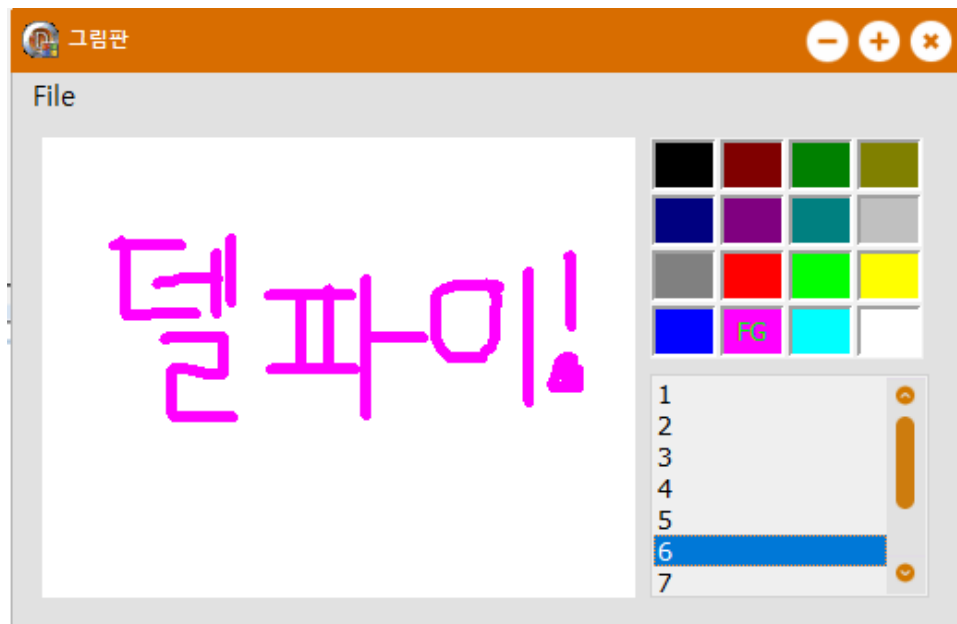


## 그림판 만들기

펜의 색상과 굵기를 선택하여 글씨와 도형들을 그리는 '그림판' 이다.

[완성된 화면]



## 실습목적

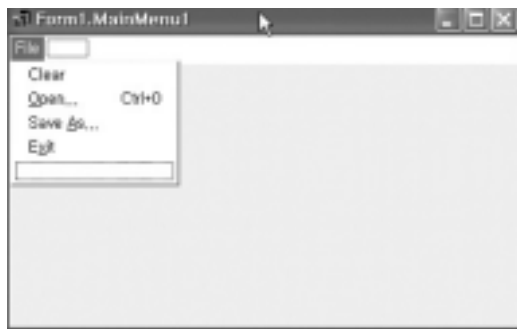
Canvas 개체의 브러쉬, 펜, 폰트등을 이용하여 도형이나 선을 그리거나 텍스트를 추가하며 그래픽 이미지를 표현할 수 있고 해당 이미지를 저장하고 오픈 할 수 있다.

## 사용 컴포넌트

- TMainMenu
- TPopupMenu
- TImage
- TColorGrid
- TListBox
- TOpenDialog
- TSaveDialog

### [따라하기]

1. File < New < VCL Forms Application - Delphi을 이용하여 어플리케이션을 시작한다.
2. File < Save All 혹은 단축키 Shift + Ctrl + S을 선택하여 UImageEditor.pas로, 프로젝트의 이름은 PImageEditor.dpr로 프로젝트를 저장한다.
3. 생성한 폼의 Caption 속성값을 '그림판' 으로 . 변경한다.
4. Project > Options > Apperance 에서 Custom Styles 항목을 다 선택한 후 Default Style 을 원하는 스타일로 지정한다.
5. TMainMenu 컴포넌트를 폼 위에 올린 후,다음과 같이 메뉴 항목을 추가한다.




6. TPopupMenu 컴포넌트를 폼위에 올린 후 위의 메인 메뉴 항목과 동일 하게 추가한다.
7. Form의 PopupMenu 속성에 PopupMenu1을 연결한다.
8. TImage 컴포넌트를 폼 위에 올려놓습니다. 그리고 크기를 알맞게 늘려줍니다.



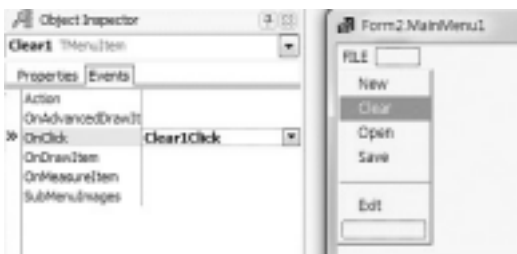
9. 원하는 색을 선택하여 그림을 그릴 수 있는 TColorGrid 컴포넌트를 선택하여 폼 위에 올려 놓습니다.

10. 팔레트에서 TListBox를 선택하여 폼 위에 올려놓습니다.

11. ListBox1의 오브젝트 인스펙터 화면에서 'Items'의  버튼을 클릭하면 아래와 같은 리스트 에디터화면을 볼 수 있습니다. 펜의 굵기를 입력하기 위해 빈 공간에 1부터 9까지 입력하고, OK 버튼을 누릅니다.



11. 이제 Menu Design에서 Clear를 클릭했을 때의 이벤트를 설정해 줍니다.

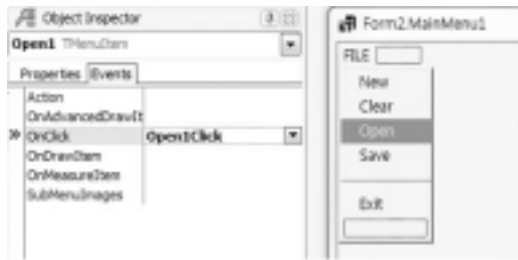


```
procedure TForm1.Clear1Click(Sender: TObject);
begin
  Image1.Picture.Bitmap.Assign(Nil); // 이미지 초기화
  FormCreate(Self);                 // 초기처리를 호출합니다
end;
```

12. 동일한 방법으로 Open을 선택하면, 기존의 \*.bmp 형식의 파일을 불러올 수 있도록

## Open

의 OnClick 이벤트 핸들러를 코딩 합니다.



```
procedure TForm1.Open1Click(Sender: TObject);
var
  OD:TOpenDialog; // 변수 선언
begin
  OD := TOpenDialog.Create(Self);
  OD.Filter := 'Bitmap Files(*.bmp)|*.bmp';
  if OD.Execute then
    Image1.Picture.Bitmap.LoadFromFile(OD.FileName);
  OD.Free; // 생성된 다이아로그를 해제합니다.
end;
```

13. 작업했던 파일을 \*.bmp 형태로 저장하기 위해 Save에 대한 코드를 구현한다.

```
procedure TForm1.Save1Click(Sender: TObject);
var
  SD:TSaveDialog;
begin
  SD := TSaveDialog.Create(Self);
  SD.Filter := 'Bitmap Files(*.bmp)|*.bmp';
  if SD.Execute then
    Image1.Picture.Bitmap.SaveToFile(SD.FileName);
  SD.Free;
end;
```

14. Exit에 대해 OnClick 이벤트 핸들러를 구현한다.

```
procedure TForm1.Exit1Click(Sender: TObject);
begin
  Close;
```

```
end;
```

15. 폼에는 **OnCreate** 이벤트 핸들러를 작성한다.

폼이 생성되었을 때, 그림을 그릴 수 있는 화면의 크기와 **ListBox**에 대한 기본값을 설정한다.

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
  Image1.Picture.Bitmap.Width := Image1.Width;  
  Image1.Picture.Bitmap.Height := Image1.Height;  
  ListBox1.ItemIndex := 0;  
end;
```

16. **Image1**에 대해서는 **OnMouseDown** 이벤트를 설정한다.

```

procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton; Shift: TShiftState;
X, Y: Integer);
Begin
// 마우스의 움직임에 따라 캔버스에 그림이 그려집니다.
Image1.Picture.Bitmap.Canvas.MoveTo(X, Y);
end;

```

17. 마우스의 선택에 따라 값이 변할 수 있도록 Image1의 OnMouseMove 이벤트 핸들러를 구현 한다.

```

procedure TForm1.Image1MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin
if Shift = [sLeft] then // 마우스 왼쪽 버튼을 이용하여
begin
// 펜의 굵기 지정
Image1.Picture.Bitmap.Canvas.Pen.Width:= StrToInt(ListBox1.Items [ListBox1.ItemIndex]);
//펜의 색상 지정
Image1.Picture.Bitmap.Canvas.Pen.Color := ColorGrid1.ForegroundColor;
Image1.Picture.Bitmap.Canvas.LineTo(X, Y);
end ;

```

완성된 그림판의 모습입니다. ColorGrid를 이용하여 원하는 색을 선택할 수 있고, ListBox 을 이용하여 굵기를 조정할 수 있습니다.

**TCanvas 클래스가 무엇인지 헬프에서 살펴보고 아래 기능들을 검색해보고 적어보십시오 ~**

- **LineTo(x,y) :**
- **MoveTo(X, Y) :**
- **FillRect**
- **TextRec**
- **TextOut(x,y)**
- **Rectangle**
- **Pie**

- Arc

교육이 끝난 후 화면에 UI 컴포넌트를 다른 컴포넌트로 대체 할 수 있는지 생각해 본다 !