

LIBRERIA FRAMES

di Aldo Carpanelli – v1.1.0, 18/10/2023

Descrizione

Una libreria **amatoriale** in ANSI C, creata con lo scopo di gestire in modo presuntamente più semplice e lineare la creazione in *Win32* di immagini in memoria che includano più zone utilizzabili indipendentemente.

Tramite apposite funzioni, è possibile anche utilizzare le immagini in memoria come elementari caratteri tipografici.

Necessita della libreria **amatoriale** in ANSI C *OffscreenWorld*.

Tipi di dati specifici della libreria

```
typedef struct {  
    RECT    a;        /* a: area */  
    POINT    hs;       /* hs: hot spot (determina il "punto focale" della zona */  
} Zone;
```

```
typedef struct {  
    OffscreenWorld ow;  
    Zone *zn;          /* zn: le zone in ow */  
    int qZn;           /* qZn: quantita' delle zone nell'array zn */  
    char c[256];       /* c: l'elenco dei caratteri, qualora sia un alfabeto */  
} Frames;
```

Le funzioni

```
int FR_ValidFrames( const Frames *fr );  
int FR_ValidAlphabet( const Frames *fr );
```

ATTENZIONE:

Per velocizzare l'esecuzione, nessuna delle funzioni che seguono controlla la validità del parametro passato in `fr`. Per evitare problemi, se non si è certi di tale validità, la si verifichi espressamente con `FR_ValidFrames()` o con `FR_ValidAlphabet()` (a seconda se si sta trattando una semplice serie di fotogrammi o una sequenza di immagini abbinata ad una sequenza di caratteri).

```
int FR_Create( Frames *fr, LONG w, LONG h );
```

Secondo l'estensione del nome in `dataFileName`, `FR_CreateFromDataFile()` carica i dati riferiti a una semplice serie di fotogrammi o una sequenza di immagini abbinata ad una serie di caratteri.

```
int FR_CreateFromDataFile( Frames *fr,  
                           const char *dataFileName );
```

```
int FR_CreateFromResource( Frames *fr,  
                           const Zone *zn, int qZn,  
                           COLORREF trnsp,  
                           HINSTANCE hInst, WORD idRes );
```

```
int FR_CreateFromBitmap( Frames *fr,  
                         const Zone *zn, int qZn,  
                         COLORREF trnsp, HBITMAP hBmp );
```

```
void FR_Destroy( Frames *fr );
```

```
int FR_SetZone( Frames *fr, int iZone,  
               int x, int y, int w, int h,  
               int hSpot_OffsetX, int hSpot_OffsetY );
```

```
int FR_SetZones( Frames *fr, const Zone *zones, int qZones );
```

```
int FR_GetZoneArea( const Frames *fr, int iZone, LPRECT a );
```

```
int FR_GetZoneHotSpot( const Frames *fr, int iZone,  
                      LPPOINT hs );
```

```
int FR_GetZoneBounds( const Frames*fr, int iZone,  
                     LONG x, LONG y, LPRECT zb );
```

```
int FR_DrawFrame( const Frames *fr, HDC hdc,  
                  LONG x, LONG y, int iZone );
```

```
/** ==> ALPHABET <===== **/
```

```
int FR_CreateAlphabetFromResource( Frames *fr,  
                                   const Zone *zn, int qZn,  
                                   COLORREF trnsp,  
                                   HINSTANCE hInst, WORD idRes,  
                                   const char *characters );
```

```
int FR_CreateAlphabetFromBitmap( Frames *fr,  
                                 const Zone *zn, int qZn,  
                                 COLORREF trnsp, HBITMAP hBmp,  
                                 const char *characters );
```

```
int FR_SetCharacters( Frames *fr, const char *characters );
```

```
int FR_GetQChar( const Frames *fr, int *qChar );
```

```
int FR_GetTxtWidth( const Frames *fr,  
                    const char *txt, int lTxt, LPLONG wTxt );  
  
int FR_GetTxtHeight( const Frames *fr,  
                     const char *txt, int lTxt, LPLONG hTxt );  
  
int FR_GetTxtSize( const Frames *fr,  
                   const char *txt, int lTxt, LPSIZE szTxt );  
  
int FR_DrawTxt( const Frames *fr, HDC hdc, LONG x, LONG y,  
                const char *txt, int lTxt, LPRECT rDrawArea );
```

```
int FR_ValidFrames( const Frames *fr );
```

Verifica che la struttura `Frames` passata tramite l'unico parametro `fr` sia utilizzabile per gli scopi che le sono propri.

Parametri

`const Frames *fr` Un puntatore alla struttura da verificare.

Valore di ritorno

`int` Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria `OffscreenWorld`.


```
int FR_ValidAlphabet( const Frames *fr );
```

Verifica che la struttura `Frames` passata tramite l'unico parametro `fr` sia utilizzabile per gli scopi che le sono propri in quanto contenente i dati necessari all'impiego come "alfabeto" grafico.

Parametri

`const Frames *fr` Un puntatore alla struttura da verificare.

Valore di ritorno

`int` Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria `OffscreenWorld`.

```
int FR_Create( Frames *fr, LONG w, LONG h );
```

Crea nella struttura di tipo `Frames` puntata dal parametro `fr` tutto quel che serve per ottenere uno *sprite* con un'unico fotogramma largo `w` pixel ed alto `h` pixel. Il fotogramma è inizialmente costituito da un riquadro completamente nero.

Parametri

<code>Frames *fr</code>	Puntatore alla struttura di tipo <code>Frames</code> destinata a ricevere quel che serve per rendere funzionante lo <i>sprite</i> .
<code>LONG w</code>	Larghezza in pixel del fotogramma da creare.
<code>LONG h</code>	Altezza in pixel del fotogramma da creare.

Valore di ritorno

<code>int</code>	Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria <code>OffscreenWorld</code> .
------------------	--

```
int FR_CreateFromDataFile( Frames *fr,  
                           const char *dataFName );
```

Crea nella struttura di tipo `Frames` puntata dal parametro `fr` tutto quel che serve per ottenere uno *sprite* corrispondente alle caratteristiche definite nel file `dataFName`.

Per il formato che il file `dataFName` deve rispettare, si veda l'apposita sezione.

Parametri

<code>Frames *fr</code>	Puntatore alla struttura di tipo <code>Frames</code> destinata a ricevere quel che serve per rendere funzionante lo <i>sprite</i> .
<code>Const char *dataFName</code>	Stringa che identifica il file dal quale caricare i dati necessari per la creazione dello <i>sprite</i> . Il file può avere estensione <code>.fgm</code> o <code>.abt</code> . Se il file ha estensione <code>.fgm</code> viene creato uno <i>sprite</i> “ordinario”, mentre se ha estensione <code>.abt</code> viene creato uno <i>sprite</i> utilizzabile come alfabeto “grafico”.

Valore di ritorno

`int` Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria [OffscreenWorld](#).

```
int FR_CreateFromResource( Frames *fr,  
                           const Zone *zn,  
                           int qZn,  
                           COLORREF trnsp,  
                           HINSTANCE hInst,  
                           WORD idRes );
```

Crea nella struttura di tipo `Frames` puntata dal parametro `fr` tutto quel che serve per ottenere uno *sprite* costituito da `qZn` aree, eventualmente rese trasparenti in corrispondenza del colore `trnsp`.

L'immagine da copiare nello *sprite* viene rilevata dalla risorsa con numero identificativo `idRes`, nel contesto del modulo indicato da `hInst`.

Per il formato che il puntatore `zn` deve rispettare, si veda l'apposita sezione.

Parametri

<code>Frames *fr</code>	Puntatore alla struttura di tipo <code>Frames</code> destinata a ricevere quel che serve per rendere funzionante lo <i>sprite</i> .
<code>const Zone *zn</code>	Puntatore all'array di strutture di tipo <code>Zone</code> che definiscono le aree dei singoli fotogrammi nell'ambito della bitmap caricata come risorsa <code>idRes</code> e copiata nello <i>sprite</i> .
<code>int qZn</code>	La quantità delle zone nell'array <code>zn</code> .
<code>COLORREF trnsp</code>	Il colore da rendere trasparente.
<code>HINSTANCE hInst</code>	Il modulo dal quale caricare la risorsa <code>hInst</code> .
<code>WORD idRes</code>	Il numero identificativo della risorsa da caricare dal modulo <code>idRes</code> .

Valore di ritorno

<code>int</code>	Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria <code>OffscreenWorld</code> .
------------------	--

```
int FR_CreateFromBitmap( Frames *fr,  
                        const Zone *zn,  
                        int qZn,  
                        COLORREF trnsp,  
                        HBITMAP hBmp );
```

Crea nella struttura di tipo **Frames** puntata dal parametro **fr** tutto quel che serve per ottenere uno *sprite* costituito da **qZn** aree, eventualmente rese trasparenti in corrispondenza del colore **trnsp**.

L'immagine da copiare nello *sprite* viene è data dalla bitmap **hBmp**.

Per il formato che il puntatore **zn** deve rispettare, si veda l'apposita sezione.

Parametri

Frames *fr

Puntatore alla struttura di tipo **Frames** destinata a ricevere quel che serve per rendere funzionante lo *sprite*.

`const Zone *zn` Puntatore all'array di strutture di tipo `Zone` che definiscono le aree dei singoli fotogrammi nell'ambito della bitmap che viene copiata nello *sprite*.

`int qZn` La quantità delle zone nell'array `zn`.

`COLORREF trnsp` Il colore da rendere trasparente.

`HBITMAP hBmp` La bitmap da copiare nello *sprite*.

Valore di ritorno

`int` Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria `OffscreenWorld`.


```
void FR_Destroy( Frames *fr );
```

Distrugge nella struttura di tipo `Frames` puntata dal parametro `fr` tutto quel che serviva per ottenere lo *sprite*. Prima di effettuare la distruzione degli elementi, li verifica uno ad uno.

In uscita, il contenuto di `fr` risulta azzerato e la memoria dinamica correlata, liberata.

Parametri

`Frames *fr` Puntatore alla struttura di tipo `Frames` che contiene lo *sprite* da distruggere.

```
int FR_SetZone( Frames *fr, int iZone,  
               int x, int y, int w, int h,  
               int hSpot_OffsetX,  
               int hSpot_OffsetY );
```

Verifica che `fr` non sia `NULL`, che `iZone` ricada entro la gamma degli indici validi e che l'array delle zone sia stato effettivamente allocato.

NON controlla che `fr->ow` punti a un `OffscreenWorld` valido.

Al termine delle verifiche, imposta la zona richiesta secondo le caratteristiche indicate in `x`, `y`, `w`, `h`, `hSpot_OffsetX` e `hSpot_OffsetY`.

Parametri

<code>Frames *fr</code>	Puntatore alla struttura di tipo <code>Frames</code> che contiene lo <i>sprite</i> del quale impostare la zona in posizione <code>iZone</code>
<code>int iZone</code>	Indice della zona da impostare

<code>int x</code>	Posizione e dimensioni dell'area dedicata alla zona
<code>int y</code>	
<code>int w</code>	
<code>int h</code>	
<code>int hSpotOffsetX</code>	Punto di centratura della zona
<code>int hSpotOffsetY</code>	

Valore di ritorno

<code>int</code>	Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria OffscreenWorld .
------------------	---

```
int FR_SetZones( Frames *fr,  
                const Zone *zones, int qZones );
```

Verifica che né `fr` né `zones` siano puntatori `NULL`, e che `qZones` sia `>0`.

Al termine della verifica, sostituisce l'array delle zone correntemente contenuto in `*fr` e lo sostituisce con **una copia** di quello puntato da `zones`.

Parametri

<code>Frames *fr</code>	Puntatore alla struttura di tipo <code>Frames</code> che contiene lo <i>sprite</i> del quale sostituire l'array <code>zone</code>
<code>const Zone *zones</code>	Array del quale collocare una copia in <code>*fr</code>
<code>int qZones</code>	La quantità delle zone nell'array <code>zones</code>

Valore di ritorno

<code>int</code>	Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria <code>OffscreenWorld</code> .
------------------	--

```
int FR_GetZoneArea( const Frames *fr,  
                    int iZone, LPRECT a );
```

Verifica che `fr`, `a` e `fr->zones` non siano puntatori `NULL` e che `iZone` sia un indice a una zona valida.

Al termine della verifica, colloca in `*a` l'area della zona `iZone`, sotto forma di `RECT`.

Parametri

<code>Frames *fr</code>	Puntatore alla struttura di tipo <code>Frames</code> che contiene lo <i>sprite</i> del quale leggere l'area della zona in posizione <code>iZone</code> nell'array <code>zone</code>
<code>int iZone</code>	Indice della zona della quale rilevare l'area
<code>LPRECT a</code>	L'area della zona <code>iZone</code> , sotto forma di <code>RECT</code> , viene copiata in <code>*a</code>

Valore di ritorno

<code>int</code>	Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria <code>OffscreenWorld</code> .
------------------	--

```
int FR_GetZoneHotSpot( const Frames *fr,  
                        int iZone, LPPPOINT hs );
```

Verifica che *fr*, *hs* e *fr->zones* non siano puntatori **NULL** e che *iZone* sia un indice a una zona valida, poi colloca in **hs* l'*hotspot* della zona *iZone*, come **POINT**. Le coordinate in **hs* sono riferite all'angolo superiore sinistro dell'area dello *sprite*.

Parametri

Frames *fr	Puntatore alla struttura di tipo Frames che contiene lo <i>sprite</i> del quale leggere l' <i>hotspot</i> della zona in posizione <i>iZone</i> nell'array <i>zone</i>
int iZone	Indice della zona della quale rilevare l' <i>hotspot</i>
LPPPOINT hs	L' <i>hotspot</i> della zona in posizione <i>iZone</i> nell'array <i>zone</i> , sotto forma di POINT , viene copiata in <i>*hs</i>

Valore di ritorno

int	Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria OffscreenWorld .
------------	--

```
int FR_GetZoneBounds( const Frames*fr, int iZone,
                      LONG x, LONG y, LPRECT zb );
```

Verifica che `zb` non sia `NULL`, quindi usa `FR_GetZoneArea()` per ricavare l'area della zona in posizione `iZone` nell'array `fr->zone`. Quell'area viene traslata in `(x,y)`, ma tenendo conto dei valori di scostamento dell'*hotspot*.

Parametri

<code>Frames *fr</code>	Puntatore alla struttura di tipo <code>Frames</code> che contiene lo <i>sprite</i> del quale ricavare i dati richiesti
<code>int iZone</code>	Indice della zona della quale rilevare i dati richiesti
<code>LPRECT zb</code>	Il perimetro della zona in posizione <code>iZone</code> nell' <i>array zone</i> , traslato in <code>(x,y)</code> tenendo conto dei valori di scostamento dell' <i>hotspot</i> , viene copiato sotto forma di <code>RECT</code> in <code>*zb</code>

Valore di ritorno

<code>int</code>	Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria <code>OffscreenWorld</code> .
------------------	--

```
int FR_DrawFrame( const Frames *fr,  
                  HDC hdc, LONG x, LONG y,  
                  int iZone );
```

Verifica che `fr` non sia un puntatore `NULL` e che `iZone` sia un indice a una zona valida, quindi ricorre a `OW_DrawImgPortion()` per tracciare il fotogramma richiesto in `(x,y)` nel *device context* `hdc`.

Parametri

<code>Frames *fr</code>	Puntatore alla struttura di tipo <code>Frames</code> che contiene il fotogramma da tracciare in <code>hdc</code>
<code>HDC hdc</code>	<i>Device context</i> dove si desidera che venga tracciato il fotogramma
<code>LONG x</code>	Le coordinate alle quali tracciare il fotogramma; a queste coordinate viene applicato lo scostamento indicato dall' <i>hotspot</i> della zona <code>iZone</code>
<code>LONG y</code>	
<code>int iZone</code>	Indice della zona corrispondente al fotogramma da tracciare

Valore di ritorno

<code>int</code>	Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria OffscreenWorld .
------------------	---

ALPHABET

```
int FR_CreateAlphabetFromResource(  
    Frames *fr,  
    const Zone *zn, int qZn,  
    COLORREF trnsp,  
    HINSTANCE hInst, WORD idRes,  
    const char *characters );
```

Usa `FR_CreateFromResource()` per creare una struttura `Frames` funzionale, quindi le abbina i caratteri indicati nell'array `characters` per mezzo della funzione `FR_SetCharacters()`.

Parametri

<code>Frames *fr</code>	Puntatore alla struttura di tipo <code>Frames</code> destinata a ricevere quel che serve per rendere funzionante lo <i>sprite</i> .
<code>const Zone *zn</code>	Puntatore all'array di strutture di tipo <code>Zone</code> che definiscono le aree dei singoli fotogrammi nell'ambito della bitmap caricata come risorsa <code>idRes</code> e copiata nello <i>sprite</i> .

<code>int qZn</code>	La quantità delle zone nell'array <code>zn</code> .
<code>COLORREF trnsp</code>	Il colore da rendere trasparente.
<code>HINSTANCE hInst</code>	Il modulo dal quale caricare la risorsa <code>idRes</code> .
<code>WORD idRes</code>	Il numero identificativo della risorsa da caricare dal modulo <code>hInst</code> .
<code>const char *characters</code>	Una stringa C dalla quale verranno tratti i caratteri da conservare come elenco dei caratteri alfabetici.

Valore di ritorno

<code>int</code>	Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria <code>OffscreenWorld</code> .
------------------	--

```
int FR_CreateAlphabetFromBitmap(  
    Frames *fr,  
    const Zone *zn, int qZn,  
    COLORREF trnsp,  
    HBITMAP hBmp,  
    const char *characters );
```

Usa `FR_CreateFromBitmap()` per creare una struttura `Frames` funzionale, quindi le abbina i caratteri indicati nell'array `characters` per mezzo della funzione `FR_SetCharacters()`.

Parametri

`Frames *fr`

Puntatore alla struttura di tipo `Frames` destinata a ricevere quel che serve per rendere funzionante lo *sprite*.

`const Zone *zn`

Puntatore all'array di strutture di tipo `Zone` che definiscono le aree dei singoli fotogrammi nell'ambito della bitmap che viene copiata nello *sprite*.

<code>int qZn</code>	La quantità delle zone nell'array <code>zn</code> .
<code>COLORREF trnsp</code>	Il colore da rendere trasparente.
<code>HBITMAP hBmp</code>	La bitmap da copiare nello <i>sprite</i> .
<code>const char *characters</code>	Una stringa C dalla quale verranno tratti i caratteri da conservare come elenco dei caratteri alfabetici.

Valore di ritorno

`int` Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria `OffscreenWorld`.

```
int FR_SetCharacters( Frames *fr,  
                    const char *characters );
```

Dopo aver verificato la validità dei parametri, copia la stringa `characters` in `fr->c`. La stringa `characters` non deve superare i 255 caratteri, e non deve presentare caratteri doppi.

Parametri

<code>Frames *fr</code>	Puntatore alla struttura di tipo <code>Frames</code> destinata a ricevere i caratteri <code>characters</code>
<code>const char *characters</code>	Una stringa C dalla quale verranno tratti i caratteri da conservare come elenco dei caratteri alfabetici.

Valore di ritorno

`int` Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria `OffscreenWorld`.

```
int FR_GetQChar( const Frames *fr, int *qChar );
```

Verifica che nessuno dei parametri sia `NULL`, quindi colloca in `*qChar` il valore di `fr->qZn` (ricorda: `qZn` coincide con la quantità dei caratteri presenti nella stringa `fr->c`).

Parametri

`Frames *fr`

Puntatore alla struttura di tipo `Frames` dalla quale ricavare il dato richiesto

`int *qChar`

In uscita, in `*qChar` viene collocata la quantità dei caratteri presenti in `fr->c`

Valore di ritorno

`int` Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria `OffscreenWorld`.


```
int FR_GetTxtWidth( const Frames *fr,  
                    const char *txt, int lTxt,  
                    LPLONG wTxt );
```

Dopo aver controllato la validità dei parametri, confronta ciascun carattere della stringa `txt` con le dimensioni della zona corrispondente in `fr->zn`, sommandone la larghezza fino al totale occorrente per “ospitare” l’intera stringa. Detto totale viene collocato in `*wTxt`.

Parametri

<code>Frames *fr</code>	Puntatore alla struttura di tipo <code>Frames</code> dalla quale ricavare il dato richiesto
<code>const char *txt</code>	La stringa della quale valutare la larghezza complessiva
<code>int lTxt</code>	La quantità dei caratteri nella stringa <code>txt</code>
<code>LPLONG wTxt</code>	Puntatore all’area di memoria destinata ad ospitare, in uscita, il valore rilevato

Valore di ritorno

`int` Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria [OffscreenWorld](#).

```
int FR_GetTxtHeight( const Frames *fr,  
                    const char *txt, int lTxt,  
                    LPLONG hTxt );
```

Controllata la validità dei parametri, confronta ciascun carattere della stringa `txt` con le dimensioni della zona corrispondente. Il valore dell'altezza massima rilevata viene restituito in `*hTxt`.

Parametri

<code>Frames *fr</code>	Puntatore alla struttura di tipo <code>Frames</code> dalla quale ricavare il dato richiesto
<code>const char *txt</code>	La stringa della quale valutare l'altezza massima
<code>int lTxt</code>	La quantità dei caratteri nella stringa <code>txt</code>
<code>LPLONG hTxt</code>	Puntatore all'area di memoria destinata ad ospitare, in uscita, il valore rilevato

Valore di ritorno

`int` Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria `OffscreenWorld`.

```
int FR_GetTxtSize( const Frames *fr,  
                  const char *txt, int lTxt,  
                  LPSIZE szTxt );
```

Con le funzionalità di `FR_GetTxtWidth()` e `FR_GetTxtHeight()` restituisce entrambi i valori rilevabili tramite le due funzioni.

Parametri

<code>Frames *fr</code>	Puntatore alla struttura di tipo <code>Frames</code> dalla quale ricavare il dato richiesto
<code>const char *txt</code>	La stringa della quale valutare le dimensioni
<code>int lTxt</code>	La quantità dei caratteri nella stringa <code>txt</code>
<code>LPSIZE szTxt</code>	Puntatore all'area di memoria destinata ad ospitare, in uscita, il valore rilevato

Valore di ritorno

`int` Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria `OffscreenWorld`.

```
int FR_DrawTxt( const Frames *fr,  
                HDC hdc, LONG x, LONG y,  
                const char *txt, int lTxt,  
                LPRECT rDrawArea );
```

Usando la grafica rappresentata in **fr*, traccia la stringa *txt* a partire dalle coordinate (*x,y*) nel *device context* *hdc*.

Parametri

<i>Frames *fr</i>	Puntatore alla struttura di tipo <i>Frames</i> da impiegare per tracciare la stringa <i>txt</i>
<i>const char *txt</i>	La stringa da tracciare
<i>int lTxt</i>	La quantità dei caratteri da prendere in considerazione nella stringa <i>txt</i> ; se è <i><0</i> viene usata l'intera stringa C
<i>LPRECT rDrawArea</i>	In uscita, il perimetro dell'area contenente <i>txt</i>

Valore di ritorno

int Uno dei codici di errore elencati in questa documentazione o nella documentazione relativa alla libreria *OffscreenWorld*.

