SGOSD

COLLABORATORS			
	TITLE : SGOSD		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		March 29, 2025	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

SGOSD

1.1 SGOSD.doc

sgos.library

pOS_ActivateSGObject()	pOS_ActivateSGObjectA()	pOS_CloneSGObject()
pOS_CloneSGObjectA()	pOS_CreateSGObject()	pOS_CreateSGObjectA()
pOS_CreateSGOInfo()	pOS_CreateSGOInfoA()	pOS_DeleteSGObject()
pOS_DeleteSGOInfo()	pOS_DrawSGOS()	pOS_InactivateSGObject()
pOS_LockSGOInfo()	pOS_RemoveSGObject()	pOS_SetSGObjectGfxMap()
pOS_SetSGObjectMask()	pOS_StartSGOS()	pOS_StopSGOS()
pOS_UnlockSGOInfo()	SGOSAllgemeines()	

1.2 sgos.library/SGOSAllgemeines()

```
SGOS steht für "Simple Graphics Object System" und ermöglicht die
einfache Erstellung und Verwaltung bewegter/animierter Grafikobjekte.


ACHTUNG: ZUR ZEIT werden noch KEINE Objekte an der ClipBox geklippt.



pOS_SGOInfo

    Auf folgende Struktur-Elemente darf LESEND zugegriffen werden:
    ACHTUNG: Bevor auf irgendeine Struktur im laufendem System
             zugegriffen wird, ist das entsprechende System mit
             pOS_LockSGOInfo() zu sperren !!!

  pOS_List sgoi_ActiveList;
    Aktiveliste. In dieser Liste befinden sich alle zur Zeit aktiven
    Grafikobjekte. Objekte dürfen NUR mit den entsprechenden System-
    funktionen entfernt bzw. hinzugefügt werden.

  pOS_List sgoi_InactiveList;
    Inaktiveliste. In dieser Liste befinden sich alle zur Zeit inaktiven
    Grafikobjekte. Diese werden nicht von pOS_DrawSGOS() behandelt.
```

Objekte dürfen NUR mit den entsprechenden Systemfunktionen entfernt bzw. hinzugefügt werden.

pOS_IBox sgoi_ClipBox;
Die Größe des Randes.

HINWEIS
Benötigt man ein bestimmtes Grafikobjekt öfter, so ist es günstig dieses EINMAL als Master zu erzeugen und in die Inaktivliste einzufügen. Wird nun dieses Objekt zu Darstellung gebraucht, kann es einfach per pOS_CloneSGObject() kopiert werden. Wird es nicht mehr benötigt, ist es einfach per pOS_RemoveSGObjetct() zu entfernen.
Es können beliebig viele Kopien eines Objektes existieren. Dabei werden die eigentlichen Grafikdaten nur einmal im Speicher behalten. Das Masterobjekt MUSS natürlich mindestens genausolange wie die letzte Kopie existieren.

pOS_SGObject

Auf folgende Struktur-Elemente darf LESEND zugegriffen werden:
ACHTUNG: Befindet man sich nicht in einer der Callbackfunktionen (Move_Func,Amin_Func...), so ist vor den Zugriff das System per pOS_SGOInfo() zu sperren !!!!

pOS_SGOInfo *sgo_Info;
Zeiger auf die Info, welcher das Objekt angehört.

UWORD sgo_Size;
Größe der gesamten Objektstruktur.

UWORD sgo_Width, sgo_Height;
Breite und Höhe des Objektes.

SLONG sgo_OldX, sgo_OldY;
Letzte Position.

SLONG sgo_X, sgo_Y;
Aktuelle Position. Innerhalb der Move_Func sind diese Werte auf die neue Position zu setzen.
Innerhalb der Kollisionsbehandlungs-Routinen dürfen diese Werte geändert werden.

ULONG sgo_HitID, sgo_HitMask;
Die HitID identifiziert ein Objekt. Die HitMask gibt an, für welche HitIDs die Kollisionsroutinen aufgerufen werden sollen.

Beispiel: obj1 HitID1=0000 HitMask1=0001
obj2 HitID2=0001 HitMask2=0110 (alle Angaben Binär)

Stoßen nun beide Objekte zusammen, so wird für obj1 die Kollisionsroutine 0 (SGOSTAG_CollHandler0) aufgerufen, da HitMask1 & HitID2 == 1 => sprich das nullte Bit ist gesetzt.

Die Kollisionsroutine wird immer für das Objekt aufgerufen, welches durch die HitMask angegeben hat, über Kollisionen mit

```
      anderen Objekten zu erfahren.

          CollRoutine(obj1,obj2);

      Objekt obj2 würde über diese Kollision nichts erfahren, da
      HitMask2 & HitID1 == 0 => kein Bit gesetzt.

      Sind nach der UND-Verknüpfung mehere Bits gesetzt, so werden
      ALLE Kollisionroutinen nacheinander (mit 0 beginnend) aufgerufen,
      für welche das entsprechende Bit 1 ist.

      Bei der Kollisionüberprüfung findet NUR ein Test auf die Bounds
      der Objekte statt. ES WERDEN KEINE MASKEN BEACHTET.

  ULONG sgo_Flags;
    Die Flags dürfen geändert werden.
    Der Zugriff ist mit pOS_LockSGOInfo() zu schützen.

      SGOF_MustDraw
        das Objekt MUSS beim nächsten Zeichendurchlauf unbedingt
        gezeichnet werden

      SGOF_AnimObj
        gestattet das Abspielen einer Animation. Nur wenn dieses Bit
        gesetzt ist, wird die Anim_Func aufgerufen bzw. die Animation
        automatisch abgespielt.
        Wird automatisch von pOS_CreateSGObject() gesetzt, wenn
        eine Animation erkannt wird.

      SGOF_UseMask
        gestattet das Benutzen der Stanzmaske. Nur wenn dieses Bit
        gesetzt wird eine mögliche Stanzmaske bzw. Maskenanimation
        auch beachtet.
        Wird automatisch von pOS_CreateSGObject() gesetzt, wenn
        eine Maske angegeben wird.

      SGOF_AnimMask
        gestattet das Abspielen einer MaskAnimation. Nur wenn dieses Bit
        gesetzt ist, wird die AnimMask_Func aufgerufen bzw. die Animation
        automatisch abgespielt.
        Wird automatisch von pOS_CreateSGObject() gesetzt, wenn
        eine Animation als Maske erkannt wird.

      SGOF_Delete
        Wenn dieses Bit gesetzt ist, wird das Objekt beim nächsten
        Zeichendurchlauf nicht mehr gezeichnet, sondern mit
        pOS_DeleteSGObject() gelöscht.
        Um ein Löschen auch von Bildschirm zu erzwingen, sollte ebenfalls
        SGOF_MustDraw gesetzt sein. Entspricht dem Verhalten von
        pOS_RemoveSGObject().

      SGOF_Inactivate
        Wenn dieses Bit gesetzt ist, wird das Objekt beim nächsten
        Zeichendurchlauf nicht mehr gezeichnet, sondern in die
        Inaktivliste überführt.
        Um ein Löschen auch von Bildschirm zu erzwingen, sollte ebenfalls
        SGOF_MustDraw gesetzt sein. Entspricht dem Verhalten von
```

```
            pOS_InactivateSGObject().

      ULONG sgo_FrameCount;
        Anzahl der Frames einer Animation.

      ULONG sgo_CurFrame;
        Aktuell angezeigter Frame.

      ULONG sgo_MaskCount;
        Anzahl der Frames einer MaskenAnimation.

      ULONG sgo_CurMask;
        Aktuell benutzer MaskenFrame.


    pOS_SGObject Callback Funktionen

      VOID (*sgo_Move_func)(_R_A0 pOS_SGObject*);
        Über diese Funktion kann ein Objekt bewegt werden. Dazu sind
        der neue X- und/oder Y-Wert zu setzen.

        INPUTS - das zu bewegende Objekt


      VOID (*sgo_BorCollision_func)(_R_A0 pOS_SGObject*,_R_A1 pOS_IBox*,_R_D0  ↩
         UWORD);
        Diese Funktion wird immer dann aufgerufen, wenn das Objekt mit der
        ClipBox kollidiert.

        INPUTS - das kollidierte Objekt
               - die ClipBox
               - Bitfeld, welches den Ort der Kollision anzeigt (UWORD)
                     SGOBCB_Left   - linker Rand
                     SGOBCB_Right  - rechter Rand
                     SGOBCB_Top    - oberer Rand
                     SGOBCB_Bottom - unterer Rand
                   Kombinationen sind ebenfalls möglich.


      ULONG (*sgo_AnimObj_func)(_R_A0 pOS_SGObject*);
      ULONG (*sgo_AnimMask_func)(_R_A0 pOS_SGObject*);
        Mit Hilfe dieser Funktionen kann der Ablauf der Objekt- bzw. Masken-
        Animation bestimmt werden. Es ist der nächste anzuzeigende Frame
        zurückzugeben.

        INPUTS - das zu animierende Objekt
```

1.3 sgos.library/pOS_ActivateSGObjectA()

```
NAME
     pOS_ActivateSGObjectA -- Aktiviert ein inaktives Objekt
     pOS_ActivateSGObject

SYNOPSIS
```

```
     pOS_ActivateSGObjectA( obj , taglist );
                          _R_A0   _R_A1

     VOID pOS_ActivateSGObjectA(pOS_SGObject*);

     pOS_ActivateSGObject( obj, tag,...);

     VOID pOS_ActivateSGObject(pOS_SGObject*,ULONG,...);

FUNCTION
     Das Objekt wird aus der Inaktiveliste in die Aktivliste über-
     nommen und beim nächsten pOS_SGOSDraw() gezeichnet.

INPUTS
     obj - das zu aktivierende Objekt. Dieses muß sich zuvor in der
           Inaktivliste der entsprechenden pOS_SGOInfo stehen.

TAGS
     SGOSTAG_InitialX (ULONG)
       (Default 0)
       X-Koordinate der Startposition des Objektes.

     SGOSTAG_InitialY (ULONG)
       (Default 0)
       Y-Koordinate der Startposition des Objektes.

     SGOSTAG_UserData (ULONG)
       (Default 0)
       Wert, welcher bei sgo_UserData[0] eingetragen wird.

SEE ALSO
     pOS_CreateSGObjectA(),pOS_CloneSGObjectA(),
     pOS_InactivateSGObject()
```

1.4 sgos.library/pOS_CloneSGObjectA()

```
NAME
     pOS_CloneSGObjectA -- Erzeugt eine Kopie eines Objektes
     pOS_CloneSGObject --

SYNOPSIS
     obj = pOS_CloneSGObjectA( obj , taglist);
     _R_D0                    _R_A0   _R_A1

     __ARID__ pOS_SGObject* pOS_CloneSGObjectA(pOS_SGObject*,
                              const pOS_TagItem*);

     obj = pOS_CloneSGObject( obj, tag,...);

     __ARID__ pOS_SGObject* pOS_CloneSGObject(pOS_SGObject*,
                              ULONG,...);

FUNCTION
     Das übergebende Objekt wird kopiert. Das neue Objekt benutzt die
```

```
     Grafikdaten des Orginalen mit. Deshalb darf das Orginal erst
     gelöscht werden, wenn keine Kopie mehr von ihm besteht.

INPUTS
     obj - das Objekt, welches kopiert werden soll
     taglist - Tagliste mit zusätzlichen Optionen

TAGS
     SGOSTAG_MoveFunc ((*)(_R_A0 pOS_SGObject*))
       (Default NULL)
       Zeiger auf Bewegungsfunktion für das Objekt. Diese Funktion
       wird von pOS_SGOSDraw() vor jedem Zeichnen aufgerufen.
       Innerhalb dieser Funktion dürfen der X- und der Y-Wert des
       Objektes geändert werden.
       Wird NULL übergeben, ist das Objekt feststehend.

     SGOSTAG_BorCollisionFunc ((*)(_R_A0 pOS_SGObject*,_R_A1 pOS_IBox,_R_D0  ↩
        UWORD))
       (Default NULL)
       Zeiger auf Funktion zur Auswertung einer Kollision des Objektes
       mit dem Rand.
       Innerhalb dieser Funktion dürfen der X- und der Y-Wert des
       Objektes geändert werden.

     SGOSTAG_AnimObjFunc ((*)(_R_A0 pOS_SGObject*))
       (Default NULL)
       Zeiger auf Funktion zur Ermittlung des nächsten Frames.
       Ist keine Funktion angegeben, so wird die Animation Bild für
       Bild abgespielt.

     SGOSTAG_AnimMaskFunc ((*)(_R_A0 pOS_SGObject*))
       (Default NULL)
       Zeiger auf Funktion zur Ermittlung des nächsten Frames der
       Masken-Animation.
       Ist keine Funktion angegeben, so wird die Animation Bild für
       Bild abgespielt.

     SGOSTAG_InitialX (ULONG)
       (Default 0)
       X-Koordinate der Startposition des neuen Objektes.

     SGOSTAG_InitialY (ULONG)
       (Default 0)
       Y-Koordinate der Startposition des neuen Objektes.

     SGOSTAG_HitID (ULONG)
       (Default 0)
       Bitmaske zur Ermittlung der Objekt-Kollisionsfunktion.

     SGOSTAG_HitMask (ULONG)
       (Default 0)
       Bitmaske zur Ermittlung der Objekt-Kollisionsfunktion.

     SGOSTAG_Inactive (BOOL)
       (Default FALSE)
       Bei TRUE wird das Objekt in die Inaktivliste der pOS_SGOInfo
       aufgenommen. Dieses Objekt wird dann nicht gezeichnet.
```

```
     SGOSTAG_UserData (ULONG)
       (Default 0)
       Wert, welcher bei sgo_UserData[0] eingetragen wird.

RESULT
     Zeiger auf das neue Objekt oder NULL im Fehlerfall.

SEE ALSO
     pOS_CreateSGOjectA()
```

1.5 sgos.library/pOS_CreateSGObjectA()

```
NAME
     pOS_CreateSGObjectA -- Erzeugen eines neuen Grafikobjektes
     pOS_CreateSGObject --

SYNOPSIS
     obj = pOS_CreateSGObjectA( info, taglist );
     _R_D0                     _R_A0  _R_A1

     __ARID__ pOS_SGObject* pOS_CreateSGObjectA(pOS_SGOInfo*,
                         const pOS_TagItem*);

     obj = pOS_CreateSGObject( info, tag,...);

     __ARID__ pOS_SGObject* pOS_CreateSGObject(pOS_SGOInfo*,
                         ULONG,...);

FUNCTION
     Erzeugt und initialisiert ein neues Grafikobjekt.

INPUTS
     info - zuvor erzeugtes pOS_SGOInfo
     taglist - Tagliste mit weiteren Optionen

TAGS
     SGOSTAG_SGObjSize (ULONG)
       (Default sizeof(pOS_SGObject))
       Die pOS_SGObject-Struktur kann mit privaten Daten erweitert
       werden. Dazu ist hier die Gesamtgröße der Struktur anzugeben.

     SGOSTAG_GfxMap (pOS_GfxMap*)
       (Default NULL)
       GfxMap, die das Grafikobjekt darstellt. Diese MUSS eine
       Friend-Map zum Window sein.

     SGOSTAG_StdGfxMap (pOS_GfxMap*)
       (Default NULL)
       Eine Standard-GfxMap, welche das Grafikobjekt darstellen soll.
       Diese wird automatisch zu einer Friend-Map umgewandelt.

     SGOSTAG_Mask (pOS_GfxMap*)
       (Default NULL)
```

```
  Zu verwendene Stanzmaske - MUSS Friend sein

SGOSTAG_GfxFile (const CHAR*)
  (Default NULL)
  Der Dateiname eines Bildes oder einer Animation. Der Name kann
  relativ zum Programm-Verzeichnis angegeben sein.
  Die Datei wird mit Hilfe des Picture.datatype geladen. Dabei
  wird automatisch erkannt, ob es sich um eine Animation oder
  um ein Einzelbild handelt.

SGOSTAG_MaskFile (const CHAR*)
  (Default NULL)
  Der Dateiname eines Bildes oder einer Animation. Der Name kann
  relativ zum Programm-Verzeichnis angegeben sein.
  Die Datei wird mit Hilfe des Picture.datatype geladen. Dabei
  wird automatisch erkannt, ob es sich um eine Animation oder
  um ein Einzelbild handelt.

SGOSTAG_MoveFunc ((*)(_R_A0 pOS_SGObject*))
  (Default NULL)
  Zeiger auf Bewegungsfunktion für das Objekt. Diese Funktion
  wird von pOS_SGOSDraw() vor jedem Zeichnen aufgerufen.
  Innerhalb dieser Funktion dürfen der X- und der Y-Wert des
  Objektes geändert werden.
  Wird NULL übergeben, ist das Objekt feststehend.

SGOSTAG_BorCollisionFunc ((*)(_R_A0 pOS_SGObject*,_R_A1 pOS_IBox,_R_D0  ↩
   UWORD))
  (Default NULL)
  Zeiger auf Funktion zur Auswertung einer Kollision des Objektes
  mit dem Rand.
  Innerhalb dieser Funktion dürfen der X- und der Y-Wert des
  Objektes geändert werden.

SGOSTAG_AnimObjFunc ((*)(_R_A0 pOS_SGObject*))
  (Default NULL)
  Zeiger auf Funktion zur Ermittlung des nächsten Frames.
  Ist keine Funktion angegeben, so wird die Animation Bild für
  Bild abgespielt.

SGOSTAG_AnimMaskFunc ((*)(_R_A0 pOS_SGObject*))
  (Default NULL)
  Zeiger auf Funktion zur Ermittlung des nächsten Frames der
  Masken-Animation.
  Ist keine Funktion angegeben, so wird die Animation Bild für
  Bild abgespielt.

SGOSTAG_InitialX (ULONG)
  (Default 0)
  X-Koordinate der Startposition des neuen Objektes.

SGOSTAG_InitialY (ULONG)
  (Default 0)
  Y-Koordinate der Startposition des neuen Objektes.

SGOSTAG_HitID (ULONG)
  (Default 0)
```

```
        Bitmaske zur Ermittlung der Objekt-Kollisionsfunktion.

      SGOSTAG_HitMask (ULONG)
        (Default 0)
        Bitmaske zur Ermittlung der Objekt-Kollisionsfunktion.

      SGOSTAG_Inactive (BOOL)
        (Default FALSE)
        Bei TRUE wird das Objekt in die Inaktivliste der pOS_SGOInfo
        aufgenommen. Dieses Objekt wird dann nicht gezeichnet.

      SGOSTAG_UserData (ULONG)
        (Default 0)
        Wert, welcher bei sgo_UserData[0] eingetragen wird.

RESULT
      obj - das neu erzeugt Objekt, oder NULL im Fehlerfall

SEE ALSO
```

1.6 sgos.library/pOS_CreateSGOInfoA()

```
NAME
      pOS_CreateSGOInfoA -- Erzeugt ein neues Objektsystem
      pOS_CreateSGOInfo --

SYNOPSIS
      info = pOS_CreateSGOInfoA(window,taglist);
      _R_D0                     _R_A0  _R_A1

      __ARID__ pOS_SGOInfo* pOS_CreateSGOInfoA(pOS_Window*,
                       const pOS_TagItem*);

      info = pOS_CreateSGOInfo(window,tag,...);

      __ARID__ pOS_SGOInfo* pOS_CreateSGOInfo(pOS_Window*,
                       ULONG,...);

FUNCTION
      Es wird ein neues Objektsystem erzeugt und initialisiert.
      Das Fenster MUSS ein GimmeZeroZero-Window (WINFLGF_GimmeZeroZero)
      sein und es MUSS eine SuperGfxMap (SCRTAG_SuperGfxMap) im Window
      installiert sein.

INPUTS
      window - Zeiger auf das zu verwendene Fenster
      taglist - Tagliste mit zusätzlichen Optionen
TAGS
      SGOSTAG_ClipBox (pOS_IBox*)
        (Default NULL)
        ClipBox, die den Border bestimmt. Bei NULL wird automatisch
        die Innenfläche des Fenster mit pOS_GetWindowIBox()
        bestimmt.
```

```
     SGOSTAG_CollHandler0 - 31 ((*)(_R_A0 pOS_SGObject*,_R_A1 pOS_SGObject*))
       (Default NULL)
       Funktion, die aufgerufen wird, wenn eine Kollision zwischen
       zwei bestimmten Objekten festgestellt wird.

     SGOSTAG_FramesPerSec (UWORD)
       (Default 25)
       Anzahl der Frames, die pro Sekunde gezeichnet werden sollen,
       nachdem das System mit pOS_StartSGOS gestartet wurde.

     SGOSTAG_Priority (SBYTE)
       (Default 0)
       Priorität mit welcher der BackgroundTask bei pOS_StartSGOS()
       gestartet wird.

RESULT
     info - Zeiger auf die initialisiert pOS_SGOInfo-Struktur. Diese
            MUSS mit pOS_DeleteSGOInfo() wieder freigegeben werden.

SEE ALSO
     pOS_CreateSGOInfo()
```

1.7 sgos.library/pOS_DeleteSGObject()

```
NAME
     pOS_DeleteSGObject -- Entfernt ein Objekt aus dem Speicher

SYNOPSIS
     pOS_DeleteSGObject( obj );
                         _R_A0

     VOID pOS_DeleteSGOInfo(pOS_SGObject*);

FUNCTION
     Alle von pOS_CreateSGObject() erzeugten Daten werden freigegeben.
     ACHTUNG: Falls das Objekt zum Zeitpunkt des Entfernens sichtbar
              ist, MUSS pOS_RemoveSGObject() benutzt werden, damit
              es auch aus dem Window entfernt wird.


INPUTS
     obj - Das mit pOS_CreateSGObject() bzw. pOS_CloneSGObject()
           erzeugte Grafikobjekt.

SEE ALSO
     pOS_CreateSGObject(),pOS_CloneSGObject()
```

1.8 sgos.library/pOS_DeleteSGOInfo()

```
NAME
```

```
     pOS_DeleteSGOInfo -- Löschen der gesamten Grafik-Objekt-Systems

SYNOPSIS
     pOS_DeleteSGOInfo( info );
                        _R_A0

     VOID pOS_DeleteSGOInfo(pOS_SGOInfo*);

FUNCTION
     Es werden alle allokierten Resourcen freigegeben. Dazu gehören
     auch Grafikobjekt, die sich noch in der Aktiv- bzw. Inaktivliste
     befinden.

INPUTS
     info - Zeiger auf Info-Struktur

SEE ALSO
     pOS_CreateSGOInfo()
```

1.9 sgos.library/pOS_DrawSGOS()

```
NAME
     pOS_DrawSGOS -- Zeichnen aller Grafikobjekte des Systems

SYNOPSIS
     pOS_DrawSGOS( info, mode );
                   _R_A0  _R_D0

     VOID pOS_DrawSGOS(pOS_SGOInfo*,ULONG);

FUNCTION
     Alle Grafikobjekt im System werden gezeichnet.

INPUTS
     info - Zeiger auf Info-Struktur
     mode - Mode mit welchem gezeichnet werden soll (pOS_SGOSDrawModes)

        SGOSDM_Normal  - alle aktiven Objekte des Systems werden
                         bewegt und neu gezeichnet

        SGOSDM_Refresh - alle aktiven Objekte werden in ihrem
                         derzeitigem Zustand neu gezeichnet. Es wird
                         also keine Bewegungs- bzw. Animationsfunktion
                         aufgerufen.

SEE ALSO
     pOS_StartSGOS(),pOS_StopSGOS()
```

1.10 sgos.library/pOS_InactivateSGObject()

```
NAME
     pOS_InactivateSGObject -- Inaktiviert ein aktives Objekt

SYNOPSIS
     pOS_InactivateSGObject( obj );
                             _R_A0

     VOID pOS_InactivateSGObject(pOS_SGObject*);

FUNCTION
     Das Grafikobjekt wird aus der Aktivliste in die Inaktivliste
     überführt.
     Es wird auch aus dem sichtbaren Bereich entfernt.

INPUTS
     obj - das zu inaktivierende Objekt

SEE ALSO
     pOS_ActivateSGObject()
```

1.11 sgos.library/pOS_LockSGOInfo()

```
NAME
     pOS_LockSGOInfo -- Sperrt das laufende System

SYNOPSIS
     pOS_LockSGOInfo( info );
                      _R_A0

     VOID pOS_LockSGOInfo(pOS_SGOInfo*);

FUNCTION
     Sperrt das laufende System, so daß auf die Objekt zugegriffen
     werden kann (z.B. Objektflags ändern).

INPUTS
     info - Zeiger auf zu sperrende pOS_SGOInfo

SEE ALSO
     pOS_UnlockSGOInfo()
```

1.12 sgos.library/pOS_RemoveSGObject()

```
NAME
     pOS_RemoveSGObject -- das aktive Objekt wird aus dem System
                           entfernt

SYNOPSIS
```

```
     pOS_RemoveSGObject( obj );
                       _R_A0

     VOID pOS_RemoveSGObject(pOS_SGObject*);

FUNCTION
     Setzt das SGOSF_Delete. Beim nächsten Zeichnen wird das Objekt
     aus dem sichtbaren Bereich entfernt und mit pOS_DeleteSGObject()
     gelöscht.

INPUTS
     obj - zu entfernendes Objekt

SEE ALSO
     pOS_DeleteSGObject()
```

1.13 sgos.library/pOS_SetSGObjectGfxMap()

```
NAME
     pOS_SetSGObjectGfxMap -- Setzt neue GfxMap

SYNOPSIS
     oldMap = pOS_SetSGObjectGfxMap( obj , newMap );
     _R_D0                           _R_A0  _R_A1

     pOS_GfxMap* pOS_SetSGObjectGfxMap(pOS_SGObject*,pOS_GfxMap*);

FUNCTION
     Setzt neue GfxMap. Beim nächsten Zeichendurchlauf wird diese
     Map verwendet.

     ACHTUNG: Die alte GfxMap MUSS mit SGOSTAG_GfxMap übergeben worden
              sein. Es kann nur eine GfxMap ausgetauscht werden,
              die auch zuvor von außen gesetzt wurde.
              Die neue GfxMap MUSS ebenfalls eine FRIEND-Map sein.

INPUTS
     obj    - das Objekt
     newMap - Zeiger auf neue, FRIEND GfxMap

RESULT
     oldMap - die alte GfxMap

SEE ALSO
     pOS_SetSGObjectMask()
```

1.14 sgos.library/pOS_SetSGObjectMask()

```
NAME
     pOS_SetSGObjectMask -- Setzt neue Stanzmaske
```

```
SYNOPSIS
     oldMask = pOS_SetSGObjectMask( obj , newMask );
     _R_D0                          _R_A0  _R_A1

     pOS_GfxMap* pOS_SetSGObjectMask(pOS_SGObject*,pOS_GfxMap*);

FUNCTION
     Setzt neue Masken-GfxMap. Beim nächsten Zeichendurchlauf wird
     diese Map verwendet.

     ACHTUNG: Die alte Maske MUSS per SGOSTAG_Mask übergeben worden
              sein. Es kann nur eine Maske ausgetauscht werden,
              die auch zuvor von außen gesetzt wurde.
              Die neue Maske MUSS ebenfalls eine FRIEND-Map sein.

INPUTS
     obj     - das Objekt
     newMask - Zeiger auf neue, FRIEND Masken-GfxMap

RESULT
     oldMask - die alte Masken-GfxMap

SEE ALSO
     pOS_SetSGObjectGfxMap()
```

1.15 sgos.library/pOS_StartSGOS()

```
NAME
     pOS_StartSGOS -- Startet zyklisches Zeichnen

SYNOPSIS
     success = pOS_StartSGOS( info );
                              _R_A0

     BOOL pOS_StartSGOS(pOS_SGOInfo*);

FUNCTION
     pOS_StartSGOS() installiert einen BackgroundTask, welcher
     SGOSTAG_FramesPerSec mal pro Sekunde pOS_SGOSDraw() aufruft.

INPUTS
     info - Zeiger auf zu startende pOS_SGOInfo

RESULT
     success - TRUE, wenn der BachgroundTask erzeugt werden konnte
               FALSE in Fehlerfall

SEE ALSO
     pOS_StopSGOS()
```

1.16 sgos.library/pOS_StopSGOS()

```
NAME
     pOS_StopSGOS -- Beendet BackgroundZeichenTask

SYNOPSIS
     pOS_StopSGOS( info );
                   _R_A0

     VOID pOS_StopSGOS(pOS_SGOInfo*);

FUNCTION
     Beendet den mit pOS_StartSGOS() installierten BackgroundTask.

INPUTS
     info - Zeiger auf zu stopende pOS_SGOInfo

SEE ALSO
     pOS_StartSGOS()
```

1.17 sgos.library/pOS_UnlockSGOInfo()

```
NAME
     pOS_UnlockSGOInfo -- Entsperren des Systems

SYNOPSIS
     pOS_UnlockSGOInfo( info );
                        _R_A0

     VOID pOS_UnlockSGOInfo(_R_A0 pOS_SGOInfo*);

FUNCTION
     Löst die mit pOS_LockSGOInfo() gesetzte Sperre.

INPUTS
     info - zu entsperrende pOS_SGOInfo

SEE ALSO
     pOS_LockSGOInfo()
```