

uMenuGr

JEDNOTKA PRO VYTVÁŘENÍ UŽIVATELSKÝCH GRAFICKÝCH MENU

Příručka uživatele a programátora



SofCon[®] spol. s r.o.
Střešovická 49
162 00 Praha 6
tel/fax: +420 220 180 454
E-mail: sofcon@sofcon.cz
www: <http://www.sofcon.cz>

Informace v tomto dokumentu byly pečlivě zkontrolovány a SofCon věří, že jsou spolehlivé, přesto SofCon nenese odpovědnost za případné nepřesnosti nebo nesprávnosti zde uvedených informací.

SofCon negarantuje bezchybnost tohoto dokumentu ani programového vybavení, které je v tomto dokumentu popsáno. Uživatel přebírá informace z tohoto dokumentu a odpovídající programové vybavení ve stavu, jak byly vytvořeny a sám je povinen provést validaci bezchybnosti produktu, který s použitím zde popsaného programového vybavení vytvořil.

SofCon si vyhrazuje právo změny obsahu tohoto dokumentu bez předchozího oznámení a nenese žádnou odpovědnost za důsledky, které z toho mohou vyplynout pro uživatele.

Datum vydání: 16.05.2003

Datum posledního uložení dokumentu: 16.05.2003

(Datum vydání a posledního uložení dokumentu musí být stejné)

Upozornění:

V dokumentu použité názvy výrobků, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Obsah :

1.O dokumentu	5
1.1. Revize dokumentu	5
1.2. Účel dokumentu	5
1.3. Rozsah platnosti	5
1.4. Související dokumenty	5
2.Termíny a definice	5
3.Úvod	6
4.Popis konstant a typů	6
5.Popis objektu tMenuGr	6
5.1. Proměnné	6
5.2. Metody	7
5.2.1. Init	7
5.2.2. DisplayMenuAndGraphic	7
5.2.3. InitParam	7
5.2.4. SetBkGround	7
5.2.5. SetHelpVar	7
6.Grafický souřadný systém	8
7.Syntaxe textového Stringu	8
8.Syntaxe grafického Stringu	9

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.XX	We		První vydání
1.10	2.XX	Tu	16.05.2003	Úprava dokumentu dle ISO9000

1.2. Účel dokumentu

Tento dokument slouží jako popis jednotky pro vytváření uživatelských grafických menu.

1.3. Rozsah platnosti

Určen pro programátory a uživatele programového vybavení SofCon.

1.4. Související dokumenty

Pro čtení tohoto dokumentu je potřeba seznámit se s manuálem uAMenu, uMenuChr, uATerm, uTermChr a uTermGr.

Popis formátu verze knihovny a souvisejících funkcí je popsán v manuálu LibVer.

2. Termíny a definice

Používané termíny a definice jsou popsány v samostatném dokumentu Termíny a definice.

3. Úvod

Tato knihovna implementuje rozšíření objektu **tMenuChr** pro práci s grafickým terminálem. Oproti objektu **tMenuChr**, který definuje pouze znaková menu, tento objekt rozšiřuje možnosti o grafický výstup. Jedná se o zobrazení bitmapy a zobrazení základních grafických útvarů jako je bod, úsečka, kružnice, obdélník a vyplněný obdélník. V zobrazení lze použít všechny prvky naráz. To znamená grafiku, podkladovou bitmapu a zobrazení alfanumerických znaků.

Objekt grafického menu **tMenuGr** pracuje pouze s indexy do pole bitmap a s řetězci popisujícími obsah obrazovky terminálu. Tyto indexy a řetězce předává objektu terminálu a ten je v procesu obnovy obrazovky předává objektu displeje. Až objekt displeje provádí pomocí grafické knihovny interpretaci těchto definičních řetězců a indexů do videostránky displeje a tu přenáší na fyzický displej.

Pole bitmap musí být uživatelem vytvořeno vně objektu grafického menu i objektu terminálu a ukazatel na ně se předává při inicializaci objektu displeje. Stejně je to s polem fontů.

Zděděné metody zde nejsou popsány, jejich popis je možno najít v manuálu ke knihovně **tAMenu** a **tMenuChr**.

4. Popis konstant a typů

```
cVerNo = např. $0251; { BCD formát }
cVer   = např. '02.51,07.08.2003';
```

Číslo verze jednotky v BCD tvaru a v textové podobě včetně datumu změny.

```
pBkIndex      =      ^Integer;
pBkIndex je typ pro uložení čísla pozadí.
```

```
pMMenuGr      =      ^tMMenuGr;
tMMenuGr      =      procedure(P: pMenuGr);
```

tMMenuGr je procedura s parametrem ukazatel na objekt **tMenuGr**. Díky tomuto parametru může procedura přímo přistupovat k metodám a proměnným objektu.

5. Popis objektu tMenuGr

```
pMenuGr      =      ^tMenuGr;
tMenuGr      =      object(tMenuChr)
```

5.1. Proměnné

```
pDispGraphicString : pString;
```

pDispGraphicString je ukazatel na řetězec s ESCape sekvencí definující obsah grafické obrazovky terminálu. **DispGraphicString** je vlastně ekvivalent **MenuDispStr**. Místo textu určuje použití grafických prvků aktuální obrazovce menu.

```
pBkGroundIdx      :      pBkIndex;
```

pBkGroundIndex je ukazatel na číslo typu Integer, určující index bitmapy, která má být umístěna na pozadí menu.

5.2. Metody

5.2.1. Init

```
constructor Init(AMenu: pMenu; EndIndex: Word;
  var DisplayString, HelpString, GraphicStr: String;
  BkGroundIndex: pBkIndex; ATerminal: pTermGr; ParTerm: tParamStr;
  WaitMenu: Word);
```

Konstruktor **Init** inicializuje objekt jako znakové menu a pak si do ukazatelů na DisplayString, GraphicString a BkGroundIndex přiřadí proměnné z parametrů konstruktoru. Provede se přibližně toto:

```
tMenuChr.Init(Menu, EndIndex, DisplayString, HelpString, Terminal,
  ParTerm, WaitMenu);
pDispGraphicString := @GraphicStr;
pBkGroundIndex      := @BkGroundIndex;
```

Nakonec konstruktor volá metodu **InitParam**, která do proměnné **DisplayMenu** objektu **tAMenu** místo **mDisplayMenu** znakového menu přiřadí zobrazovací proceduru **DisplayMenuAndGraphics**.

5.2.2. DisplayMenuAndGraphic

```
procedure DisplayMenuAndGraphic;
```

Statická metoda **DisplayMenuAndGraphic** je náhradou za metody **mDisplayMenu** z knihovny **uMenuChr**. Stejně jako **mDisplayMenu** zajišťuje vykreslení obrazovky menu na terminálu, ale rozšiřuje ji o detekci grafického stringu a podložení bitmapy.

5.2.3. InitParam

```
procedure InitParam; virtual;
```

Tato metoda rozšiřuje metodu **InitParam** znakového menu o přiřazení zobrazovací procedury **DisplayMenuAndGraphics** do proměnné **DisplayMenu** objektu **tAMenu** místo **mDisplayMenu** znakového menu. Metoda **InitParam** se volá při každém přechodu mezi menu a nastavuje parametry menu na implicitní hodnoty.

5.2.4. SetBkGround

```
procedure SetBkGround(NewBkGroundIndex : pBkIndex); virtual;
```

Tato metoda nastavuje proměnnou **pBkGroundIndex** na novou hodnotu. Slouží pro nastavení čísla indexu aktuálně vypisované bitmapy v menu.

5.2.5. SetHelpVar

```
procedure SetHelpVar(var HGrStr:string;var HBkBmp:integer);virtual;
```

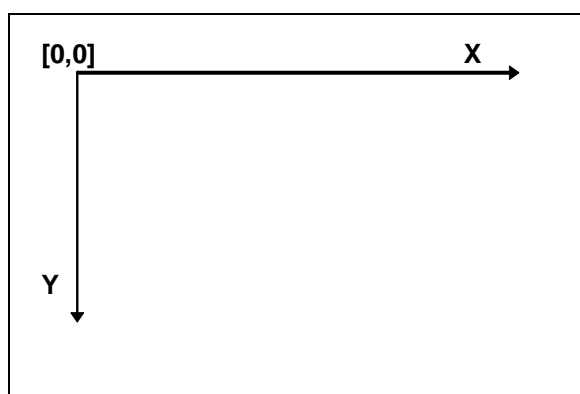
Metoda **SetHelpVar** slouží k nastavení ukazatelů na proměnné s textovým popisem grafických objektů a indexu bitmapy pozadí nápovědy. Tyto proměnné jsou

předány v parametrech **HGrStr** a **HBkBmp**. Tuto činnost provádí voláním metody **tTermGr.TermSetHelpVar**.

Tyto ukazatele jsou obsaženy v objektu **tTermGr** a jsou implicitně nastaveny konstruktorem na nil. Pokud uživatel toto nastavení nezmění nebudou se grafické objekty a bitmapa pozadí v nápovědě zobrazovat.

6. Grafický souřadný systém

Pro potřeby umístování grafických objektů na display je zaveden souřadný systém s metrikou vyjádřenou v jednotkách pixel (tj. nejmenší zobrazitelný bod na displeji). Počátek souřadného systému leží v levém horním rohu display, osa X je orientována vodorovně směrem doprava, osa Y je orientována směrem dolů.



7. Syntaxe textového Stringu

Syntaxe zápisu textového stringu je následující:

<TextString>=

{<ESC> F<FontIndex>, <X>, <Y> {, <Rop>};} <text> {<TextString>}

kde <ESC> je znak ESC (#27)

kde <FontIndex> je index fontu v poli uživatelských fontů

kde <X> je X-souřadnice levého horního rohu výpisu.

kde <Y> je Y-souřadnice levého horního rohu výpisu.

kde <Rop> je operace nad video-pamětí při vykreslování textu (implicitně OR).

Rop=0 ... operace OR

Rop=1 ... operace NOT

Rop=2 ... operace XOR

<text> je vypisovaný text

Implicitní umístění výpisu je v levém horním rohu obrazovky a implicitní font je font s indexem 0. Je-li použita volba fontu a umístění nejvýše jednou je prováděn výpis ve znakovém rastru definovaném při volání konstrukturu objektu displeje. V opačném případě výpis není prováděn ve znakovém rastru a každá volba fontu a umístění se vztahuje na samostatný řádek.

8. Syntaxe grafického Stringu

Při popisu syntaxe jsou použity tyto konvence: symboly uzavřené v <> označují číselné hodnoty zapsané znakově, parametry uvedené ve složených závorkách { } jsou parametry volitelnými.

Syntaxe zápisu tohoto deskriptoru je následující:

- **objekt bod** $P(<X>,<Y>\{,<Rop>\})$
kde <X> je X-souřadnice bodu na display v jednotkách pixel
kde <Y> je Y-souřadnice bodu na display v jednotkách pixel
kde <Rop> je operace nad video-pamětí při vykreslování grafického objektu
Rop=0 ... operace OR
Rop=1 ... operace NOT
Rop=2 ... operace XOR
- **objekt úsečka** $L(<X1>,<Y1>,<X2>,<Y2>\{,<Rop>\})$
kde <X1> <Y1> jsou souřadnice počátečního bodu úsečky a
kde <X2> <Y2> jsou souřadnice koncového bodu úsečky
- **objekt obdélník** $R(<X1>,<Y1>,<X2>,<Y2>\{,<Rop>\})$
kde <X1> <Y1> jsou souřadnice levého horního rohu,
kde <X2> <Y2> jsou souřadnice pravého dolního rohu
- **objekt vyplněný obdélník** $F(<X1>,<Y1>,<X2>,<Y2>\{,<Rop>\})$
kde <X1> <Y1> jsou souřadnice levého horního rohu,
kde <X2> <Y2> jsou souřadnice pravého dolního rohu
- **objekt kružnice** $C(<CX>,<CY>,<R>\{,<Rop>\})$
kde <CX><CY> jsou souřadnice středu kružnice,
kde <R> je poloměr kružnice