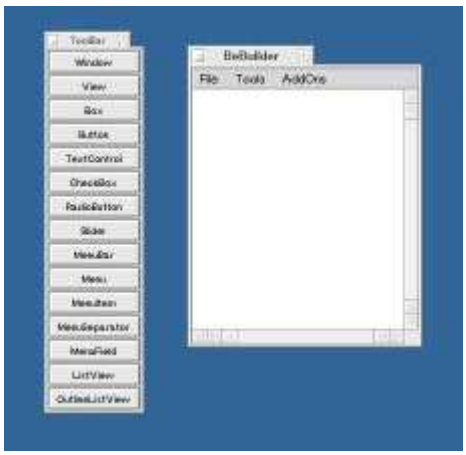
Grafikprogrammierung mit BeBuilder V0.5

Index

Mit Hilfe des BeBuilders lassen sich auf einfache Weise Fenster basteln. Für BeOS gibt es mittlerweile mehrere solcher Programme, da der BeBuilder jedoch kostenlos ist werde ich diesen in diesem Teil des Kurses einsetzten. Der BeBuilder kann unter folgender Adresse heruntergeladen werden:
http://www.bezip.de/app/224. Die Datei ist ca. 1 MB gross.

Nachdem der BeBuilder heruntergeladen und entpackt ist, läst er sich durch einen Doppelklick auf BeBuilder starten. Nach dem start siehst Du folgende Oberfläche.



In der Toolbar sind verschiede Komponenten enthalten die für die Fenstergestaltung verwendet

werden können. Im BeBuilder-Fenster werden später alle Komponenten aufgeführt die in dem geöffneten Projekt verwendet werden.

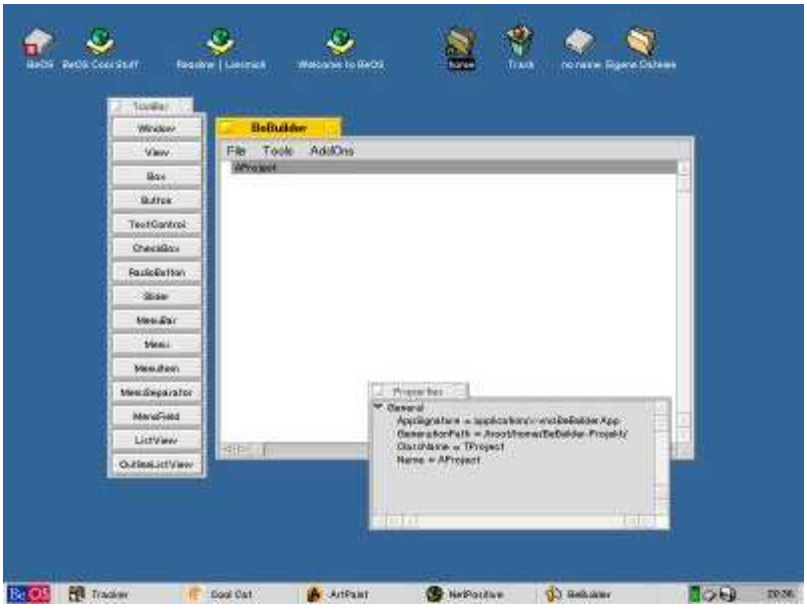
---

1. Das erste Programm

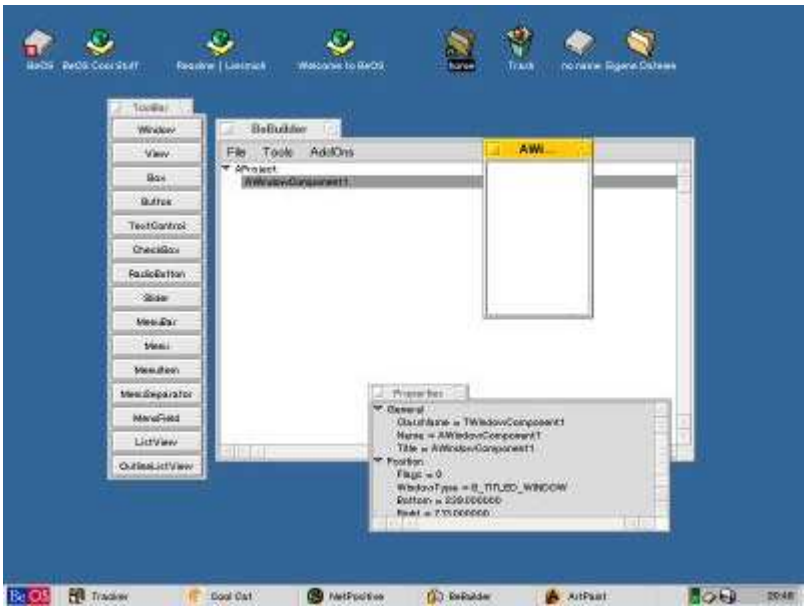
Ein Projekt erstellen

Um ein Projekt zu erstellen musst Du im BeBuilder-Fenster auf File und dann auf New Project klicken. Im folgenden Fenster kannst Du ein Verzeichnis wählen in dem das Projekt gespeichert werden soll, öffne es und klicke dann auf Select 'Verzeichnisname'.

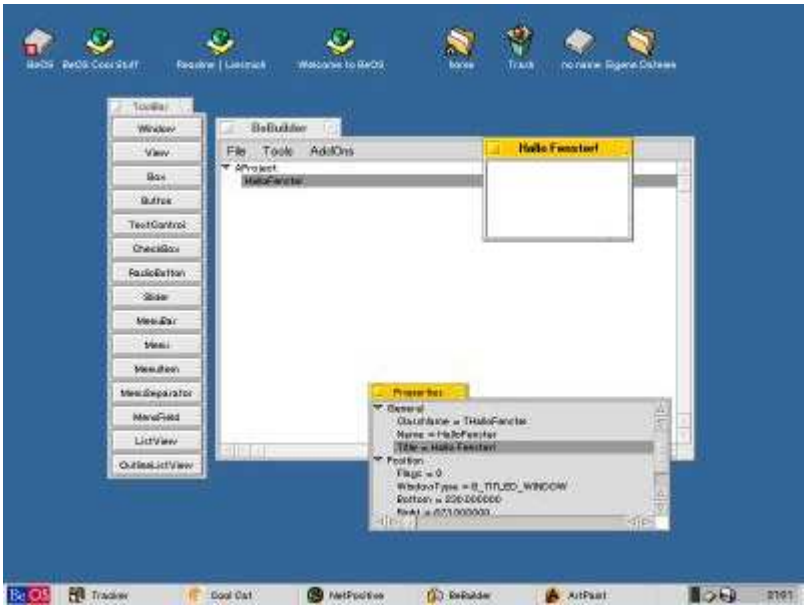
Im BeBuilder-Fenster steht jetzt AProject. Klick darauf um die Eigenschaften des Projektes anzusehen.



Zunächst wollen wir hier nichts ändern. Klicke jetzt in der Toolbar auf Window, danach erscheint im BeBuilder-Fenster ein Eintrag AWindowComponent1 und ein neues Fenster wird geöffnet (verteckt sich manchmal unter den anden). Klicke jetzt im BeBuilder-Fenster doppelt auf AWindowComponent1 um die Eigenschaften des Fensters anzuzeigen.



Klicke jetzt im Properties - Fenster (Eigenschaftsfenster) auf ClassName und ändere den Klassennamen im sich öffnenden Fenster auf THalloFenster. Klicke danach auf Validate. Klicke danach im Properties-Fenster auf Name und ändere den Namen auf HalloFenster. Nachdem Du den Namen geändert hast klickst Du auf Title um den Titel des Fensters in Hallo Fenster! zu ändern.



Das Fenster Deines Programms kannst Du übrigends genauso vergrössern, verkleinern oder verschieben wie jedes andere Fenster in BeOS.

Um das Dein Programm jetzt zu starten klicke im BeBuilder-Fenster auf den Menüpunkt Tools und anschliessend auf Make. Jetzt erscheint ein Meldefenster das fragt ob Du eine Datei öffen willst die keine Projekt oder Testdatei ist. Klicke hier auf Open und schliesse die Datei wieder. Jetzt kannst Du im Menüpunkt Tools auf Run klicken um das Programm zu starten. Eine ausführbare Datei wurde erzeugt, die im Unterverzeichnis obj.x86 Deines Projektverzeichnisses unter dem Namen BeBuilderApp zu finden ist.

Fertig!!! das war Dein erstes richtiges BeOS-Programm. Zugegeben es ist nichts besonderes und programmiert ist daran auch noch nichts. Aber es erklärt die Funktion von BeBuilder. Wir wollen das Programm jetzt Schritt für Schritt weiter ausbauen.

2. Das Programnm um einen Ende-Button erweitern

Was ist ein View?

Ein View ist ein Bereich innerhalb eines Fensters indem etwas angezeigt werden kann. Um einen Button zu erzeugen benötigt man zuächst einen solchen View damit er überhaupt angezeigt werden kann. Für unser Programm benötigen wir zunächst nur ein View.

Einen View hinzufügen

Um einen View hinzuzufügen klicke jetzt auf HalloFenster im BeBuilder - Fenster. Danach klickst Du doppelt auf View in der Toolbar. Das innere Deines Programmfensters wird grau und im BeBuilder-Fenster ist ein neuer Eintrag AViewComponent2 aufgetaucht.

Klicke jetzt doppelt auf AViewComponent2 und ändere im Properties-Fenster denn Klassennamen (ClassName) auf THalloAnzeige und den Namen (Name) auf HalloAnzeige.

Einen Button hinzufügen

Um einen Button hinzuzufügen muss jetzt HalloAnzeige im BeBuilder-Fenster makiert sein. Durch einen klick auf Button in der Toolbar wird jetzt ein Button hinzugefügt. In Deinen Programmfenster erscheint jetzt ein Button und im BeBuilder-Fenster taucht ein neuer Eintrag AButtonComponent3 auf. Klicke jetzt doppelt darauf um ins Properties-Fenster des Buttons zu kommen. Ändere die Message auf BUTTON_ENDE, das Label (das was nacher auf Deinem Button steht) auf Ende, den ClassName auf TEndeButton und den Namen auf EndeButton.

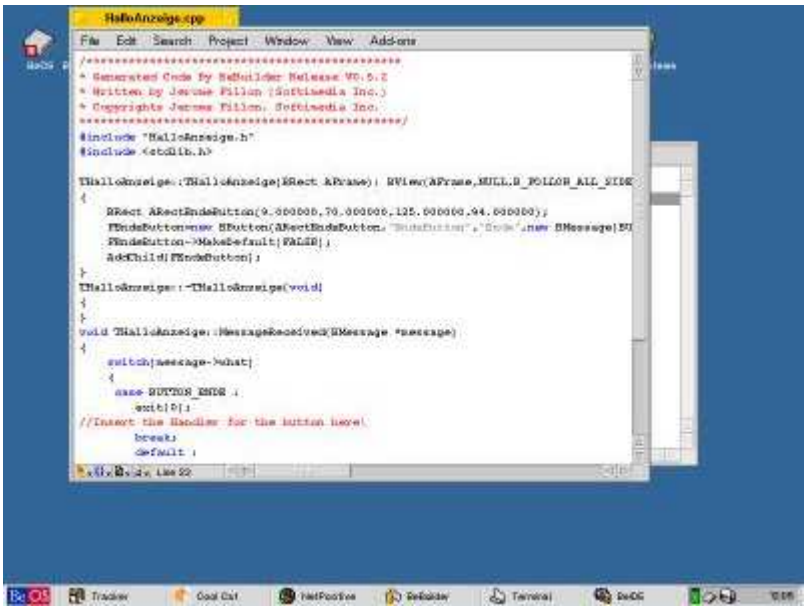
Was ist eine Message und wie arbeitet man damit?

Eine Message wird immer dann ausgelöst wenn der Benutzter irgendetwas macht, beispielsweise die Maus bewegt, eine Taste drückt oder unseren Button anklickt. Diese Message wird an alle Programme weitergegeben. Jedes Programm prüft nun ob die Message für sich ist, wenn nicht wird es an das nächste Programm weitergegeben. Hat die Message dann das zugehöhrige Programm gefunden wird die Reaktion darauf abgearbeitet. In unserem Beispiel heisst die Message BUTTON_ENDE.

Um mit der Message arbeiten zu können musst Du jetzt auf HalloAnzeige im BeBuilder-Fenster klicken. Danach gehst Du auf den Menüpunkt Tools und danach auf Edit CPP-File. Sollte sich hier nichts tun, so musst Du den Compiler noch eintragen. Gehe dazu auf File/Preference und stelle dort BeIDE ein. Jetzt gehst Du wieder auf Tools/Edit CPP-File.

Am Anfang der Datei musst Du die Headerdatei stdlib.h einbinden indem Du folgendes reinschreibst (Nach dem Komentarfeld):
#include <stdlib.h>
Ca. in der Mitte findest Du case BUTTON_ENDE: Schreib danach hinein:
exit(0);

Der Code bewirkt das das Fenster geschlossen wird sobald man auf den Button klickt. Speichere die cpp-Datei und schliesse sie wieder.



Du kannst das Programm jetzt ausführen indem Du auf Tools/Make und danach auf Tools/Run klickst.

---

3. Maus und Tastatur

Messages von Maus und Tastatur

Um eine Message von der Tastatur bearbeiten zu können benötigen wir eine neue Funktion, namens KeyDown. Diese muss der View-Klasse noch hinzugefügt werden. . In dieser Funktion können wir mit Hilfe einer switch-case-Anweisung auf die Tasten reagieren, die wir für unser Programm benötigen. Beispielsweise die ESC-Taste um das Programm zu beenden.

Um die Funktion KeyDown dem Projekt hinzufügen, klicke jetzt auf HalloAnzeige im BeBuilder-Fenster. Jetzt klicke im Menü auf Tools und anschliessend auf Edit Header File. Ergänze in der Klasse HalloAnzeige die Funktionen:
void KeyDown(const char *bytes, int32 numBytes); und
void MouseDown(BPoint point);
Die Funktion MouseDown wird dazu verwendet um den Focus ins Fenster des Programms zu setzten damit das Programm bei einem Tastendruck auch weiss das die ausgelöste Message bearbeitet werden muss. Speichere die Datei und schliesse sie. Klicke jetzt auf Tools und anschliessend auf Edit Cpp-File. Gehe zum Ende der Datei und schreibe dort:

```
void THalloAnzeige::KeyDown(const char *bytes, int32 numBytes)
{
switch( *bytes )
{
case B_ESCAPE:
case B_END:
case ' e' :
exit(0);
break;
default:
break;
}
```

```
}

 void THalloAnzeige::MouseDown(BPoint point)
{
MakeFocus();
}
```

Wer will kann sich das Beispielprogramm natürlich auch herunterladen -> Download (246 kb)

Speichere das ganze und schliesse die Datei wieder. Indem Du jetzt auf Tools/Make klickst wird das Programm Kompiliert. Du kannst es jetzt mit Make/Run ausführen. Um das Programm mit Hilfe der ENDE, ESC oder e-Taste zu beenden musst Du mit der Maus in das Fenster klicken. Anschliessend kannst Du das Programm durch die Tasten ESC, Ende oder ' e' b eenden.

Die Tastaturcodes für die Funktionstasten findest Du im BeBook besonders schnell wenn Du B_ESCAPE oder B_END markierst und anschliessend die Tasten SHIFT+ALT+D gleichzeitig drückst.

Um die Koordinaten des Cursors zu erhalten kannst Du in der Funktion MousDown auf die Variablen point.x und point.y zugreifen. Mehr dazu später!!!!

---

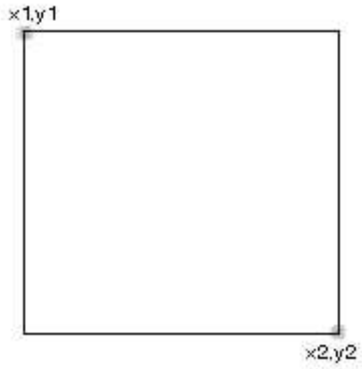
4. Zeichnen

Mit den Mauskoordinaten ein Rechteck zeichen

Wie oben bereits erwähnt erhält man die Mauskoordinaten mit Hilfe der Variablen point.x und point.y. Diese Koordinaten sollen nun dazu verwendet werden um ein Rechteck zu zeichnen. Öffne nun die Cpp-Datei indem Du auf HalloAnzeige markierst und auf Tools/Edit CPP-File klickst. Die Methode MouseDown muss wie folgt ergänzt werden:

```
void THalloAnzeige::MouseDown(BPoint point)
{       BRect rect;
        MakeFocus();
        rect.Set(point.x, point.y, point.x+10, point.y+10);
        StrokeRect(rect);
}
```

BRect ist eine Klasse die es erlaubt, Rechteckskoordinaten wie folgt zu speichern: (x1, y1, x2, y2)



Diese Werte werden mit rect.Set gesetzt. In diesem Beispiel werden dazu die Mauskoordinaten

point.x und point.y verwendet. Mit StrokeRect(rect) wird das Rechteck nun gezeichnet. Um das ganze zu testen, kannst Du die Datei nun wieder schliessen und das Programm compilieren. Anschliessend kannst Du es ausführen. Wenn Du jetzt irgendwo ins Fenster klickst wird dort ein Rechteck gezeichnet.

Das Rechteck füllen

Jetzt soll das Rechteck noch gefüllt werden. Dazu gibt es die Methode FillRect(BRect rect). Diese kannst Du einfach unter StrokeRect schreiben. Das ganze sieht dann so aus:

```
void THalloAnzeige::MouseDown(BPoint point)
{       BRect rect;
        MakeFocus();
        rect.Set(point.x, point.y, point.x+10, point.y+10);
        StrokeRect(rect);
        FillRect(rect);
}
```

Wenn Du jetzt das Programm compilierst und ausführst wird überall wo Du mit der Maus im Fenster klickst ein schwarzes Rechteck gezeichnet.

Einen Kreis zeichnen

Einen Kreis kannst Du genauso zeichnen wie ein Rechteck, Du musst nur statt StrokeRect(rect) StrokeEllipse(rect) schreiben. Um den Kreis zu füllen schreibst Du statt FillRect(rect) FillEllipse (rect) und schon hasst Du einen Kreis.

Wie fandest Du diesen Kurs bisher und was sollte ich anders machen?
Email