

# Menüs mit Bitmap-Button (Koord ab V1.12 Rev.1)

[Einleitung](#)

[Beispiel einer Bitmap Datei:](#)

[Beispiel einer Steuerungs Datei:](#)

[Beispiel einer SkinMenu1.txt Datei:](#)

[Bilder vom Beispiel Menü](#)

## Einleitung:

Mit Koord kann man auch Menüs mit Button erstellen.

Die Button müssen als BMP erstellt werden.

Am besten man legt sich einen Ordner an z.B. SkinMenu und in diesen packt man alle relevanten Daten zum Menü.

Das wären:

SkinMenu\_Bitmap.txt

hier werden die Pfade zu den BMPs eingetragen

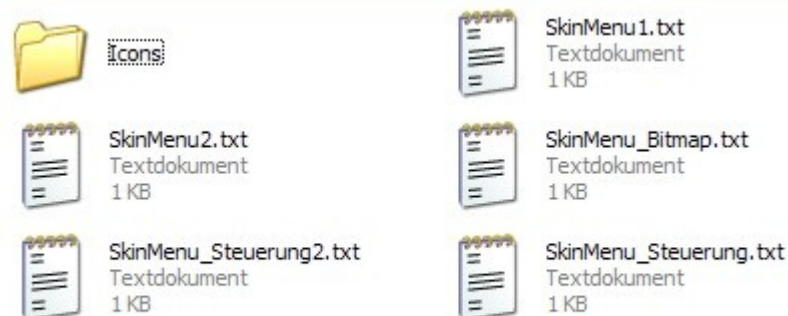
SkinMenu\_Steuerung.txt

hier werden die Befehle eingetragen

SkinMenu1.txt

hier wird das Menü erstellt

Einen Unterordner /Icons oder /BMP in den die Grafiken kommen.



## Beispiel einer Bitmap Datei:

```
// Koord-Bitmapdatei: SkinMenu_Bitmap.txt
//
1:\My Flash Disk\programme\SkinMenu\Icons\Button1.bmp|0,0,0,0,0,0,0
2:\My Flash Disk\programme\SkinMenu\Icons\Button.bmp|0,0,0,0,0,0,0
3:\My Flash Disk\programme\SkinMenu\Icons\ButtonKlein1.bmp|0,0,0,0,0,0,0
4:\My Flash Disk\programme\SkinMenu\Icons\ButtonKlein.bmp|0,0,0,0,0,0,0
5:
6:
7:
8:
```

Die Zahlen vor den Pfaden werden benötigt um in der SkinMenu1.txt den Bezug zu dem richtigem Button zu erstellen (alternativ kann man auch mit Zeilennummern arbeiten). Nach der Zahl muss der genaue Pfad zu dem gewünschtem Bitmap angegeben werden.

### Aufbau der Bitmap-Textdatei:

```
<Zeilennummer>:<voller Pfad zur
BMP-Datei>|<x>,<y>,<width>,<height>,<r-transparent>,<g-
transparent>,<b-transparent>
```

Ist <width> oder <height> Null, so wird die Größe der Bitmap genommen, andernfalls kann man einen Teil der Bitmap ausschneiden; mit r, g, b wird die transparente Farbe im Bitmap definiert. Will man keine Transparenz haben, so muss man eine Farbe wählen, die in der BMP-Datei **nicht** vorkommt!

## Beispiel einer Steuerungs Datei:

```
// Koord-Steuerungsdatei: SkinMenu_Steuerung.txt
//
// Das ist Zeile 3
window=0 SendKey=\l\My Flash Disk\programme\mortscript\NR_Skinbuilder.exe|
window=0 SendKey=\l\My Flash Disk\programme\mortscript\NR_Elemente.exe|
window=0 SendKey=\l\My Flash Disk\programme\mortscript\NR_Buttonfarbe.exe|
window=0 SendKey=\l\My Flash Disk\programme\mortscript\NR_TMC_Einstellung.exe|
window=0 SendKey=\l\My Flash Disk\programme\mortscript\NR_horichange.exe|
window=0 SendKey=\l\My Flash Disk\programme\mortscript\NR_POIWarnerWarnung.exe|
*include=\My Flash Disk\programme\SkinMenu\SkinMenu2.txt
*~
// Das ist Zeile 12
```

Jeder Befehle muss in einer Zeile durch geschrieben werden. In der SkinMenu1.txt wird dann auf die jeweilige Zeile des Befehls verwiesen.

Window=0 verhindert das das Koord-Fenster angezeigt wird.

\*include= bindet eine weiter Menüseite ein, in diesem Beispiel die zweite Seite SkinMenu2.txt

Der Stern (\*) am Anfang eines Befehls in der Steuerungs-Datei bewirkt, dass der Server **nicht** beendet wird, sondern nur die Fensterdarstellung geändert (überschrieben) wird.

## Beispiel einer SkinMenu1.txt Datei:

```
// SkinMenu 1
// Dateiname: SkinMenu.txt
// Steuerdatei: SkinMenu_Steuerung.txt
// Bitmapdatei: SkinMenu_Bitmap.txt
//
Window=0 KoordServ=3 wndrefresh=-1 t=1 rb=0 gb=0 bb=0
PixWnd=
ServWnd=text,0,0,480,272,1401032,255,255,255,-1,1{Skin-Einstellungen 1}\\
\text,12,47,143,81,5016,255,255,255,-1,-4000,1002{Skinbuilder}\\
\text,12,140,143,81,5016,255,255,255,-1,-5000,1002{Karten Elemente}\\
\text,168,47,143,81,5016,255,255,255,-1,-6000,1002{Button Farbe}\\
\text,168,140,143,81,5016,255,255,255,-1,-7000,1002{TMC Einstellung}\\
\text,324,47,143,81,5016,255,255,255,-1,-8000,1002{Horizonttauscher}\\
\text,324,140,143,81,5016,255,255,255,-1,-9000,1002{POI Warner Einstellung}\\
\text,324,232,143,38,5016,255,255,0,-1,-10000,3004{Seite vor}\\
\text,168,232,143,38,-105038,255,0,0,-1,-11000,3004{Exit}|
ServKeyFile=\My Flash Disk\programme\SkinMenu\SkinMenu_Steuerung.txt|
ServBmpFile=\My Flash Disk\programme\SkinMenu\SkinMenu_Bitmap.txt|
```

### Das „Herzstück“ des Koord Menüs:

Hier werden die Button Positionen, Form, Beschriftung etc. festgelegt.

Die Datei im einzelnen:

```
// SkinMenu 1
// Dateiname: SkinMenu.txt
// Steuerdatei: SkinMenu_Steuerung.txt
// Bitmapdatei: SkinMenu_Bitmap.txt
//
```

Dies ist ein Bezeichner (Kommentarzeilen), hier werden keine Befehle fest gelegt, Ihr solltet es aber genauso machen, damit Ihr später noch nach schauen könnt welche Menüdateien alles beteiligt sind.

```
Window=0 KoordServ=3 wndrefresh=-1 t=1 rb=0 gb=0 bb=0
```

Window=0	Koord Eingabefenster ausblenden
KoordServ=3	Koordserver starten (3 bedeutet, dass die Buttons von anderen Programmen überdeckt werden können, 1 erzeugt Buttons, die immer „on top“ sind!)
wndrefresh=-1	Bildschirm Wiederholrate in ms (-1 bedeutet kein autom. Refresh)
t=1	Transparent oder nicht transparent (Texthintergrund)
rb=0 gb=0 bb=0	Hintergrundfarbe
PixWnd=	bleibt leer wenn man mit BMPs arbeitet, damit ggf. vorher eingestellte „Styles“ gelöscht werden. Man kann diese natürlich auch bewusst mit Bitmaps kombinieren.

```
ServWnd=text,0,0,480,272,1401032,255,255,255,-1,1{Skin-Einstellungen 1}\\
```

OK, hier geht es zum eingemachten, ServWnd= für die Übergabe der Daten an den Koordserver

text	schafft die Möglichkeit den Button zu Beschriften
0,0	gibt die Position von links oben an
480,272	gibt die Größe des Fensters an
1401032	setzt den Text oben mittig in den Hintergrund (siehe unten)
255,255,255	gibt die Transparente Farbe des Hintergrundes an
-1	Parameter "event" (siehe unten)
1	Parameter "Bitmap" (siehe unten)
{ }	Hier den Button Text eingeben
ServKeyFile=	hier wird die Steuerung.txt Datei eingebunden
ServBmpFile=	hier wird die Bitmap.txt Datei eingebunden

Bei beiden muss der komplette Pfad zur Datei angegeben werden.

### 1) Parameter „Textformat“,

also der sechste Parameter, den man angibt um das Textformat zu definieren:

032 -> Texthöhe in Pixel (Wertebereich 0..999)

01 mit 1000 multipliziert-> Textausrichtung (Wertebereich 0..10)

0	1	2
4	5	6
8	9	10

14 mit 1000\*100 multipliziert-> TextStyle (Wertebereich 0..99). Nicht definierte Textstyles (PixWnd=2,...) werden ohne Style angezeigt (default).

### 2) Parameter "event",

also vorletzter Parameter, den man negativ angibt, um eine Zeile in der `Steuerungs.txt` anzusprechen:

Verwendung wie bisher, dann wird der "event" beim DRÜCKEN ausgeführt (Zeilennummern 0..999)

Mit 1000 multipliziert definiert den "event" beim LOSLASSEN (Zeilennummern 1..999)

Mit 1000\*1000 multipliziert definiert den "event" beim LANGDRÜCKEN (Zeilennummern 1..99)

Mit 1000\*1000\*100 multipliziert definiert den "event" beim DOPPELKLICK (Zeilennummern 1..20)

Die Werte können natürlich auch kombiniert werden:

1002 -> Zeile 2 beim Drücken und Zeile 1 beim Loslassen

4000000 -> Zeile 4 bei langem Klick

504000000 -> Zeile 4 bei langem Klick und Zeile 5 bei Doppelklick

#### Hinweis:

"Langer Klick" bedeutet Drücken und gedrückt lassen für ca. 2-3 Sekunden

Die Zeit für "Doppelklick" wird in der Systemsteuerung definiert.

Die Abfolge der events bei *Doppelklick* ist: *Klick-Loslassen-Doppelklick-Loslassen*

Die Abfolge der events bei *langem Klick* ist: *Klick-langer Klick-Loslassen*

Die Abfolge der events bei „normalem“ Klick ist: *Klick-Loslassen*

Der event wird immer an der Stelle ausgeführt, an der man sich gerade befindet. Das klingt banal, kann sich aber dann auswirken, wenn man in gedrücktem Zustand die Position ändert (Klick-Ziehen-Loslassen).

### 3) Parameter "*Bitmap*",

also letzter Parameter, den man angibt um eine Zeile in der `Bitmap.txt` anzusprechen:

Verwendung wie bisher, dann wird das Bitmap für den Zustand NORMAL dargestellt (Zeilennummern 0..999)

Mit 1000 multipliziert dann wird das Bitmap für den Zustand GEDRÜCKT dargestellt (Zeilennummern 1..999)

Mit 1000\*1000 multipliziert dann wird das Bitmap für den Zustand LANGGEDRÜCKT dargestellt (Zeilen 1..99)  
(Der Zustand "*Doppelklick*" macht keinen Sinn!)

Die Werte können natürlich auch kombiniert werden:

2 -> Immer Bitmap 2

1002 -> Bitmap 2 beim Drücken und Bitmap 1 beim Loslassen

3001002 -> Bitmap 2 beim Drücken und Bitmap 1 beim Loslassen und Bitmap 3 bei Langem Klick.

Und so kann ein Menü aussehen.

