

Documentació LS_Allegro

Departament d'Informàtica (La Salle Barcelona)

Darrera Revisió : 27/12/2014

Índex

1	Introducció	1
2	Constants	1
2.1	Constants de Colors	1
2.2	Constants de mida de lletra	2
3	Funcions i procediments del mòdul LS_allegro	3
3.1	Procediments	3
3.1.1	void LS_allegro_exit();	3
3.1.2	void LS_allegro_console_fflush();	3
3.1.3	void LS_allegro_console_clear_screen();	4
3.1.4	void LS_allegro_clear_and_paint(ALLEGRO_COLOR color);	4
3.2	Funcions	5
3.2.1	int LS_allegro_init(int nWidth, int nHeight, char *sWindowName);	5
3.2.2	int LS_allegro_key_pressed(int nKey)	6
3.2.3	ALLEGRO_COLOR LS_allegro_get_color(int nColor)	9
3.2.4	ALLEGRO_FONT* LS_allegro_get_font(int nSize)	9
4	Funcions i procediments pròpies d'Allegro	10
4.1	Procediments	10
4.1.1	void al_draw_line(float x1, float y1, float x2, float y2, ALLEGRO_COLOR color, float thickness)	10
4.1.2	void al_draw_triangle(float x1, float y1, float x2, float y2, float x3, float y3, ALLEGRO_COLOR color, float thickness)	11
4.1.3	void al_draw_filled_triangle(float x1, float y1, float x2, float y2, float x3, float y3, ALLEGRO_COLOR color)	12
4.1.4	void al_draw_rectangle(float x1, float y1, float x2, float y2, ALLEGRO_COLOR color, float thickness)	13
4.1.5	void al_draw_filled_rectangle(float x1, float y1, float x2, float y2, ALLEGRO_COLOR color)	14
4.1.6	void al_draw_circle(float cx, float cy, float r, ALLEGRO_COLOR color, float thickness)	15
4.1.7	void al_draw_filled_circle(float cx, float cy, float r, ALLEGRO_COLOR color)	16
4.1.8	void al_draw_ellipse(float cx, float cy, float rx, float ry, ALLEGRO_COLOR color, float thickness)	17
4.1.9	void al_draw_filled_ellipse(float cx, float cy, float rx, float ry, ALLEGRO_COLOR color)	18

1 Introducció

El mòdul LS_Allegro es va crear amb la finalitat de simplificar l'ús de la llibreria Allegro 5 als alumnes de Programació I. La necessitat d'aquesta simplificació va sorgir quan, en la migració entre la versió 4.4 i la versió 5, els desenvolupadors de la llibreria van dur a terme molts canvis per tal de convertir-la en una eina més potent per al desenvolupament de videojocs. No obstant això, el guany en potencia de desenvolupament va anar en detriment de la senzillesa d'ús fent que, a partir de la versió 5, les operacions bàsiques requerissin de coneixements més avançats de programació que en versions anteriors.

La idea de LS_Allegro és ser un mòdul que funcioni d'adaptador entre el codi dels alumnes i la llibreria Allegro 5. D'aquesta manera, aquest seguit de crides més complexes que necessita Allegro queden amagades dins del mòdul i l'alumne se'n pot abstraure tot centrant-se en el desenvolupament de la pràctica.

2 Constants

En aquesta secció s'expliquen les diverses constants que estan definides dins del mòdul LS_Allegro per tal de que es pugui entendre el que signifiquen i saber quan es poden utilitzar.

2.1 Constants de Colors

La llibreria Allegro 5 permet generar colors a partir de combinacions RGB mitjançant la funció **al_map_rgb(R,G,B)** que retorna una variable del tipus ALLEGRO_COLOR que permet pintar les figures d'Allegro amb el color corresponent a la combinació RGB rebuda per paràmetre.

Això permet que la combinació de colors de la que disposem per a les nostres creacions sigui gairebé infinita. No obstant això, es va voler simplificar l'accés als colors per evitar que els alumnes de primer haguessin de treballar amb les combinacions RGB.

Per fer-ho es va definir una paleta de colors estàndard que contingués els colors més habituals. D'aquesta manera, de cara als alumnes de primer, els únics colors disponibles per a les seves creacions serien els de dita paleta i la manera d'accedir-hi seria mitjançant la crida a la funció **LS_allegro_get_color(CONSTANT_DE_COLOR)** que s'explicarà amb més detall al llarg d'aquest document.

A continuació es llisten els colors que es troben dins de la paleta de colors estàndard i que poden ser accedits a través de crides a la funció **LS_allegro_get_color(CONSTANT_DE_COLOR)**:

Constant	Crida
Negre	LS_allegro_get_color(BLACK);
Blau Fosc	LS_allegro_get_color(DARK_BLUE);
Verd Fosc	LS_allegro_get_color(DARK_GREEN);
Cian	LS_allegro_get_color(CYAN);
Vermell	LS_allegro_get_color(RED);
Rosa	LS_allegro_get_color(PINK);
Taronja	LS_allegro_get_color(ORANGE);
Gris Clar	LS_allegro_get_color(LIGHT_GRAY);
Gris	LS_allegro_get_color(GRAY);
Blau	LS_allegro_get_color(BLUE);
Verd Clar	LS_allegro_get_color(LIGHT_GREEN);
Blau Cel	LS_allegro_get_color(LIGHT_BLUE);
Granat	LS_allegro_get_color(GARNET);
Rosa Clar	LS_allegro_get_color(LIGHT_PINK);
Groc	LS_allegro_get_color(YELLOW);
Blanc	LS_allegro_get_color(WHITE);
Verd	LS_allegro_get_color(GREEN);

Tal i com es mostra a la taula anterior, les constants de color són per utilitzar-les com a argument de la funció **LS_allegro_get_color(...)** per tal d'obtenir com a valor de retorn un color del tipus ALLEGRO_COLOR que correspongui amb les característiques del color sol·licitat per l'usuari. Aquest tipus de variable serà de gran utilitat per les funcions de dibuix de figures bàsiques, com per exemple, rectangles o cercles que necessiten d'un argument d'aquest tipus de variable.

2.2 Constants de mida de lletra

La llibreria Allegro 5 ja no utilitza la font predefinida pel sistema operatiu com succeïa en versions anteriors. Com a novetat, s'obliga a que l'usuari defineixi els tipus de fonts que es volen utilitzar dins del videojoc a través de fitxers de definició de fonts True Type *.ttf*. Això comporta que, a l'inici del programa, s'hagin de carregar i interpretar aquests fitxers tot fixant la mida de la font amb la que es vol treballar.

Com ja es pot preveure, aquest tipus de pràctica resulta massa complexa d'entendre per a un alumne de primer, ja que a l'hora de carregar les fonts és necessiten punters i memòria dinàmica que, a segons quines alçades de curs, encara no s'han vist a classe de teoria.

Novament, per simplificar aquesta tasca, el mòdul LS_Allegro incorpora una interfície que permet treballar de manera fàcil amb les fonts. Concretament, el mòdul sempre intentarà carregar un fitxer anomenat *font.tft* que haurà d'estar al mateix directori que l'executable del nostre videojoc. D'aquesta manera, l'usuari final haurà de descarregar-se una font en format *.ttf*, copiar el fitxer al directori on es trobi l'executable del videojoc i reanomenar el fitxer tot fixant-li el nom *font.tft*.

Un cop fet això, només caldrà conèixer les diferents mides de fonts predefinides que existeixen dins del mòdul. Aquestes constants són les que podrem passar com a paràmetre a la funció LS_allegro_get_font(CONSTANT_DE_FONT) per tal d'obtenir, com a valor de retorn, una variable del tipus ALLEGRO_FONT * que necessitem a totes les funcions d'escriptura de text dins de l'entorn gràfic.

A continuació es deixen les constants disponibles per les diferents mides de font disponibles dins del mòdul així com les seves respectives crides a través de la funció `LS_allegro_get_font(..)`:

Mida	Crida
SMALL	<code>LS_allegro_get_font(SMALL)</code>
NORMAL	<code>LS_allegro_get_font(NORMAL)</code>
LARGE	<code>LS_allegro_get_font(LARGE)</code>
EXTRA_LARGE	<code>LS_allegro_get_font(EXTRA_LARGE)</code>

3 Funcions i procediments del mòdul `LS_allegro`

3.1 Procediments

En aquesta secció s'expliquen tots els detalls que fan referència a les rutines que no retornen cap tipus de variable dins del mòdul `LS_Allegro` (retornen `void`).

3.1.1 `void LS_allegro_exit();`

Explicació:

Aquest procediment realitza l'alliberació de recursos emprats pel mòdul `LS_Allegro` com per la llibreria `Allegro 5`. Això permet que, un cop haguem acabat la tasca que requereixi d'interfície gràfica, puguem alliberar la memòria RAM del nostre ordinador per tal de deixar espai a la resta de processos i programes que corren simultàniament a l'ordinador.

És important saber, que tota configuració prèvia d'`Allegro` o del mòdul `LS_Allegro` serà eliminada de memòria. Per tant, si es volen recuperar, caldrà tornar a definir-les abans de iniciar de nou la finestra gràfica.

Paràmetres:

Cap

3.1.2 `void LS_allegro_console_fflush();`

Explicació:

Aquest procediment neteja els continguts dels *buffers* d'entrada i de sortida del teclat. Per explicar exactament que fa aquesta funció cal entendre que el sistema operatiu llegeix el teclat com si fos una cua, és a dir, tota tecla que l'usuari premi al teclat és encuada a l'espera de ser llegida.

Aquesta cua que conté totes les tecles que s'han premut al teclat és el que anomenem *buffer* d'entrada. D'aquesta manera, quan es volen llegir dades de teclat el sistema operatiu només ha d'anar desencuant tants caràcters com se li hagin demanat. No obstant això, aquesta estratègia presenta un petit problema i és que si l'usuari prem més tecles de les que posteriorment es demanen per llegir, la resta es queden emmagatzemades al *buffer* fins que s'efectui la següent lectura.

Com ja es pot entreveure, això pot ser perjudicial perquè la següent lectura de teclat que realitzem no contindrà els caràcters que hagi teclejat l'usuari fins al moment, sinó que contindrà els caràcters que s'hagin quedat de forma residual al buffer més els caràcters que l'usuari hagi premut fins al moment de la lectura.

Per solucionar aquest detall, aquest procediment realitza un esborrat incondicional dels caràcters encuats al *buffer* d'entrada del sistema operatiu. D'aquesta manera, si realitzem una crida a aquest procediment abans de qualsevol lectura de teclat, ens assegurarem de que les dades llegides siguin, únicament, les que l'usuari ha premut en aquell moment sense la presència de caràcters residuals.

Nota: Aquest procediment presenta múltiples implementacions per a ser compatible en màquines Linux, Windows i MAC, ja que el manegament d'aquests *buffers* és diferent en cada sistema operatiu.

Paràmetres:

Cap

3.1.3 void LS_allegro_console_clear_screen();

Explicació:

Aquest procediment esborra els continguts de la pantalla de línia de comandes que acompanya sempre tot programa que utilitzi la llibreria Allegro 5. D'aquesta manera, podem mostrar nova informació a l'usuari sense que aquest es confongui amb el contingut anterior de la finestra de consola.

Nota: Aquest procediment té implementacions diferents en funció de si es compila sobre una màquina Linux, Windows o MAC ja que el tractament del netejat de consola és diferent en cada sistema operatiu.

Paràmetres:

Cap

3.1.4 void LS_allegro_clear_and_paint(ALLEGRO_COLOR color);

Explicació:

Per entendre el funcionament d'aquest procediment, s'ha d'explicar com s'ho fa Allegro 5 per mostrar els continguts a través de la finestra gràfica.

La idea és prou senzilla, ens hem d'imaginar que internament, a la RAM del nostre ordinador, existeixen dos finestres gràfiques. Una d'elles és la que veiem per pantalla i l'altre és invisible. Quan nosaltres fem crides a les funcions d'Allegro 5 que permeten dibuixar elements a la finestra gràfica, realment el que estem fent és que aquests elements es dibuixin sobre la finestra gràfica invisible. Aquesta estratègia es deu a que el temps necessari per pintar objectes en una finestra gràfica invisible és molt més baix que el temps per fer-ho en una de visible.

Finalment, per fer que els canvis apareguin per pantalla, només haurem d'intercanviar l'estat de les finestres gràfiques, fent que la visible esdevingui invisible i viceversa. Això permet que mai haguem de perdre temps pintant sobre una finestra gràfica visible, cosa que provocaria que el nostre joc presentés retards alhora de pintar les imatges per pantalla.

Un cop explicat això, podem dir que aquest procediment el que fa és intercanviar l'estat de les dues finestres gràfiques i pintar, del color que rep per paràmetre, el fons de la finestra que queda en estat invisible.

Paràmetres:

ALLEGRO_COLOR color: Color del qual volem pintar la pantalla. Ha de ser del tipus **ALLEGRO_COLOR**, que recordem que es pot generar amb la funció **ALLEGRO_COLOR** **LS_allegro_get_color(int nColor)**.

3.2 Funcions

En aquesta secció s'expliquen tots els detalls que fan referència a les rutines del mòdul **LS_Allegro** que retornen algun tipus de dada diferent a void que pot ser interessant recollir en alguns dels cassos.

3.2.1 `int LS_allegro_init(int nWidth,int nHeight, char *sWindowName);`

Explicació:

Aquesta funció s'encarrega d'inicialitzar i demanar memòria per a tots els components necessaris per crear la finestra gràfica d'Allegro 5. Els paràmetres d'aquesta funció permeten fixar la mida de la finestra que es vol crear així com també el seu títol.

Nota: Aquesta funció haurà de ser cridada abans de fer ús qualsevol altra funció d'aquest mòdul, ja que si no s'inicialitza Allegro 5, la resta de funcions no funcionaran o experimentaran un funcionament anòmal.

Paràmetres:

int nWidth: Enter que determina la mida en amplada de la finestra gràfica que es desitja crear.

int nHeight: Enter que determina la mida en alçada de la finestra gràfica que es desitja crear.

char *sWindowName: Cadena de caràcter que estableix el nom de la finestra gràfica que es desitja crear.

Retorn:

Aquesta funció retornarà un enter el valor del qual dependrà de si s'ha pogut o no inicialitzar la llibreria Allegro 5. En cas satisfactori, aquesta funció retornarà un 1 (CERT). En qualsevol altre cas, retornarà un 0 (FALS) que ens alertarà que Allegro 5 no s'ha inicialitzat i que per tant **no** es poden usar les funcions ni d'Allegro 5 ni del mòdul **LS_Allegro**.

3.2.2 int LS_allegro_key_pressed(int nKey)

Explicació:

Aquesta funció s'encarrega de retornar un enter el valor del qual dependrà de si l'usuari ha premut o no la tecla que se li passa com a paràmetre.

Nota: Cal tenir molt en compte que aquesta funció efectua una lectura destructiva quan consulta l'estat d'una tecla en concret. Això vol dir que, al cridar-la, aquesta retornarà si ha estat o no premuda però forçarà que el nou estat de la tecla, just després de la consulta, sigui **no** premuda.

Paràmetres: **int nKey:** Aquest enter simbolitza la tecla de la qual volem consultar l'estat. Els valors que pot adoptar aquest paràmetre es mostren a continuació:

Lletres:

ALLEGRO_KEY_A	ALLEGRO_KEY_K	ALLEGRO_KEY_T
ALLEGRO_KEY_B	ALLEGRO_KEY_L	ALLEGRO_KEY_U
ALLEGRO_KEY_C	ALLEGRO_KEY_M	ALLEGRO_KEY_V
ALLEGRO_KEY_D	ALLEGRO_KEY_M	ALLEGRO_KEY_W
ALLEGRO_KEY_E	ALLEGRO_KEY_N	ALLEGRO_KEY_X
ALLEGRO_KEY_F	ALLEGRO_KEY_O	ALLEGRO_KEY_Y
ALLEGRO_KEY_G	ALLEGRO_KEY_P	ALLEGRO_KEY_Z
ALLEGRO_KEY_H	ALLEGRO_KEY_Q	
ALLEGRO_KEY_I	ALLEGRO_KEY_R	
ALLEGRO_KEY_J	ALLEGRO_KEY_S	

Caràcters Especials:

ALLEGRO_KEY_F1	ALLEGRO_KEY_ENTER
ALLEGRO_KEY_PAD_ASTERISK	ALLEGRO_KEY_SEMICOLON2
ALLEGRO_KEY_F2	ALLEGRO_KEY_SEMICOLON
ALLEGRO_KEY_PAD_MINUS	ALLEGRO_KEY_COMMAND
ALLEGRO_KEY_F3	ALLEGRO_KEY_QUOTE
ALLEGRO_KEY_PAD_PLUS	ALLEGRO_KEY_LSHIFT
ALLEGRO_KEY_F4	ALLEGRO_KEY_BACKSLASH
ALLEGRO_KEY_PAD_DELETE	ALLEGRO_KEY_RSHIFT
ALLEGRO_KEY_F5	ALLEGRO_KEY_BACKSLASH2
ALLEGRO_KEY_PAD_ENTER	ALLEGRO_KEY_LCTRL
ALLEGRO_KEY_F6	ALLEGRO_KEY_COMMA
ALLEGRO_KEY_PRINTSCREEN	ALLEGRO_KEY_RCTRL
ALLEGRO_KEY_F7	ALLEGRO_KEY_FULLSTOP
ALLEGRO_KEY_PAUSE	ALLEGRO_KEY_ALT
ALLEGRO_KEY_F8	ALLEGRO_KEY_SLASH
ALLEGRO_KEY_ABNT_C1	ALLEGRO_KEY_ALTGR
ALLEGRO_KEY_F9	ALLEGRO_KEY_SPACE
ALLEGRO_KEY_YEN	ALLEGRO_KEY_LWIN
ALLEGRO_KEY_F10	ALLEGRO_KEY_INSERT
ALLEGRO_KEY_KANA	ALLEGRO_KEY_RWIN
ALLEGRO_KEY_F11	ALLEGRO_KEY_DELETE
ALLEGRO_KEY_CONVERT	ALLEGRO_KEY_MENU
ALLEGRO_KEY_F12	ALLEGRO_KEY_HOME
ALLEGRO_KEY_NOCONVERT	ALLEGRO_KEY_SCROLLLOCK
ALLEGRO_KEY_ESCAPE	ALLEGRO_KEY_END
ALLEGRO_KEY_NOCONVERT	ALLEGRO_KEY_NUMLOCK
ALLEGRO_KEY_TILDE	ALLEGRO_KEY_PGUP
ALLEGRO_KEY_AT	ALLEGRO_KEY_CAPSLOCK
ALLEGRO_KEY_MINUS	ALLEGRO_KEY_PGDN
ALLEGRO_KEY_AT	ALLEGRO_KEY_EQUALS
ALLEGRO_KEY_LEFT	ALLEGRO_KEY_CIRCUMFLEX
ALLEGRO_KEY_BACKSPACE	ALLEGRO_KEY_RIGHT
ALLEGRO_KEY_COLON2	ALLEGRO_KEY_RCTRL
ALLEGRO_KEY_TAB	ALLEGRO_KEY_UP
ALLEGRO_KEY_KANJI	ALLEGRO_KEY_OPENBRACE
ALLEGRO_KEY_DOWN	ALLEGRO_KEY_PAD_EQUALS
ALLEGRO_KEY_CLOSEBRACE	ALLEGRO_KEY_PAD_SLASH
ALLEGRO_KEY_BACKQUOTE	

Numèrics:

ALLEGRO_KEY_0	ALLEGRO_KEY_PAD_0
ALLEGRO_KEY_1	ALLEGRO_KEY_PAD_1
ALLEGRO_KEY_2	ALLEGRO_KEY_PAD_2
ALLEGRO_KEY_3	ALLEGRO_KEY_PAD_3
ALLEGRO_KEY_4	ALLEGRO_KEY_PAD_4
ALLEGRO_KEY_5	ALLEGRO_KEY_PAD_5
ALLEGRO_KEY_6	ALLEGRO_KEY_PAD_6
ALLEGRO_KEY_7	ALLEGRO_KEY_PAD_7
ALLEGRO_KEY_8	ALLEGRO_KEY_PAD_8
ALLEGRO_KEY_9	ALLEGRO_KEY_PAD_9

Retorn:

Aquesta funció retorna un enter el valor del qual dependrà de si la tecla ha estat premuda per l'usuari o no. Els únics valors que aquest enter pot prendre són 1 (CERT) [Tecla premuda] o bé, 0 (FALS) [Tecla NO premuda].

3.2.3 ALLEGRO_COLOR LS_allegro_get_color(int nColor)

Explicació:

Aquesta funció s'encarrega de retornar una variable del tipus ALLEGRO_COLOR amb la configuració adequada per representar el color que ha rebut per paràmetre. Serà una funció molt utilitzada en totes aquelles funcions d'Allegro 5 que serveixen per pintar figures senzilles a la finestra gràfica i que demanen un paràmetre del tipus ALLEGRO_COLOR per especificar el color de la figura que permeten pintar.

Paràmetres:

int nColor: Aquest enter representa el color que es vol obtenir en el format de variable ALLEGRO_COLOR. A continuació es mostra una llista dels colors disponibles:

BLACK	DARK_BLUE
DARK_GREEN	CYAN
RED	PINK 5
ORANGE	LIGHT_GRAY
GRAY	BLUE
LIGHT_GREEN	LIGHT_BLUE
GARNET	LIGHT_PINK
YELLOW	WHITE
GREEN	

Retorn:

Aquesta funció retorna una variable del tipus ALLEGRO_COLOR configurada per representar el color que se li ha especificat mitjançant el paràmetre nColor.

3.2.4 ALLEGRO_FONT* LS_allegro_get_font(int nSize)

Explicació:

Aquesta funció s'encarrega de retornar una variable del tipus ALLEGRO_FONT* amb la configuració de mida que s'ha rebut per paràmetre. Serà una funció molt utilitzada en totes aquelles crides a funcions d'Allegro 5 que serveixin per pintar text per la finestra gràfica, ja que totes demanen un paràmetre del tipus ALLEGRO_FONT* per especificar el tipus i la mida de la font que s'utilitzarà per representar el text.

Nota: La font que s'utilitzarà serà aquella que es trobi dins de la carpeta del projecte i s'anomeni font.ttf. En cas de que no hi hagi cap fitxer en el directori del projecte amb el nom font.ttf o que sigui un fitxer corrupte, el mòdul LS-Allegro no carregarà i mostrarà un missatge d'error.

Paràmetres:

int nColor : Aquest enter indica la mida de la font que es desitja obtenir com a resultat de la crida de la funció. A continuació es llisten les mides disponibles:

SMALL	NORMAL
LARGE	EXTRA_LARGE

Retorn:

Aquesta funció retorna una variable del tipus ALLEGRO_FONT* que conté la configuració adjunta per representar la font especificada al fitxer font.ttf amb la mida especificada al paràmetre nSize.

4 Funcions i procediments pròpies d'Allegro

Fins ara s'han descrit les funcions i procediments per treballar amb el mòdul `LS_allegro`. Com ja hem dit abans, aquest s'ofereix per simplificar la interacció entre els alumnes i la llibreria Allegro 5. No obstant això, hi haurà funcions natives d'Allegro que sí hauran de conèixer els alumnes, ja que són les que els permetran dibuixar i mostrar text dins de la finestra gràfica i que, per tant, necessitaran cridar a l'hora d'implementar la seva pràctica.

La idea d'aquesta secció del manual és explicar de manera detallada cadascuna d'aquestes funcions natives d'Allegro per tal que els alumnes les puguin fer servir al llarg del desenvolupament de la pràctica.

4.1 Procediments

A continuació s'explicaran els procediments que proporciona Allegro 5 per pintar figures bàsiques a la finestra gràfica.

4.1.1 `void al_draw_line(float x1, float y1, float x2, float y2, ALLEGRO_COLOR color, float thickness)`

Explicació:

Aquest procediment dibuixa una recta entre els dos punts que se li especifiquen a través dels paràmetres **x1**, **y1**, **x2** i **y2**. D'aquesta manera, si ens ho imaginem sobre un espai 2D, ens dibuixarà una línia recta entre els punts (x1,y1) i (x2,y2).

Paràmetres:

float x1: Coordenada X del punt que farà d'origen de la recta.

float y1: Coordenada Y del punt que farà d'origen de la recta.

float x2: Coordenada X del punt que farà de destí de la recta.

float y2: Coordenada Y del punt que farà de destí de la recta.

ALLEGRO_COLOR color: Estructura `ALLEGRO_COLOR` que especifica de quin color serà la recta.

Nota: Aquesta estructura la podem generar mitjançant la crida a la funció `LS_allegro_get_color(int nColor)`.

float thickness: Aquest paràmetre descriu el gruix de la línia a pintar. Per defecte, el valor que se li ha de fixar és 0.

4.1.2 void al_draw_triangle(float x1, float y1, float x2, float y2, float x3, float y3, ALLEGRO_COLOR color, float thickness)

Explicació:

Aquest procediment dibuixa el contorn d'un triangle entre els tres punts especificats, sent cada un dels punts un vertex del triangle.

Paràmetres:

float x1: Coordenada X del punt que fa de primer vertex del triangle.

float y1: Coordenada Y del punt que fa de primer vertex del triangle.

float x2: Coordenada X del punt que fa de segon vertex del triangle.

float y2: Coordenada Y del punt que fa de segon vertex del triangle.

float x3: Coordenada X del punt que fa de tercer vertex del triangle.

float y3: Coordenada Y del punt que fa de tercer vertex del triangle.

ALLEGRO_COLOR color: Estructura ALLEGRO_COLOR que especifica de quin color serà la figura.

Nota: Aquesta estructura la podem generar mitjançant la crida a la funció **LS_allegro_get_color(int nColor)**.

float thickness: Aquest paràmetre descriu l'amplada del contorn del triangle. Per defecte, el valor que se li ha de fixar és 0.

4.1.3 void al_draw_filled_triangle(float x1, float y1, float x2, float y2, float x3, float y3, ALLEGRO_COLOR color)

Explicació:

Aquest procediment dibuixa un triangle entre els tres punts especificats. A diferència del procediment anterior, en aquest cas, el triangle té l'interior pintat del color que s'especifiqui al paràmetre *color*.

Paràmetres:

float x1: Coordenada X del punt que actua de primer vertex del triangle.

float y1: Coordenada Y del punt que actua de primer vertex del triangle.

float x2: Coordenada X del punt que actua de segon vertex del triangle.

float y2: Coordenada Y del punt que actua de segon vertex del triangle.

float x3: Coordenada X del punt que actua de tercer vertex del triangle.

float y3: Coordenada Y del punt que actua de terece vertex del triangle.

ALLEGRO_COLOR color: Estructura ALLEGRO_COLOR que especifica de quin color serà la figura i el seu interior.

Nota: Aquesta estructura la podem generar mitjançant la crida a la funció **LS_allegro_get_color(int nColor)**.

4.1.4 void al_draw_rectangle(float x1, float y1, float x2, float y2, ALLEGRO_COLOR color, float thickness)

Explicació:

Aquest procediment dibuixa el contorn d'un rectangle. El rectangle es dibuixarà entre dos vèrtexs. Les primeres coordenades especifiquen la posició del vèrtex superior esquerre del rectangle, mentre que les segones especifiquen el vèrtex inferior dret.

Paràmetres:

float x1: Coordenada X del vertex superior esquerre del rectangle.

float y1: Coordenada Y del vertex superior esquerre del rectangle.

float x2: Coordenada X del vertex inferior dret del rectangle .

float y2: Coordenada Y del vertex inferior dret del rectangle.

ALLEGRO_COLOR color: Estructura ALLEGRO_COLOR que especifica de quin color serà la figura. **Nota:** Aquesta estructura la podem generar mitjançant la crida a la funció **LS_allegro_get_color(int nColor)**.

float thickness: Aquest paràmetre descriu l'amplada del contorn del rectangle. Per defecte, el valor que se li ha de fixar és 0.

4.1.5 void al_draw_filled_rectangle(float x1, float y1, float x2, float y2, ALLEGRO_COLOR color)

Explicació:

Aquest procediment dibuixa un rectangle ple de color. A diferència del procediment anterior, aquest no només dibuixa el contorn de la figura sinó que també n'omple el domini amb el color que s'especifica al paràmetre *color* del procediment.

Paràmetres:

float x1: Coordenada X del vertex superior esquerre del rectangle.

float y1: Coordenada Y del vertex superior esquerre del rectangle.

float x2: Coordenada X del vertex inferior dret del rectangle.

float y2: Coordenada Y del vertex inferior dret del rectangle.

ALLEGRO_COLOR color: Estructura ALLEGRO_COLOR que especifica de quin color serà la figura.

Nota: Aquesta estructura la podem generar mitjançant la crida a la funció **LS_allegro_get_color(int nColor)**.

4.1.6 void al_draw_circle(float cx, float cy, float r, ALLEGRO_COLOR color, float thickness)

Explicació:

Aquest procediment dibuixa el contorn d'un cercle. Per fer-ho necessita un punt (en coordenades) que especifiqui el centre del cercle i el valor del radi.

Paràmetres:

float cx: Coordenada X del punt que actua de centre del cercle.

float cy: Coordenada Y del punt que actua de centre del cercle.

float r: Distància en píxels entre qualsevol punt del contorn del cercle i el centre. No és més que el radi expressat en píxels.

ALLEGRO_COLOR color: Estructura ALLEGRO_COLOR que especifica de quin color serà el contorn de la figura.

Nota: Aquesta estructura la podem generar mitjançant la crida a la funció **LS_allegro_get_color(int nColor)**.

float thickness: Aquest paràmetre descriu l'amplada del contorn del cercle.

4.1.7 void al_draw_filled_circle(float cx, float cy, float r, ALLEGRO_COLOR color)

Explicació:

Aquest procediment dibuixa un cercle ple de color. Per fer-ho, necessita un punt (en coordenades) que determini el centre del cercle i el valor del radi.

Paràmetres:

float cx: Coordenada X del punt que actua de centre del cercle.

float cy: Coordenada Y del punt que actua de centre del cercle.

float r: Distància en píxels entre qualsevol punt del contorn del cercle i el centre. No és més que el radi expressat en píxels.

ALLEGRO_COLOR color: Estructura ALLEGRO_COLOR que especifica de quin color serà la figura.

Nota: Aquesta estructura la podem generar mitjançant la crida a la funció **LS_allegro_get_color(int nColor)**.

4.1.8 void al_draw_ellipse(float cx, float cy, float rx, float ry, ALLEGRO_COLOR color, float thickness))

Explicació:

Aquest procediment dibuixa el contorn d'una el·lipse a la finestra gràfica d'Allegro 5. Per fer-ho necessita rebre un punt (en coordenades) que determini el centre de l'el·lipse. Per altra banda i a diferència dels procediments vinculats amb els cercles, aquest procediment necessita rebre la mesura dels dos radis que definiran la forma de l'el·lipse a dibuixar.

Paràmetres:

float cx: Coordenada X del punt que actuarà de centre de l'el·lipse.

float cy: Coordenada Y del punt que actuarà de centre de l'el·lipse.

float rx: Nombre de píxels que especifiquen la longitud del radi horitzontal de l'el·lipse.

float ry: Nombre de píxels que especifiquen la longitud del radi vertical de l'el·lipse.

ALLEGRO_COLOR color: Estructura ALLEGRO_COLOR que especifica de quin color serà la figura.

Nota: Aquesta estructura la podem generar mitjançant la crida a la funció **LS_allegro_get_color(int nColor)**.

float thickness: Aquest paràmetre descriu l'amplada del contorn de l'el·lipse.

4.1.9 void al_draw_filled_ellipse(float cx, float cy, float rx, float ry, ALLEGRO_COLOR color)

Explicació:

Aquest procediment dibuixa una el·lipse plena de color. A diferència del procediment anterior, aquí el paràmetre *color* no determina de quin color serà el contorn de l'el·lipse sino que, en aquest cas, determina de quin color es pintarà tant el contorn com el domini d'aquesta.

De manera anàloga, també necessita rebre un punt (en coordenades) que determini el centre de l'el·lipse i dos radis que determinin quina forma tindrà aquesta.

Paràmetres:

float cx: Coordenada X del punt que actuarà de centre de l'el·lipse.

float cy: Coordenada Y del punt que actuarà de centre de l'el·lipse.

float rx: Nombre de píxels que especifiquen la longitud del radi horitzontal de l'el·lipse.

float ry: Nombre de píxels que especifiquen la longitud del radi vertical de l'el·lipse.

ALLEGRO_COLOR color: Estructura ALLEGRO_COLOR que especifica de quin color serà la figura.

Nota: Aquesta estructura la podem generar mitjançant la crida a la funció **LS_allegro_get_color(int nColor)**.