

## SDL\_bgi functions and macros

This is a list of functions and macros provided by SDL\_bgi.

### Standard BGI

```
void arc (int x, int y, int stangle, int endangle, int radius);
void bar (int left, int top, int right, int bottom);
void bar3d (int left, int top, int right, int bottom, int depth, int topflag);
void circle (int x, int y, int radius);
void cleardevice (void);
void clearviewport (void);
void closegraph (void);
void detectgraph (int *graphdriver, int *graphmode);
void drawpoly (int numpoints, int *polypoints);
void ellipse (int x, int y, int stangle, int endangle, int xradius, int yradius);
void fillellipse (int x, int y, int xradius, int yradius);
void fillpoly (int numpoints, int *polypoints);
void floodfill (int x, int y, int border);
int getactivepage (void);
void getarccoords (struct arccoordstype *arccoords);
void getaspectratio (int *xasp, int *yasp);
int getbkcolor (void);
int getcolor (void);
struct palettetype* getdefaultpalette (void);
char* getdrivername (void);
void getfillpattern (char *pattern);
void getfillsettings (struct fillsettingstype *fillinfo);
int getgraphmode (void);
void getimage (int left, int top, int right, int bottom, void *bitmap);
void getlinesettings (struct linesettingstype *lineinfo);
int getmaxcolor (void);
int getmaxmode (void);
```

```

int getmaxx (void);
int getmaxy (void);
char* getmodename (int mode_number);
void getmoderange (int graphdriver, int *lomode, int *himode);
void getpalette (struct palettetype *palette);
int getpalettesize (void);
unsigned int getpixel (int x, int y);
void gettextsettings (struct textsettingstype *texttypeinfo);
void getviewsettings (struct viewporttype *viewport);
int getvisualpage (void);
int getx (void);
int gety (void);
void graphdefaults (void);
char* grapherrormsg (int errorcode);
int graphresult(void);
unsigned imagesize (int left, int top, int right, int bottom);
void initgraph (int *graphdriver, int *graphmode, char *pathtodriver);
int installuserdriver (char *name, int huge (*detect)(void));
int installuserfont (char *name);
void line (int x1, int y1, int x2, int y2);
void linerel (int dx, int dy);
void lineto (int x, int y);
void moverel (int dx, int dy);
void moveto (int x, int y);
void outtext (char *textstring);
void outtextxy (int x, int y, char *textstring);
void pieslice (int x, int y, int stangle, int endangle, int radius);
void putimage (int left, int top, void *bitmap, int op);
void putpixel (int x, int y, int color);
void rectangle (int left, int top, int right, int bottom);
int registerbgidriver (void (*driver)(void));

```

```

int registerbgifont (void (*font)(void));
void restorecrtmode (void);
void sector (int x, int y, int stangle, int endangle, int xradius, int yradius);
void setactivepage (int page);
void setallpalette (struct palettetype *palette);
void setaspectratio (int xasp, int yasp);
void setbkcolor (int color);
void setcolor (int color);
void setfillpattern (char *upattern, int color);
void setfillstyle (int pattern, int color);
unsigned setgraphbufsize (unsigned bufsize);
void setgraphmode (int mode);
void setlinestyle (int linestyle, unsigned upattern, int thickness);
void setpalette (int colnum, int color);
void settextjustify (int horiz, int vert);
void settextstyle (int font, int direction, int charsize);
void setusercharsize (int multx, int divx, int multy, int divy);
void setviewport (int left, int top, int right, int bottom, int clip);
void setvisualpage (int page);
void setwritemode (int mode);
int textheight (char *textstring);
int textwidth (char *textstring);

```

### **SDL\_bgi extensions**

```

int ALPHA_VALUE (int color);
int BLUE_VALUE (int color);
int COLOR (int r, int g, int b);
int COLOR32(Uint32 color);
Uint32 colorRGB (int r, int g, int b) (macro)
int GREEN_VALUE(int color);
int IS_BGI_COLOR (int color);

```

```

int IS_RGB_COLOR(int color);
int RED_VALUE (int color);
void _putpixel (int x, int y);
void closewindow (int id);
int event (void);
int eventtype (void);
void getbuffer (Uint32 *buffer);
int getcurrentwindow (void);
int getevent (void);
void getleftclick (void);
void getlinebuffer (int y, Uint32 *linebuffer);
int getmaxheight (void);
int getmaxwidth (void);
void getmiddleclick (void);
void getmouseclick (int kind, int *x, int *y);
void getrightclick (void);
void getscreensize (int x, int y);
int getwindowheight (void) (macro)
int getwindowwidth (void) (macro)
void initpalette (void);
int initwindow (int width, int height);
int ismouseclick (int kind);
int mouseclick (void);
int mousex (void);
int mousey (void);
void putbuffer (Uint32 *buffer);
void putlinebuffer (int y, Uint32 *buffer);
void readimagefile (char *filename, int x1, int y1, int x2, int y2);
void refresh (void);
void resetwinoptions (int id, char *title, int x, int y);
int resizepalette (Uint32);

```

```

void sdlbgiauto (void);
void sdlbgifast (void);
void sdlbgislow (void);
void setalpha (int col, Uint8 alpha);
void setbkgbcolor (int color);
void setblendmode (int blendmode);
void setcurrentwindow (int id);
void setrgbcolor (int color);
void setrgbpalette (int colnum, int red, int green, int blue);
void setwinoptions (char *title, int x, int y, Uint32 flags);
void setwintitle (int id, char *title);
void showerrorbox (const char *message);
void showinfobox (const char *message);
void swapbuffers (void);
void writeimagefile (char *filename, int left, int top, int right, int bottom);

```

## **Non graphics functions**

```

void delay (int millisec) // DOS.H
int edelay (int msec);
int getch (void); // CONIO.H
int kbhit (void); (macro) // CONIO.H
int random (int range) (macro) // STDLIB.H
int xkbhit (void);

```