangeres årlig med seminarer, utstilling, kurs og arbeidgrupper.

NORSIGD samarbeider med Eurographics. Personlige medlemmer i NORSIGD får 20 SFr rabatt på medlemskap i Eurographics, og vi formidler informasjon om aktuelle aktiviteter og arrangementer som avholdes i Eurographics-regi. Tilsvarende får Eurographics medlemmer kr 100 i rabatt på medlemskap i NORSIGD.

Institusjonsmedlemskap i Eurographics er priset avhengig av antall kontaktpersoner i institusjonen. Institusjonsmedlemmer får flere eksemplarer av *Forum*, og de får rabatt ved deltagelse på EG-konferanse. Eurographics kan bistå ved å holde kurs og seminarer hos institusjonsmedlemmer. Institusjonsmedlemmer kan også presentere seg i *Forum*, og de får tilgang til Eurographics' adresseliste. Det er en egen stand for institusjonsmedlemmene på EG-konferansene. Hver kontaktperson har ellers de samme rettigheter som personlige Eurographics medlemmer.

NORSIGD  
v/ Reidar Rekdal  
DNV SESAM  
Postboks 300  
1322 HØVIK

**Returadresse:**  
 NORSIGD v/ Reidar Rekdal  
 DNV SESAM AS  
 Postboks 300  
 1322 HØVIK

## Styret i NORSIGD 1996

Funksjon	Adresse	Telefon	email
Leder	Ketil Aamnes ViewTech AS c/o SINTEF 7034 TRONDHEIM	73 59 70 54 (direkte) 73 59 29 71 (fax)	Ketil.Aamnes @viewtech.no
Fagansvarlig	Wolfgang Leister Metronor AS Postboks 238 1360 NESBRU	66 98 38 39 (direkte) 66 98 38 00 (sentralbord) 66 98 38 01 (fax)	leister@sn.no
Sekretær	Reidar Rekdal Det Norske Veritas Sesam AS Postboks 300 1322 HØVIK	67 57 73 18 (direkte) 67 57 72 50 (sentralbord) 67 57 72 72 (fax)	rre@dnv.no
Styremedlem	Gisle Fiksdal MARINTEK A.S Postboks 4125, Valentinlyst 7002 TRONDHEIM	73 59 59 07 (direkte) 73 59 57 76 (fax)	Gisle.Fiksdal @marintek.sintef.no
Varamedlem	Svein Taksdal Norges Vassdrags- og Energiselskap Hydrologisk Avdeling, Seksjon data Postboks 5091, Majorstua 0301 OSLO	22 95 92 86 (direkte) 22 95 92 01 (fax)	svein.taksdal @nve.no
Varamedlem	Rune Torkildsen Chr. Michelsen Research Postboks 3 5036 FANTOFT	55 57 43 54 (direkte) 55 57 40 40 (sentralbord) 55 57 40 41 (fax)	Rune.Torkildsen @cmr.no

<p><b>Svarkupong</b></p> <p><input type="radio"/> Innmelding – institusjonsmedlem</p> <p><input type="radio"/> Innmelding – personlig medlem</p> <p><input type="radio"/> Ny kontaktperson</p> <p><input type="radio"/> Eurographics innmelding – personlig</p> <p><input type="radio"/> Eurographics innmelding – institusjon</p> <p><input type="radio"/> Adresseforandring</p>	<p>Navn: .....</p> <p>Firma: .....</p> <p>Gateadresse: .....</p> <p>.....</p> <p>Postadresse: .....</p> <p>.....</p> <p>Postnummer/sted: .....</p> <p>.....</p> <p>Telefon: .....</p> <p>Telefaks: .....</p> <p>email: .....</p>
---	--