

# Core Library Functions

The core library contains functions which are core to almost every example in the text.	
InitialiseScreen(w, h, t\$, col, r)	Creates the main app window. Size: <i>w</i> x <i>h</i> . Title: <i>t\$</i> . Background colour: <i>col</i> , Orientations allowed: <i>r</i>
ShowSplashScreen(f\$)	Fills screen with image <i>f\$</i> .
HideSplashScreen(s)	Hides splash screen image after <i>s</i> secs or mouse press.

# SpriteLine Library Functions

The sprite line library contains function to draw lines and basic outlines (rectangle, circle and polygon) using sprites.

Line	
int DrawSpriteLine(x1#,y1#,x2#,y2#,th#,col,op)	Creates a line between (x1#,y1#) and (x2#,y2#). Thick: th#. Colour: col. Opacity: op. Returns ID of line.
RedrawSpriteLine(id,x1#,y1#,x2#,y2#,th#,col,op)	Redraws existing line, <i>id</i> , with new values.
DeleteSpriteLine(id)	Deletes line <i>id</i> .

Box	
int DrawSpriteBox(x1#,y1#,x2#,y2#,th#,col,op)	Creates box outline. Top-Left:(x1#,y1#). Bottom-right: (x2#,y2#). Thick: th#. Colour: col. Opacity: op. Returns ID of box.
RedrawSpriteBox(id,x1#,y1#,x2#,y2#,th#,col,op)	Redraws existing box, <i>id</i> , with new values.
DeleteSpriteBox(id)	Deletes box <i>id</i> .

Circle	
int DrawSpriteCircle(x#,y#,rad#,th#,col,op)	Creates circle outline. Centre:(x#,y#). Radius:rad#. Thick: th#. Colour: col. Opacity:op. Returns ID of circle.
RedrawSpriteCircle(id,x1#,y1#,x2#,y2#,th#,col,op)	Redraws existing circle, <i>id</i> , with new values.
DeleteSpriteCircle(id)	Deletes circle <i>id</i> .

Polygon	
int DrawSpritePolygon(pnts#[],th#,col,op)	Creates polygon outline. Coords:pnts#[] (x,y,x,y, etc.). Thickness: th#. Colour: col. Opacity: op. Returns ID of polygon.
RedrawSpriteBox(id,pnts#[],th#,col,op)	Redraws existing polygon, <i>id</i> , with new values.
DeleteSpriteBox(id, num)	Deletes polygon <i>id</i> containing <i>num</i> edges.

Bezier Curve	
int CreateBCurve((sx#,sy#,ex#,ey#,cx#,cy#)	Creates a Bezier curve and returns its ID. Start (sx#,sy#) End (ex#,ey#). Control (cx#,cy#) Defaults black, 0.25 thick, 20 segments.
SetBCurveControl(id,cx#,cy#)	Moves control point of B curve id to (cx#,cy#).
SetBCurveStart(id,sx#,sy#)	Moves start point of B curve id to (sx#,sy#).
SetBCurveEnd(id,ex#,ey#)	Moves end point of B curve id to (ex#,ey#)
SetBCurveColour(id,col)	Sets colour of B curve id to col.
SetBCurveThickness(id, th)	Sets thickness of B curve id's lines to th.
SetBCurveSegments(id, num)	Sets number of lines used to draw B curve id to num
DrawBCurve(id)	Draws B curve id

# GUI Library Functions

The GUI library allows the creation of some basic GUI elements such as buttons, checkboxes, radio buttons, dialog boxes, popup menus and frames. It also has an option to create a number pad for numeric data entry.

GUIButton	
int CreateGUIButton(x#,y#,w#,h#, g\$, t\$)	Creates button (dim <i>w#</i> x <i>h#</i> ) at ( <i>x#</i> , <i>y#</i> ) ,img <i>g\$</i> , txt <i>t\$</i> . Returns id of button. Deletes button <i>id</i> .
DeleteGUIButton(id)	Returns 1 if <i>id</i> pressed. Makes button reacts to user.
int HandleGUIButton(id)	Places button on depth <i>ly</i> . Returns 1 if okay.
int SetGUIButtonDepth(id, ly)	Sets button position to ( <i>x#</i> , <i>y#</i> ). Returns 1 if okay.
int SetGUIButtonPosition(id, x#, y#)	Sets button <i>id</i> to size to <i>w#</i> by <i>h#</i> . Returns 1 if okay.
int SetGUIButtonSize(id, w#, h#)	

GUIDialogBox	
int CreateGUIDialogBox(x#, y#, w#, h#, g\$, t\$, bg\$, bt\$)	Creates a dialog box <i>w#</i> x <i>h#</i> , at ( <i>x#</i> , <i>y#</i> ),box framed by image <i>g\$</i> and title <i>t\$</i> . Button images <i>bg\$</i> (! separated) showing <i>bt\$</i> (! separated). Returns id of dialog box.
HandleGUIDialogBox()	Returns no. of button (not id) pressed. Deletes dialog box.
int SetGUIDialogBoxButtonPosition(n, x#, y#)	Repositions button <i>n</i> to ( <i>x#</i> , <i>y#</i> ). Returns 1 if okay.
int SetGUIDialBoxButtonSize(n, w#,h#)	Resizes button <i>n</i> to <i>w#</i> by <i>h#</i> . Returns 1 if okay.

GUICheckbox	
int CreateGUICheckbox(x#, y#, g\$, t\$)	Positions checkbox at ( <i>x#</i> , <i>y#</i> ). Shows image <i>g\$</i> and text <i>t\$</i> . Returns id assigned. Deletes checkbox <i>id</i> .
DeleteCheckbox(id)	Returns checkbox <i>id</i> 's current frame (1/2).
int GetGUICheckboxState(id)	Returns frame shown by checkbox <i>id</i> (1/2). Makes checkbox reacts to user clicks.
int HandleGUICheckbox(id)	Changes checkbox <i>id</i> 's text colour to <i>col</i> .
SetGUICheckboxTextColor(id, col)	Places checkbox <i>id</i> at ( <i>x#</i> , <i>y#</i> ).
SetGUICheckboxPosition(id, x#, y#)	Changes checkbox <i>id</i> 's text size to <i>sz#</i> .
SetGUICheckboxTextSize(id, sz#)	

GUIRadioButton	
int CreateGUIRadioButton(x#,y#,g\$,t\$,gp)	Positions radio button at ( <i>x#</i> , <i>y#</i> ). Shows image <i>g\$</i> & text <i>t\$</i> . Belongs to group <i>gp</i> . Returns id assigned.
DeleteGUIRadioButtonGroup(gp)	Deletes all buttons in group <i>gp</i> .
int GetGUIRadioButtonSelectedInGroup(gp)	Returns no. of selected button in group (1,2,3 etc.).
HandleGUIRadioButtonGroup(gp)	Selects/deselects when clicked. Returns current frame (1/2)
SetGUIRadioButtonTextColor(id,col)	Sets text colour of button <i>id</i> to <i>col</i> .
int SetGUIRadioButtonDepth(gp, ly)	Sets depth of all buttons in group <i>gp</i> to <i>ly</i> . Returns 1 if okay.

SetGUIRadioButtonPosition(id, x#, y#)	Places button <i>id</i> at ( <i>x#</i> , <i>y#</i> ).
SetGUIRadioButtonTextSize(id,sz#)	Sets text size of button <i>id</i> to <i>sz#</i> .

GUIFrame	
int CreateGUIFrame(x#,y#,w#,h#,g\$)	Creates frame (dim <i>w#</i> x <i>h#</i> ) at ( <i>x#</i> , <i>y#</i> ) filled with image <i>g\$</i> . Returns frame id.
int AddButtonToGUIFrame(frm,x#,y#,w#,h#,g\$,t\$)	Creates a button in frame <i>frm</i> at ( <i>x#</i> , <i>y#</i> ). size:( <i>w#</i> x <i>h#</i> ); image: <i>g\$</i> ; text: <i>t\$</i> . Returns button's frame index.
int AddCheckboxToGUIFrame(frm,x#,y#,g\$,t\$)	Creates a checkbox in frame <i>frm</i> at ( <i>x#</i> , <i>y#</i> ). image: <i>g\$</i> ; text: <i>t\$</i> . Returns checkbox's frame index.
int AddEditboxToGUIFrame(frm, x#, y#, w#, h#)	Creates an edit box in frame <i>frm</i> at ( <i>x#</i> , <i>y#</i> ). Size: <i>w#</i> x <i>h#</i> . Returns edit box's frame index.
int AddRadioButtonToGUIFrame(frm,x#,y#,g\$,t\$,gp)	Creates a radio button in frame <i>frm</i> at ( <i>x#</i> , <i>y#</i> ). image: <i>g\$</i> ; text: <i>t\$</i> . In group <i>gp</i> . Returns radio button's frame index.
int AddSpriteToGUIFrame(frm,x#,y#,w#,h#,g\$)	Creates a sprite in frame <i>frm</i> , dim: <i>w#</i> x <i>h#</i> ,at ( <i>x#</i> , <i>y#</i> ). Returns created sprite's frame index.
int AddTextToGUIFrame(frm,x#,y#,sz#,t\$,col)	Creates a text in frame <i>frm</i> at ( <i>x#</i> , <i>y#</i> ). size:sz#; colour:col text: <i>t\$</i> . Returns text's frame index.
int GetGUIFrameElementDetails(frm,idx)	Returns details of element <i>idx</i> in <i>frm</i> . (element type *100000 + true id)
int DeleteGUIFrame(frm)	Deletes frame <i>frm</i> . Returns 1 if okay.
int HandleGUIFrame(frm)	Returns frame index of any frame element clicked by user.
int SetGUIFrameDepth(frm,ly)	Places frame on depth <i>ly</i> . Returns 1 if okay.
int SetGUIFramePosition(frm,x#,y#)	Positions frame <i>frm</i> at ( <i>x#</i> , <i>y#</i> ). Returns 1 if okay.

GUIPopUpMenu	
int CreateGUIPopUpMenu(x#,y#,w#,h#,fg\$,bg\$,ops\$)	Creates a popup menu (dim <i>w#</i> x <i>h#</i> ) at ( <i>x#</i> , <i>y#</i> ) ; frame image:fg\$, btn image: <i>bg\$</i> . Menu options: <i>ops\$</i> (! separated).
DeleteGUIPopUpMenu()	Deletes the menu.
int HandleGUIPopUpMenu()	Returns the number of the option selected (1,2,3,etc.).

GUINumberPad	
int CreateGUINumberPad(x#,y#,w#,h#,fg\$,bg\$,del)	Creates a number pad (dim <i>w#</i> x <i>h#</i> ) at ( <i>x#</i> , <i>y#</i> ) ; frame image:fg\$, btn image: <i>bg\$</i> . Delete after: <i>del</i> (1 = delete)
int HandleGUINumberpad()	Accepts key presses. Displays value entered. When Enter pressed, delete number pad or reset its display to zero. Returns value entered.
DeleteGUINumberPad()	Deletes the number pad.
MoveGUINumberPadText(x#,y#)	Moves pad's display to ( <i>x#</i> , <i>y#</i> ) within pad.
ResizeGUINumberPadText(sz#)	Resizes display text to <i>sz#</i> .

# List Library Functions

A List is a data structure designed to contain integers which represent the IDs of memblocks. The format of the memblocks themselves needs to be defined within each new app as does the access to the fields within those memblocks.

The list operations are designed to manipulate the integer values within the core List data structure. Parameters marked by an asterisk (*ref* parameters) are modified by the function.

When using a List which references memblocks, start by defining *DataType* giving the fields that need to be stored in the memblock. This is your record structure.

Write a *RecordToMemblock()* function which takes a *DataType* parameter and stores its contents in a memblock and returns the ID of that memblock (created by the function). It is this ID that should be stored in the List structure. Write other functions as required. See AliceList example in book.

CreateList(*list, sz, fx)	Creates an empty list ( <i>list</i> ) containing <i>sz</i> elements. May be of a fixed size ( <i>fx</i> = 1) or may expand as required ( <i>fx</i> = 0).
AddToList(*list, v)	Adds <i>v</i> to end of <i>list</i> .
DeleteFromList(*list, p)	Deletes value at position <i>p</i> in <i>list</i> .
DeleteList(*list)	Deletes the contents of <i>list</i> .
int FindInList(list, v)	Returns the position of <i>v</i> in <i>list</i> (-1 if not found).
int GetFromList(list, p)	Returns the value at position <i>p</i> in <i>list</i> (-1 if invalid <i>p</i> ).
InsertInList(*list, v, p)	Inserts <i>v</i> at position <i>p</i> in <i>list</i> ( <i>p</i> starts at 1).
int IsEmptyList(list)	Returns 1 if list empty, else zero.
int IsFullList(list)	Returns 1 if list full, else zero.
int LengthOfList(list)	Returns the number of entries in <i>list</i> .
str ToStringList(list)	Returns a string contain every value in <i>list</i> (  separated).

# Date Library Functions

These are a collection of functions which may be used when manipulating dates.

int CalcDayOfWeek(d,m,y)	Returns the day of the week <i>d/m/y</i> falls on (0=Sunday).
int DateToJDN(d,m,y)	Returns number of days between <i>d/m/y</i> and 1/1/4713BC.
str JDNToDate(jdn)	Returns date equivalent of <i>jdn</i> as string in format dd/mm/yyyy.
int DaysBetween(d1,m1,y1,d2,m2,y2)	Returns the number of days between <i>d1/m1/y1</i> and <i>d2/m2/y2</i> .
str AddDays(d,m,y,dys)	Returns a string giving date of <i>d/m/y</i> + <i>dys</i> days.

For

HANDS ON AGK2 BASIC

Volume 2

# User Defined Library Functions