

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 bar3d (int left, int top, int right, int bottom, int depth, int topflag);
void bar (int left, int top, int right, int bottom);
int BLUE_VALUE (int color);
void circle (int x, int y, int radius);
void cleardevice (void);
void clearviewport (void);
void closegraph (void);
void delay(int millisec)
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 getch (void);
int getcolor (void);
struct palettetype* getdefaultpalette (void);
char* getdrivername (void);
void getfillpattern (char *pattern);
void getfillsettings (struct fillsettingstype *fillinfo);
void getimage (int left, int top, int right, int bottom, void *bitmap);
```

```

int getgraphmode (void);
void getlinesettings (struct linesettingstype *lineinfo);
int getmaxcolor (void);
int getmaxmode (void);
int getmaxxx (void);
int getmaxy (void);
char* getmodename (int mode__number);
void getmoderange (int graphdriver, int lmode, int himode);
void getpalette (struct palettetype *palette);
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);
int kbhit (void);
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, int, void *, int);
void putpixel (int x, int y, int color);
int random (int range) (macro)
void readimagefile (char *filename, int left, int top, int right, int bottom);
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 setbkrgbcolor (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);
void swapbuffers (void);
int textheight (char *textstring);
int textwidth (char *textstring);
void writeimagefile (char *filename, int left, int top, int right, int bottom);

```

```
int xkbhit (void);
```

SDL_bgi extensions

```
void sdlbgifast (void);
```

```
void sdlbgislow (void);
```

```
void closewindow (int);
```

```
int COLOR(int r, int g, int b);
```

```
int event (void);
```

```
int eventtype (void);
```

```
int getcurrentwindow (void);
```

```
int getevent (void);
```

```
void getmouseclick (int kind, int x, int y);
```

```
int getpalettesize (void);
```

```
int GREEN_VALUE(int color);
```

```
int initwindow (int width, int height);
```

```
int IS_BGI_COLOR (int color);
```

```
int ismouseclick (int kind);
```

```
int IS_RGB_COLOR(int color);
```

```
int mouseclick(void);
```

```
int mousex (void);
```

```
int mousey (void);
```

```
void __putpixel (int x, int y);
```

```
int RED_VALUE (int color);
```

```
void refresh (void);
```

```
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);
```