Manual Técnico para Aplicación de Dibujo en Consola

Descripción General

Esta aplicación en C++ permite dibujar diversas formas geométricas en la consola de Windows. Los usuarios pueden seleccionar opciones de un menú para dibujar triángulos, cuadrados, rectángulos, círculos, líneas, rombos y hexágonos. Además, se pueden cambiar caracteres y colores de dibujo, limpiar la pantalla, guardar la pantalla actual en un archivo y abrir archivos de texto para mostrarlos en la consola.

Índice

Configuración Inicial

Incluye las bibliotecas necesarias y define las constantes y variables globales:

```
#include <iostream>
#include <fstream>
#include <cmath>
#include <Windows.h>
#include <conio.h>
#include <shlobj.h>

using namespace std;

const int SCREEN_WIDTH = 150;
const int SCREEN_HEIGHT = 50;
const int MENU_WIDTH = 50;

HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
```

Funciones Auxiliares

gotoxy

Mueve el cursor a una posición específica en la consola.

```
void gotoxy(int x, int y) {
    COORD cursor;
    cursor.X = x;
```

```
    cursor.Y = y;
    SetConsoleCursorPosition(hConsole, cursor);
}
```

setConsoleSize

Ajusta el tamaño de la consola.

```
void setConsoleSize(int ancho, int alto) {
    COORD bufferSize;
    bufferSize.X = ancho;
    bufferSize.Y = alto;
    SetConsoleScreenBufferSize(hConsole, bufferSize);

    SMALL_RECT windowSize = {0, 0, ancho - 1, alto - 1};
    SetConsoleWindowInfo(hConsole, TRUE, &windowSize);
}
```

maximizarConsola

Maximiza la ventana de la consola.

```
void maximizarConsola() {
    HWND hwnd = GetConsoleWindow();
    ShowWindow(hwnd, SW_SHOWMAXIMIZED);
}
```

clearLine

Limpia una línea específica en la consola.

```
void clearLine(int y) {
    gotoxy(0, y);
    cout << string(SCREEN_WIDTH, ' ');
    gotoxy(0, y);
}
```

Menu

Dibuja el menú de opciones en la consola.

```
void Menu() {
    int menuX = SCREEN_WIDTH + 1;
    gotoxy(menuX, 0);
    cout << "Menu:";
    gotoxy(menuX, 1);
    cout << "PRESIONE LA TECLA SEGUN LA OPCION QUE DESEA USAR";
    gotoxy(menuX, 3);
    cout << "F1. TRIANGULO";
    gotoxy(menuX, 5);
    cout << "F2. CUADRADO";
    gotoxy(menuX, 7);
    cout << "F3. RECTANGULO";
    gotoxy(menuX, 9);
    cout << "F4. CIRCULO";
    gotoxy(menuX, 11);
    cout << "F5. LINEA";
    gotoxy(menuX, 13);
    cout << "F6. ROMBO";
    gotoxy(menuX, 15);
    cout << "F7. HEXAGONO";
    gotoxy(menuX, 17);
    cout << "F8. NUEVO CARACTER";
    gotoxy(menuX, 19);
    cout << "F9. LIMPIAR PANTALLA";
    gotoxy(menuX, 21);
    cout << "F10. COLOR DE CARACTER";
    gotoxy(menuX, 23);
    cout << "F12. GRABAR PANTALLA";
    gotoxy(menuX, 25);
    cout << "CTRL+A. ABRIR ARCHIVO";
    gotoxy(menuX, 27);
    cout << "ESC. Salir";
    gotoxy(menuX, 30);
    cout << "LEYENDA DE COLORES";
    gotoxy(menuX, 32);
    cout << "1:AZUL        6:AMARILLO";
    gotoxy(menuX, 34);
    cout << "2:VERDE       7:BLANCO";
    gotoxy(menuX, 36);
    cout << "3:AGUAMARINA  8:GRIS";
    gotoxy(menuX, 38);
    cout << "4:ROJO        9:AZUL CLARO";
```

```
    gotoxy(menuX, 40);
    cout << "5:PURPURA      0: NEGRO";
}
```

dibujarLineaDelimitacion

Dibuja una línea de delimitación en la consola.

```
void dibujarLineaDelimitacion() {
    for (int i = 0; i < SCREEN_WIDTH; i++) {
        gotoxy(i, SCREEN_HEIGHT - 3);
        cout << "-";
    }
}
```

preguntarRelleno

Pregunta si la figura debe ser dibujada con relleno.

```
bool preguntarRelleno() {
    char opcion;
    int inputY = SCREEN_HEIGHT - 1;
    gotoxy(0, inputY);
    clearLine(inputY);
    cout << "Desea dibujar con relleno? (s/n): ";
    cin >> opcion;
    clearLine(inputY);
    return (opcion == 's' || opcion == 'S');
}
```

cambiarColorTexto

Cambia el color del texto en la consola.

```
void cambiarColorTexto(WORD color) {
    SetConsoleTextAttribute(hConsole, color);
}
```

mostrarMensaje

Muestra un mensaje en una línea específica.

```
void mostrarMensaje(const string& mensaje, int y) {
    clearLine(y);
    gotoxy(0, y);
    cout << mensaje;
}
```

preguntarOrientacion

Pregunta la orientación de la figura.

```
char preguntarOrientacion(const string& opciones) {
    char opcion;
    int inputY = SCREEN_HEIGHT - 1;
    gotoxy(0, inputY);
    clearLine(inputY);
    cout << "Elija la orientacion (" << opciones << "): ";
    cin >> opcion;
    clearLine(inputY);
    return opcion;
}
```

cambiarCaracter

Cambia el carácter de dibujo.

```
void cambiarCaracter(char &cursorc) {
    int inputY = SCREEN_HEIGHT - 2;

    gotoxy(0, inputY);
    clearLine(inputY);
    cout << "Ingrese el nuevo caracter para dibujar figuras: ";
    cin >> cursorc;
    clearLine(inputY);
}
```

nuevoCaracter

Pide un nuevo carácter para mostrar.

```
void nuevoCaracter(int x, int y, char &cursorc) {
    int inputY = SCREEN_HEIGHT - 2;

    gotoxy(0, inputY);
    clearLine(inputY);
    cout << "Ingrese el caracter a mostrar: ";
    cin >> cursorc;
    clearLine(inputY);
}
```

limpiarPantalla

Limpia la pantalla de la consola.

```
void limpiarPantalla() {
    system("cls");
    SetConsoleTextAttribute(hConsole, 0xF0);
    Menu();
    dibujarLineaDelimitacion();
}
```

getCharAtPosition

Obtiene el carácter en una posición específica de la consola.

```
char getCharAtPosition(int x, int y) {
    CHAR_INFO charInfo;
    COORD bufferSize = {1, 1};
    COORD bufferCoord = {0, 0};
    SMALL_RECT readRegion = {x, y, x, y};
    HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);

    if (!ReadConsoleOutput(hConsole, &charInfo, bufferSize, bufferCoord, &readRegion))
        return ' ';
    }
```

```
    return charInfo.Char.AsciiChar;
}
```

configurarConsola

Configura la consola para usar Unicode y establece la fuente.

```
void configurarConsola() {
    SetConsoleOutputCP(65001);
    SetConsoleCP(65001);

    CONSOLE_FONT_INFOEX cfi;
    cfi.cbSize = sizeof(cfi);
    cfi.nFont = 0;
    cfi.dwFontSize.X = 0;
    cfi.dwFontSize.Y = 16;
    cfi.FontFamily = FF_DONTCARE;
    cfi.FontWeight = FW_NORMAL;


 wcscpy_s(cfi.FaceName, L"Consolas");
    SetCurrentConsoleFontEx(GetStdHandle(STD_OUTPUT_HANDLE), FALSE, &cfi);
}
```

guardarPantalla

Guarda la pantalla actual en un archivo de texto.

```
void guardarPantalla() {
    CHAR_INFO buffer[SCREEN_WIDTH * SCREEN_HEIGHT];
    COORD bufferSize = {SCREEN_WIDTH, SCREEN_HEIGHT};
    COORD bufferCoord = {0, 0};
    SMALL_RECT readRegion = {0, 0, SCREEN_WIDTH - 1, SCREEN_HEIGHT - 1};

    if (!ReadConsoleOutput(hConsole, buffer, bufferSize, bufferCoord, &readRegion)) {
        cout << "Error al leer la salida de la consola." << endl;
        return;
    }

    // Obtener la ruta del escritorio
    char desktopPath[MAX_PATH];
```

```
    if (FAILED(SHGetFolderPathA(NULL, CSIDL_DESKTOP, NULL, 0, desktopPath))) {
        cout << "Error al obtener la ruta del escritorio." << endl;
        return;
    }
    string filePath = string(desktopPath) + "\\pantalla.txt";

    ofstream outFile(filePath.c_str());
    if (!outFile) {
        cout << "Error al abrir el archivo para escribir." << endl;
        return;
    }

    for (int y = 0; y < SCREEN_HEIGHT; ++y) {
        for (int x = 0; x < SCREEN_WIDTH; ++x) {
            outFile << buffer[x + y * SCREEN_WIDTH].Char.AsciiChar;
        }
        outFile << endl;
    }

    outFile.close();
    cout << "Pantalla guardada en " << filePath << endl;
}
```

abrirArchivo

Abre un archivo de texto y muestra su contenido en la consola.

```
void abrirArchivo() {
    string filePath;
    int inputY = SCREEN_HEIGHT - 2;

    mostrarMensaje("Ingrese la ruta del archivo a abrir: ", inputY);
    cin >> filePath;
    clearLine(inputY);

    ifstream inFile(filePath.c_str());
    if (!inFile) {
        mostrarMensaje("Error al abrir el archivo.", inputY);
        return;
    }

    string line;
    int y = 0;
```

```
    while (getline(inFile, line) && y < SCREEN_HEIGHT - 3) {
        gotoxy(0, y);
        cout << line.substr(0, SCREEN_WIDTH);
        y++;
    }

    inFile.close();
    mostrarMensaje("Archivo cargado correctamente.", inputY);
}
```

Funciones de Dibujo

dibujarTriangulo

Dibuja un triángulo con varias opciones de orientación y relleno.

```
void dibujarTriangulo(int &startX, int &startY, char cursorc, bool relleno) {
    // Código para dibujar un triángulo
}
```

dibujarCuadrado

Dibuja un cuadrado con varias opciones de orientación y relleno.

```
void dibujarCuadrado(int &startX, int &startY, char cursorc, bool relleno) {
    // Código para dibujar un cuadrado
}
```

dibujarRectangulo

Dibuja un rectángulo con varias opciones de orientación y relleno.

```
void dibujarRectangulo(int &startX, int &startY, char cursorc, bool relleno) {
    // Código para dibujar un rectángulo
}
```

dibujarCirculo

Dibuja un círculo con opción de relleno.

```
void dibujarCirculo(int &startX, int &startY, char cursorc, bool relleno) {
    // Código para dibujar un círculo
}
```

dibujarLinea

Dibuja una línea en varias direcciones.

```
void dibujarLinea(int &startX, int &startY, char cursorc) {
    // Código para dibujar una línea
}
```

dibujarRombo

Dibuja un rombo con varias opciones de orientación y relleno.

```
void dibujarRombo(int &startX, int &startY, char cursorc, bool relleno) {
    // Código para dibujar un rombo
}
```

dibujarHexagono

Dibuja un hexágono con varias opciones de orientación y relleno.

```
void dibujarHexagono(int &startX, int &startY, char cursorc, bool relleno) {
    // Código para dibujar un hexágono
}
```

Uso del Programa

1. **Iniciar la aplicación**:
    - Compila y ejecuta el código en un entorno compatible con Windows.

2. **Navegar por el menú**:

    - Usa las teclas de función (`F1` a `F12`) y combinaciones de teclas (`CTRL+A` , `ESC`) para seleccionar opciones en el menú.

3. **Dibujar formas**:

    - Selecciona una forma desde el menú y sigue las instrucciones en la consola para definir el tamaño, orientación y si debe estar rellena.

4. **Cambiar carácter y color**:

    - Usa `F8` para cambiar el carácter de dibujo y `F10` para cambiar el color del texto.

5. **Guardar y abrir archivos**:

    - Usa `F12` para guardar la pantalla actual en un archivo de texto y `CTRL+A` para abrir y mostrar el contenido de un archivo de texto en la consola.

6. **Limpiar pantalla y salir**:

    - Usa `F9` para limpiar la pantalla y `ESC` para salir del programa.