



Disciplina: PROGRAMAÇÃO DE SOFTWARE BÁSICO

Aula 2: Bibliotecas – data, hora e gráficos

Instalação da biblioteca graphics.h no Dev C++

A biblioteca graphics.h foi adaptada para funcionar no compilador do Dev C++, que não permite chamadas ao sistema baseadas em DOS 16 bits, como o Turbo C++ pode fazer. Para importar a biblioteca para o Dev C++ Versão 5.11 devem ser seguidos os seguintes passos:

1. Baixar o driver adaptado em https://drive.google.com/file/d/14BLoBy0PA_A7_gdgO6UWiLEzrMP7Hua/view (clique em download na página que se abre).
2. Descompactar o arquivo "Graphics in Dev C++.zip".
3. Copiar os arquivos 6-ConsoleAppGraphics.template e ConsoleApp_cpp_graph.txt (figura 1) para a pasta template dentro da pasta onde o Dev C++ está instalado, em geral nos arquivos de programa do Windows (C:\Program Files (x86)\Dev-Cpp\Templates)

Nome	Data de modificaç...	Tipo	Tamanho
Test Graphics.h	13/05/2019 17:20	Pasta de arquivos	
Test Winbgim.h	13/05/2019 17:20	Pasta de arquivos	
6-ConsoleAppGraphics.template	12/04/2019 12:17	Dev-C++ Templat...	1 KB
ConsoleApp_cpp_graph.txt	12/04/2019 12:17	Documento de Te...	0 KB
graphics.h	12/04/2019 12:17	C Header File	14 KB
libbgi.a	12/04/2019 12:17	Arquivo A	55 KB
winbgim.h	12/04/2019 12:17	C Header File	14 KB

Fig. 1

4. Copiar os arquivos `graphics.h` e `winbgim.h` para a pasta `include` do compilador que será usado (32 bits) em geral nos arquivos de programa do Windows (`C:\Program Files (x86)\Dev-Cpp\MinGW64\x86_64-w64-mingw32\include`).
5. Copiar o arquivo `libbgi.a` para a pasta `lib` e para a pasta `lib32` do compilador que será usado (32 bits) em geral nos arquivos de programa do Windows (`C:\Program Files (x86)\Dev-Cpp\MinGW64\x86_64-w64-mingw32\lib`) e (`C:\Program Files (x86)\Dev-Cpp\MinGW64\x86_64-w64-mingw32\lib32`).
6. Criar um projeto no Dev C++ com as opções `Console Graphics` e `C++ Project`, como mostrado na figura 2 (apesar de ser uma biblioteca de C, a opção `C project` não funciona).

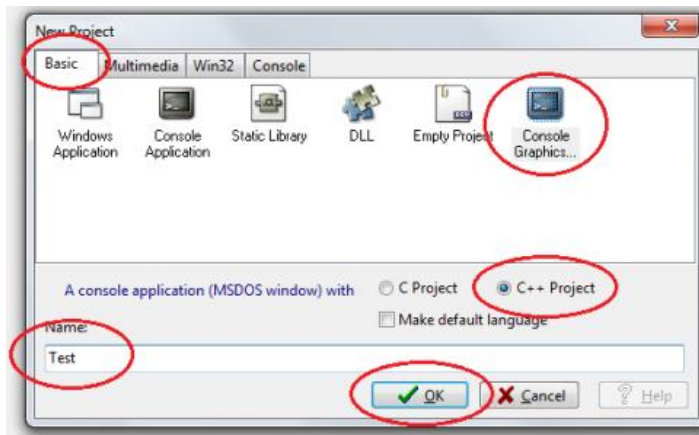


Fig. 2

7. Escolher o compilador de 32 bits, como mostrado na figura 3 (TDM-GCC 4.9.2 32-bit Release).

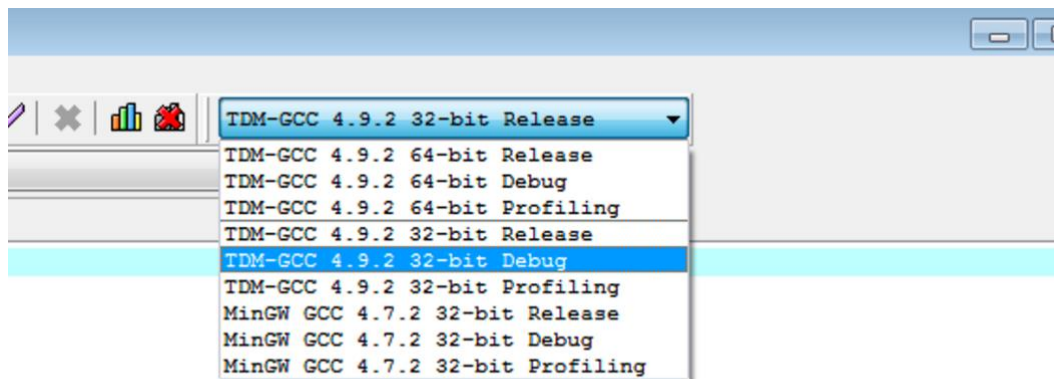


Fig. 3

Agora no arquivo main.cpp do projeto é possível escrever programas que acessem esta biblioteca graphics.h. O exemplo abaixo realiza elementos gráficos na tela.

```
#include<graphics.h>

int main (){
    initwindow (800, 800);

    int left=100,top=100,right=200,bottom=200,x= 300,y=150,radius=50;

    rectangle(left, top, right, bottom);
    circle(x, y, radius);
    bar(left + 300, top, right + 300, bottom);
    line(left - 10, top + 150, left + 410, top + 150);
    ellipse(x, y + 200, 0, 360, 100, 50);
    outtextxy(left + 100, top + 325, "Meu programa grafico");

    getch();
}
```



Atividade Verificadora de Aprendizagem:

Crie um projeto no Dev C++ para usar as funções da biblioteca graphics.h. Execute o programa abaixo e verifique o que ele faz.

```
#include<graphics.h>

int main (){
    initwindow (400, 400);
    char a[5];
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 3);
    setcolor(RED);

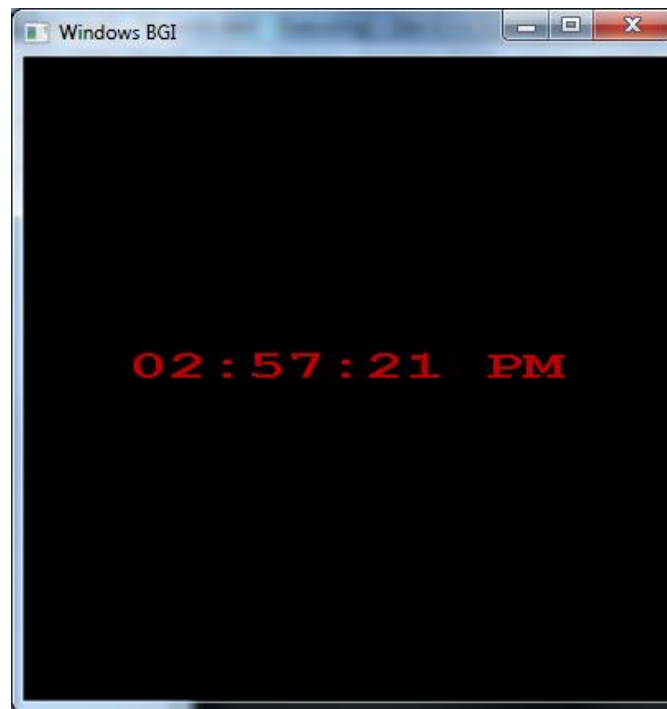
    for (int i = 30; i >=0; i--)
    {
        sprintf(a, "%d", i);
        outtextxy(getmaxx()/2, getmaxy()/2, a);
        delay(1000);

        if (i == 0)
            break;
        cleardevice();
    }

    getch();
}
```

Com base nas bibliotecas de gráfico e tempo estudadas, altere o programa para que apresente com letras gráficas, a hora, minuto e segundo atuais no formato HH:MM:SS PM, durante 1 minuto.

A figura abaixo mostra a imagem da execução do programa.





Gabarito

O programa fornecido escreve na tela um número que decremente de 30 a 0. Para alterar como solicitado é necessário incluir a biblioteca time.h. Dentro de um for que itera 60 vezes, coloca-se a captura da hora local e faz-se a impressão, como descrito abaixo:

```
#include <time.h>
#include<graphics.h>

#define LEN 150

int main (){
    char buf[LEN];
    time_t curtime;
    struct tm *loc_time;

    initwindow (400, 400);
    settxtjustify(CENTER_TEXT, CENTER_TEXT);
    settxtstyle(DEFAULT_FONT, HORIZ_DIR, 3);
    setcolor(RED);

    for (int i = 60; i >=0; i--)
    {

        //Obtem hora corrente do sistema
        curtime = time (NULL);

        // Converte para o tempo local
        loc_time = localtime (&curtime);

        strftime (buf, LEN, "%I:%M:%S %p", loc_time);
        outtextxy(getmaxx()/2, getmaxy()/2, buf);
        delay(1000);

        if (i == 0)
            break;
        cleardevice();
    }

    getch();
}
```