

# Paint Program

with C++ and WinAPI

Name : 김나영

Tel : 010-3902-6074

e-mail : nayeong0121@gmail.com

# 목차

1. 개요

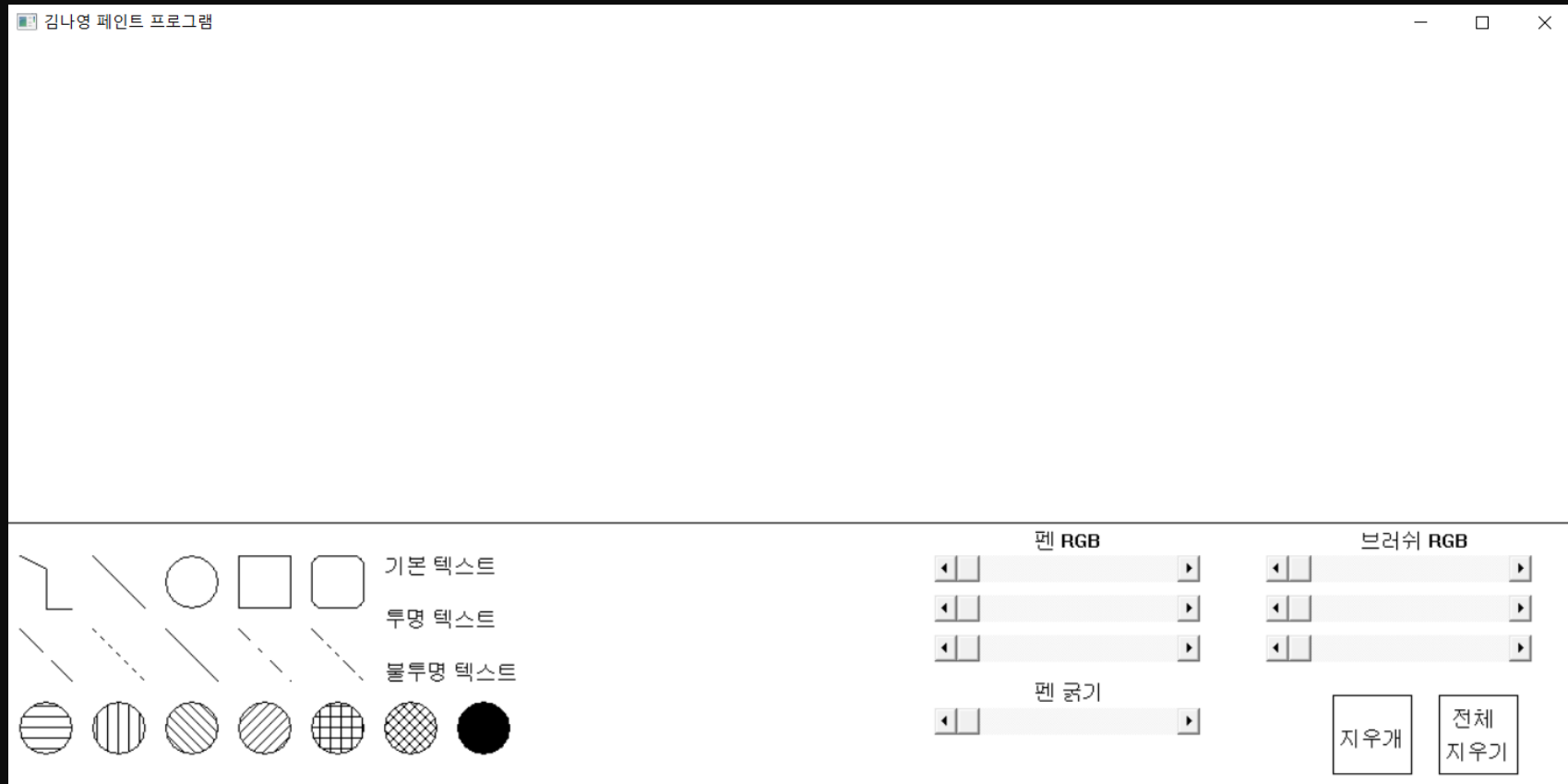
2. 화면

3. 구현 코드

# 개요

- 언어 : C++
- API : WinAPI
- 프로그램 영상 링크
- 깃 허브 링크
- 개발 기간 : 2024.08.22 ~ 2024.09.12
- C++과 WinAPI를 사용하여 개발한 페인트 프로그램입니다.

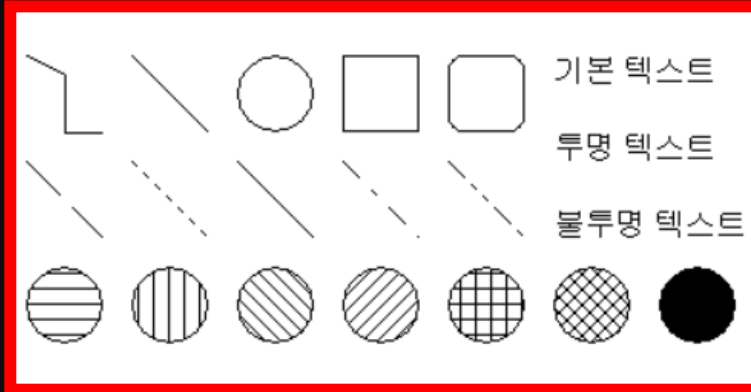
# 화면



초기 실행 화면

# 화면

1



2



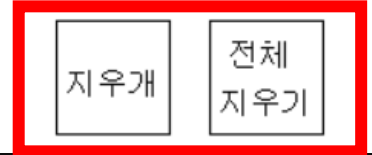
3



4



5



1. 선, 도형, 텍스트 중 어떤 것을 그릴 것인지 선택하고  
선 종류, 브러쉬 종류, 텍스트 종류를 선택하는 버튼입니다.
2. 1에서 선택한 것을 보여주는 화면입니다.
3. 펜의 색상 및 굵기를 조절하는 스크롤 바입니다.
4. 브러쉬의 색상을 조절하는 스크롤 바입니다.
5. 지우개를 선택하는 버튼입니다.

# 화면



원하는 선, 도형, 텍스트로 캔버스를 채운 화면

# 구현 코드

```
case WM_LBUTTONDOWN:
    SetRect(&rt, 0, g_nHeight - 200, g_nWidth, g_nHeight);
    InvalidateRect(hWnd, &rt, FALSE);
    x1 = x2 = LOWORD(lParam);
    y1 = y2 = HIWORD(lParam);

    SetCursor(LoadCursor(NULL, IDC_CROSS));
    ///// 커서가 영역밖을 벗어나도 지속적 마우스 메시지를 받을 수 있다
    SetCapture(hWnd);

    //클릭할 때마다 문자열이 초기화됨
    strLen = 0;
    str[0] = '\0';

    bDrawing = TRUE;
    break;
```

왼쪽 마우스를 눌렀을 때 실행되는 코드

# 구현 코드

```
case WM_MOUSEMOVE:
    if (bDrawing)
    {
        hdc = GetDC(hWnd);
        int cursorX = (int)(short)LOWORD(lParam);
        int cursorY = (int)(short)HIWORD(lParam);

        if (eraserStyle == 1) {
            RestrictPaint(&x1, &x2, 0, g_nWidth, 0, g_nWidth - 1);
            RestrictPaint(&y1, &y2, 0, g_nHeight - 200, 0, g_nHeight - 201);

            x2 = RestrictCursor(cursorX, 0, g_nWidth, 0, g_nWidth - 1);
            y2 = RestrictCursor(cursorY, 0, g_nHeight - 200, 0, g_nHeight - 201);

            __line(hdc, x1, y1, x2, y2, penWidth, penStyle, RGB(255,255,255));

            x1 = LOWORD(lParam);
            y1 = HIWORD(lParam);
            figureStyle = -1;
        }
        if (figureStyle == 0) {
            RestrictPaint(&x1, &x2, 0, g_nWidth, 0, g_nWidth-1);
            RestrictPaint(&y1, &y2, 0, g_nHeight - 200, 0, g_nHeight - 201);

            x2 = RestrictCursor(cursorX, 0, g_nWidth, 0, g_nWidth - 1);
            y2 = RestrictCursor(cursorY, 0, g_nHeight - 200, 0, g_nHeight - 201);
            __line(hdc, x1, y1, x2, y2, penWidth, penStyle, RGB(redPen, greenPen, bluePen));

            x1 = LOWORD(lParam);
            y1 = HIWORD(lParam);
        }
    }
    else {
        SetCursor(LoadCursor(NULL, IDC_CROSS));
        DrawFigure(hWnd, x1, y1, x2, y2, FALSE);
        x2 = RestrictCursor(cursorX, 0, g_nWidth, 1, g_nWidth - 1);
        y2 = RestrictCursor(cursorY, 0, g_nHeight - 200, 1, g_nHeight - 201);
        DrawFigure(hWnd, x1, y1, x2, y2, FALSE);
    }
}
```

마우스가 움직일 때 실행되는 코드



# 구현 코드

```
case WM_LBUTTONDOWN:  
    if (bDrawing)  
    {  
        DrawFigure(hWnd, x1, y1, x2, y2, FALSE);  
        //// 마우스 캡처를 푼다  
        ReleaseCapture();  
        bDrawing = FALSE;  
  
        SetRect(&rt, 0, g_nHeight - 200, g_nWidth, g_nHeight-180);  
        InvalidateRect(hWnd, &rt, TRUE);  
  
        SetRect(&rt, 400, g_nHeight - 200, 800, g_nHeight - 10);  
        InvalidateRect(hWnd, &rt, TRUE);  
  
        UpdateWindow(hWnd);  
        DrawFigure(hWnd, x1, y1, x2, y2, TRUE);  
    }  
    break;
```

마우스를 뺏을 때 실행되는 코드

# 구현 코드

```
void __Line(HDC hdc, int x1, int y1, int x2, int y2, int thickness, int penStyle, COLORREF crPen) {  
    HPEN oldPen, newPen;  
    newPen = CreatePen(penStyle, thickness, crPen);  
    oldPen = (HPEN)SelectObject(hdc, newPen);  
    MoveToEx(hdc, x1, y1, NULL);  
    LineTo(hdc, x2, y2);  
    SelectObject(hdc, oldPen);  
    DeleteObject(newPen);  
}
```

선을 그릴 때 실행되는 함수

# 구현 코드

```
void __Rectangle(HDC hdc, int x1, int y1, int x2, int y2, int thickness,
    int penStyle, int brushStyle, COLORREF crPen, COLORREF crBrush) {
    HPEN oldPen, newPen;
    HBRUSH oldBrush, newBrush;

    newPen = CreatePen(penStyle, thickness, crPen);
    oldPen = (HPEN)SelectObject(hdc, newPen);

    if (brushStyle == HS_FILL && defaultRect != 0)
        newBrush = CreateSolidBrush(crBrush);
    else if (defaultRect == 0)
        newBrush = (HBRUSH)GetStockObject(NULL_BRUSH);
    else
        newBrush = CreateHatchBrush(brushStyle, crBrush);
    oldBrush = (HBRUSH)SelectObject(hdc, newBrush);

    Rectangle(hdc, x1, y1, x2, y2);

    SelectObject(hdc, oldPen);
    DeleteObject(newPen);
    SelectObject(hdc, oldBrush);
    DeleteObject(newBrush);
}
```

직사각형을 그릴 때  
실행되는 함수

# 구현 코드

```
void __Text(HDC hdc, int x, int y, int textStyle, COLORREF crText, COLORREF crBK, const char* str, int stringLen) {
    switch (textStyle) {
        case TRANSPARENT:
            SetBkMode(hdc, TRANSPARENT);
            break;
        case OPAQUE:
            SetBkMode(hdc, OPAQUE);
            SetBkColor(hdc, crBK);
            break;
        case DEFAULT_TEXT:
            SetBkMode(hdc, OPAQUE);
            SetBkColor(hdc, RGB(255, 255, 255));
            break;
    }
    SetTextColor(hdc, crText);
    TextOut(hdc, x, y, str, stringLen);
}
```

글자를 작성할 때 실행되는 함수

**감사합니다.**