

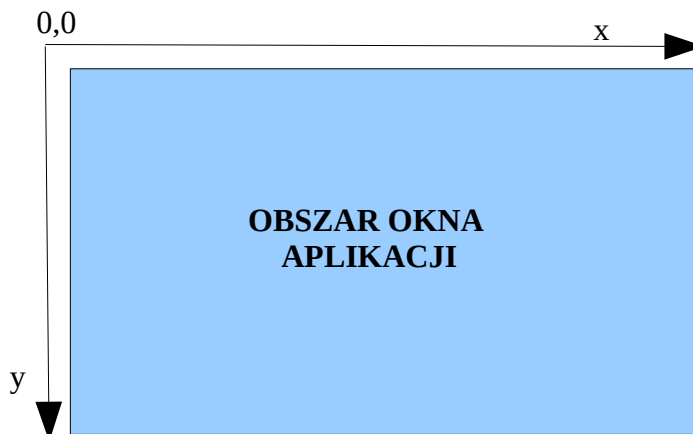
Instrukcja laboratoryjna 1	Grafika komputerowa 2D
	Temat: Wprowadzenia do biblioteki graficznej Allegro
	Przygotował: dr inż. Grzegorz Łukawski, mgr inż. Maciej Lasota, mgr inż. Tomasz Michno

1 Biblioteka Allegro

Allegro jest darmową, wieloplatformową biblioteką udostępnianą na licencji gift-ware (do wersji 5.0) i zlib (od wersji 5.0). Biblioteka najczęściej wykorzystywana jest do tworzenia animacji oraz gier 2D, ponieważ oprócz tworzenia okien i zarządzania samym obrazem, dostarcza funkcji m. in. do obsługi urządzeń peryferyjnych, plików, dźwięku, czy timerów. Na laboratoriach będziemy korzystali z biblioteki Allegro w wersji 4.

Reprezentacja ekranu

Biblioteka Allegro za ekran uznaje obszar własnego okna dla aplikacji okienkowych lub obszar całego ekranu dla aplikacji pełnoekranowych. Numeracja pikseli zaczyna się w lewym górnym rogu, wzrastając w prawo i w dół.



Z pojęciem ekranu związane jest też pojęcie tzw. ekranu wirtualnego, który jest dodatkowym obszarem znajdującym się w pamięci karty graficznej.

Ekran w bibliotece Allegro jest reprezentowany poprzez zmienną o nazwie **screen**, która jest tworzona w trakcie uruchamiania trybu graficznego. Zmienna jest typu strukturalnego **BITMAP**, umożliwia odczytanie szerokości i wysokości poprzez atrybuty `w` oraz `h`.

Najprostsza aplikacja

Program przykładowy:

```
#include <allegro.h> // plik nagłówkowy biblioteki

int main(){

    allegro_init(); // inicjalizacja biblioteki
```

```

set_gfx_mode(GFX_AUTODETECT_WINDOWED, 640, 480, 0, 0); /* ustawienie trybu
                                                                    graficznego */
install_keyboard(); // włączenie obsługi klawiatury

// TUTAJ TREŚĆ PROGRAMU (rysowanie itp.)

while( (readkey() & 0xff) != 27); // zatrzymanie programu (oczekiwanie na klawisz ESC)

allegro_exit(); // zwolnienie zasobów biblioteki
return 0;
}
END_OF_MAIN() // makro zapewniające wieloplatformowość kodu

```

Pierwszym krokiem w tworzeniu aplikacji w Allegro jest inicjalizacja biblioteki poprzez funkcję `allegro_init()`, następnie należy ustawić tryb graficzny funkcją `set_gfx_mode(int card, int w, int h, int v_w, int v_h)`, gdzie:

card – tryb pracy:

GFX_AUTODETECT – automatyczne wykrycie ustawień, pierwszeństwo ma tryb pełnego ekranu

GFX_AUTODETECT_FULLSCREEN - pełny ekran

GFX_AUTODETECT_WINDOWED - okno

GFX_SAFE – jeżeli nie można użyć podanej rozdzielczości, użycie domyślnej rozdzielczości dla danego systemu (np. dla Windows – 640x480)

GFX_TEXT – tryb tekstowy

w – szerokość

h – wysokość

v_w – szerokość obszaru wirtualnego

v_h – wysokość obszaru wirtualnego

Kolejnym krokiem jest dodanie obsługi klawiatury (`install_keyboard()`) oraz podanie właściwego kodu programu. Po narysowaniu obrazu dobrym rozwiązaniem jest zatrzymanie działania aplikacji, aż do naciśnięcia dowolnego klawisza (w przykładzie klawisz ESC, którego kod ASCII to 27). Po użyciu biblioteki należy zwolnić wszystkie wykorzystywane przez nią zasoby poprzez `allegro_exit()`. Ostatnim elementem jest dodanie wywołania makra `END_OF_MAIN()` po funkcji `main`.

Funkcje graficzne

Narysowanie punktu na ekranie jest realizowane za pomocą funkcji `putpixel(BITMAP *bmp, int x, int y, int color)`, gdzie:

bmp – bitmapa, na której ma zostać narysowany punkt, dla ekranu należy użyć predefiniowanej zmiennej `screen`, która określa okno aplikacji

x, y – współrzędne na ekranie

color – kolor punktu, w celu ustalenia jego wartości najlepiej jest użyć funkcji `makecol(int r, int g, int b)`

przykład:

narysowanie niebieskiego punktu o współrzędnych 100, 128:

```
putpixel(screen, 100, 128, makecol(0,0,255));
```

Biblioteka pozwala również na pobranie koloru piksela:

int getpixel(BITMAP *bmp, int x, int y);

Wynikiem funkcji jest wartość typu int, którą można użyć w innych funkcjach (np. putpixel()).

Pozostałe przydatne funkcje:

Funkcja	Opis
void line (BITMAP *bmp, int x1, int y1, int x2, int y2, int color)	Linia o początku w punkcie (x1, y1), końcu w (x2, y2) i kolorze color
void triangle (BITMAP *bmp, int x1, y1, x2, y2, x3, y3, int color)	Trójkąt (wypełniony) o wierzchołkach w punktach (x1, y1), (x2, y2), (x3, y3)
void rect (BITMAP *bmp, int x1, int y1, int x2, int y2, int color)	Prostokąt
void rectfill (BITMAP *bmp, int x1, int y1, int x2, int y2, int color)	Wypełniony prostokąt
void ellipse (BITMAP *bmp, int x, int y, int rx, int ry, int color)	Elipsa
void ellipsefill (BITMAP *bmp, int x, int y, int rx, int ry, int color)	Wypełniona elipsa
void clear_bitmap (BITMAP *bitmap)	Wyczyszczenie bitmapy i wypełnienie kolorem czarnym
void clear_to_color (BITMAP *bitmap, int color);	Wyczyszczenie i wypełnienie bitmapy kolorem podanym w parametrze color

Obsługa błędów

Każda z funkcji: [allegro_init\(\)](#), [set_gfx_mode\(...\)](#), [install_keyboard\(\)](#) jako wynik swojego działania zwraca wartość typu int. Wartość 0 oznacza prawidłowe wykonanie funkcji, inne wartości - błąd. Dodatkowo w zmiennej [allegro_error](#) przechowywana jest tekstowa informacja o ostatnim błędzie.

Do wyświetlania komunikatów (np. o błędzie) można wykorzystać funkcję [allegro_message](#)(const char *text_format, ...), która wyświetla okno z tekstem podanym jako parametr (w systemach, w których nie ma możliwości utworzenia okna, tekst jest wyświetlany na konsoli tekstowej). Sposób tworzenia komunikatu jest identyczny jak w funkcji printf() języka C.

przykład:

```
if(allegro_init()!=0){
    allegro_message("Wystapil blad przy inicjalizacji biblioteki.\n%s", allegro_error);
    exit(1);
}
```

Obsługa klawiatury

Biblioteka Allegro, jak można było zauważyć w przykładowym programie, posiada zaimplementowaną obsługę klawiatury. W celu jej włączenia należy na początku programu wywołać funkcję `install_keyboard()`, która podmienia standardową procedurę przerywania na własną. Z tego względu, w przypadku gdy nie korzystamy z funkcji `allegro_exit()`, należy zawsze pod koniec programu usunąć obsługę klawiatury za pomocą `remove_keyboard()`.

Wartość wciśniętego klawisza można odczytać za pomocą funkcji `readkey()`, która zwraca wynik typu `int`. Wartość ta zawiera w sobie zarówno kod ASCII znaku (w młodszych bajtach) jak również tzw. scancode (w starszych bajtach). Podczas odczytywania wartości należy odpowiednio wyzerować najstarsze bity lub przesunąć starsze bity na młodsze. Scancode jest zdefiniowaną w bibliotece wartością klawisza w postaci `KEY_NAZWA`, np. `KEY_ESC`. Wszystkie kody można znaleźć w pliku `allegro/keyboard.h`.

Przykład odczytania znaku za pomocą kodu ASCII:

```
if ((readkey() & 0xff) == 27) allegro_message("Wciśnięto klawisz ESC. \n");
```

Przykład odczytania znaku za pomocą scancode:

```
if ((readkey() >> 8) == KEY_ESC) allegro_message("Wciśnięto klawisz prawej strzałki. \n");
```

Inne przydatne funkcje:

Funkcja	Opis
<code>int keypressed()</code>	Zwraca wartość TRUE, jeśli naciśnięto klawisz
<code>int ureadkey(int *scancode)</code>	Działa analogicznie do <code>readkey()</code> , odczytuje wartość klawisza w postaci znaku UNICODE
<code>int scancode_to_ascii(int scancode)</code>	Zamienia scancode na znak ASCII

Kompilacja

Linux:

W celu wykrycia jakie flagi należy dodać podczas kompilacji, najlepiej jest użyć programu `allegro-config`.

np.

ręczna kompilacja:

```
g++ main.cpp -o main `allegro-config --cflags` `allegro-config --cppflags` `allegro-config --libs`
```

IDE Code::Blocks:

Project->Build options, zakładka Compiler settings->Other options dodać:

```
`allegro-config --cflags`
```

```
`allegro-config --cppflags`
```

zakładka *Linker settings*->*Other linker options* dodać:

```
`allegro-config --libs`
```

Windows:

Należy pobrać archiwum *Allegro.4.2.2.win32.zip* z Achillesa i rozpakować je. Po rozpakowaniu skopiować zawartość folderów *include* oraz *lib* do folderów *include* i *lib* Code::Blocks'a (standardowo: *C:\Program Files\CodeBlocks\MinGW*). Z folderu *bin* należy skopiować plik *alleg42.dll* do *C:\WINDOWS\system32*.

Następnie w celu kompilacji należy dodać w:

Project->*Build options*, zakładka *Linker settings*->*Other linker options*

plik **liballeg.a** oraz **liballeg_s.a** (w tej kolejności) znajdujący się w folderze *lib* (standardowo w: *C:\Program Files\CodeBlocks\MinGW\lib*).

Przydatne strony:

<http://alleg.sourceforge.net/>

http://wiki.allegro.cc/index.php?title=Allegro_4_Tutorial

<http://www.allegro.cc/manual/4/>

<http://www.codeblocks.org/> - IDE, z którego będziemy korzystali na laboratoriach

2 Zadania do wykonania

1. Przetestować ustawienia trybu graficznego biblioteki Allegro.
2. Napisać program, który narysuje podstawowe figury geometryczne (prostokąt, trójkąt, okrąg), zgrupowane w środku ekranu i obramowane prostokątem stworzonym z linii. Wykorzystać atrybuty *w* oraz *h* zmiennej *screen*.
3. Napisać program, który rysuje na ekranie odcinki, okręgi, trójkąty i prostokąty o wylosowanych parametrach współrzędnych w losowym kolorze.