MajorなMeasure

囚人

アジェンダ

- 72や96って？
- Win32(GDI)の測定
- Windowsフォーム(GDI+)の測定
- WPFの測定

72、96

何か知ってますよね

96

- Windows の一般的な DPI
- DPI･･･1インチ辺りのドット数
- でも、今私が使っているノートPCのモニタは12.1インチで1024x768
- (1024^2+768^2)/12.1≒105
- 要するに、Windowsは96ドットで1インチだと思っているが、実際は105ドットで1インチ
- 「ペイント」をチェックしてみよう

72

- フォントの１ポイントは1/72インチ

定義

- 論理DPI…画面設定のDPI
- 真DPI…LCDの推奨総ピクセル数とLCDの大きさより求めたもの
- ペイントDPI…MSペイントで表示されるDPI

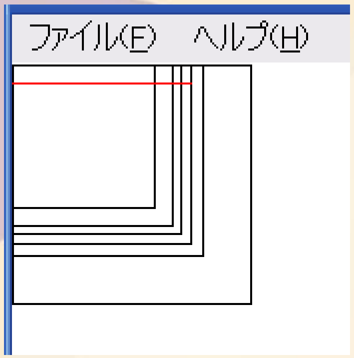
マッピングモードの違いによる GDI の描画

- GDIで正方形を描画
- 内側から72px,81px,85px,90px,96px,120pxの正方形
- 1インチの直線も描画

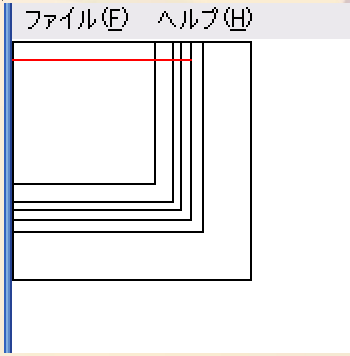
WM_PAINTの処理

```
SetMapMode(hdc, MM_TEXT);
SelectObject(hdc, GetStockObject(NULL_BRUSH));

MoveToEx(hdc, 0, 1, NULL);
Rectangle(hdc, 0, 0, 72, 72);

MoveToEx(hdc, 0, 5, NULL);
Rectangle(hdc, 0, 0, 81, 81);

MoveToEx(hdc, 0, 10, NULL);
Rectangle(hdc, 0, 0, 85, 85);

MoveToEx(hdc, 0, 15, NULL);
Rectangle(hdc, 0, 0, 90, 90);

MoveToEx(hdc, 0, 20, NULL);
Rectangle(hdc, 0, 0, 96, 96);

MoveToEx(hdc, 0, 25, NULL);
Rectangle(hdc, 0, 0, 120, 120);

SelectObject(hdc, CreatePen(PS_SOLID, 1, RGB(255, 0, 0)));
SetMapMode(hdc, MM_LOENGLISH);
MoveToEx(hdc, 0, -10, NULL);
LineTo(hdc, 100, -10);
```

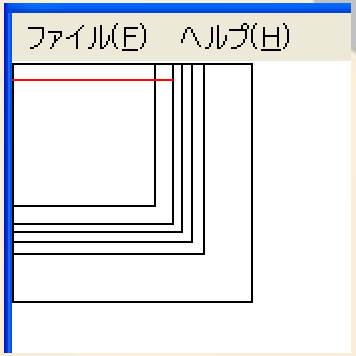
Windws XP の違い



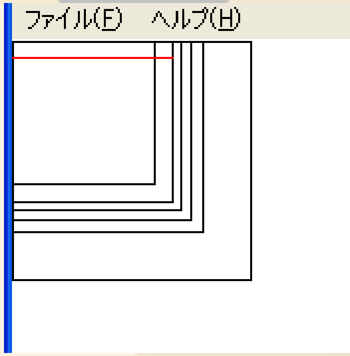
論理120DPI
真85 DPI(1024x768 15インチ)
ペイント90DPI



論理96DPI
真85 DPI(1024x768 15インチ)
ペイント90DPI

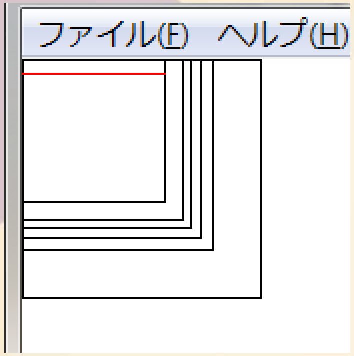


論理120DPI
真?DPI(Virtual PC)
ペイント81DPI

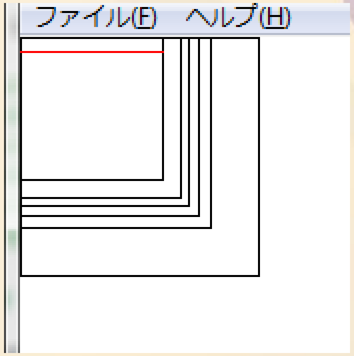


論理96DPI
真?DPI(Virtual PC)
ペイント81DPI

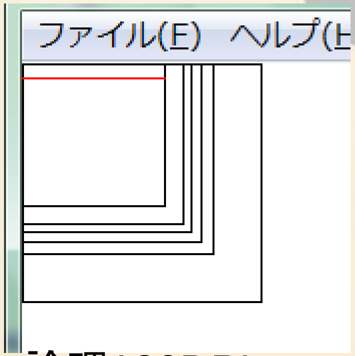
Windws Vista の違い



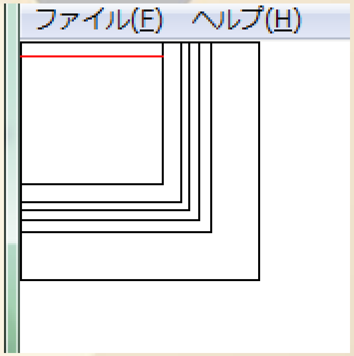
論理120DPI
真94 DPI(1920x1200 24インチ)
ペイント72DPI



論理96DPI
真94 DPI(1920x1200 24インチ)
ペイント72DPI



論理120DPI
真105 DPI(1024x768 12.1インチ)
ペイント72DPI



論理96DPI
真105 DPI(1024x768 12.1インチ)
ペイント72DPI

GDIにおける1インチとは

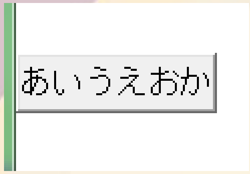
- 1インチ = GetDeviceCaps(hdc, HORZRES) / (GetDeviceCaps(hdc, HORZSIZE) / 25.4)
- XPでは環境による。Vistaでは72固定
- この数値を簡単に知るにはペイントを使う。
- 決して真のDPIではない。

- ※ GetDeviceCaps(hdc, HORZRES)ピクセル単位の画面の幅
- ※ GetDeviceCaps(hdc, HORZSIZE)mm単位の画面の物理幅（正確ではない）

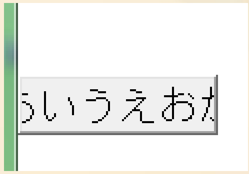
正確に1インチを描画するには？

- ユーザーに情報を入力してもらう(モニタのサイズ、現在の総ピクセル数）。
- マッピングモードMM_TEXTで、上記情報より正確なピクセル数を計算する。
- 他のマッピングモードでは、論理的な1インチになってしまう。

DPI変更の問題



96DPI



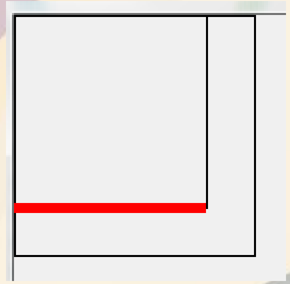
120DPI

- ボタンなどのサイズはピクセル単位でしか指定できない。
- フォントのサイズは論理DPIに基づく。
- 故にDPIを変更するとフォントだけ大きくなる。

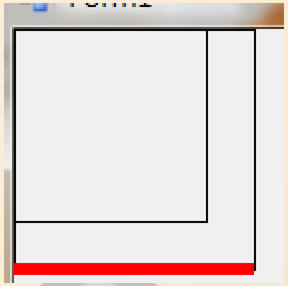
Windowsフォーム描画（OnPaint の処理）

```
Graphics g = e.Graphics;
g.DrawRectangle(p, 0, 0, 96, 96);
g.DrawRectangle(p, 0, 0, 120, 120);

g.PageUnit = GraphicsUnit.Inch;
Pen p2 = new Pen(Color.Black, 0.05f);
g.DrawLine(p2, 0, 1, 1, 1);
```

Windowsフォームの DPI による違い



96DPI

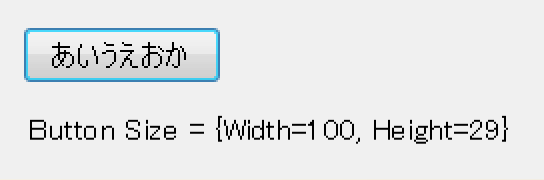


120DPI

- Windows Formsにおける1インチは論理DPIに基づく。

Windowsフォームの自動スケーリング

96DPI



120DPI

- 開発時点のDPIやシステムフォントの大きさを記録しておき、実行時にそれらを調べて、各要素のサイズを調整する。
- 開発時点の情報を記録するので、できれば各開発環境のDPIやシステムフォントの大きさは揃えておいた方が良い。
- http://msdn2.microsoft.com/ja-jp/library/ms229605.aspx
Windowsフォームにおける自働スケーリング

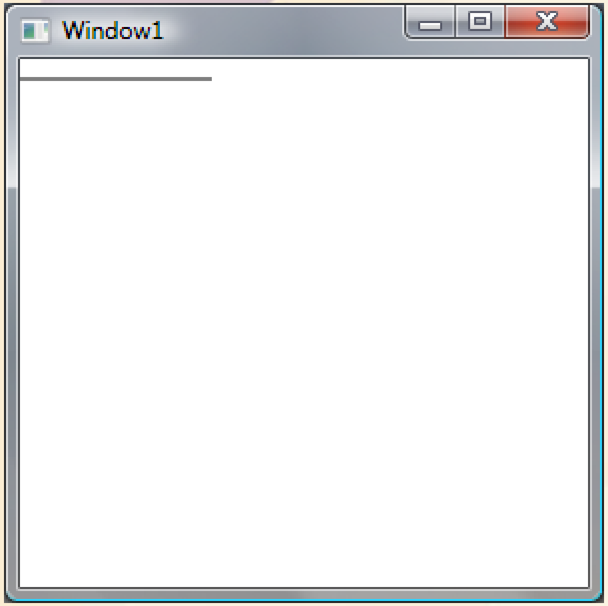
WPF に導入された新しい単位 -デバイス非依存ピクセル

- どのような環境であれ、1デバイス非依存ピクセルは1/96インチとなる。
- 描画、コントロールのサイズ、フォントのサイズ、あらゆる測定がデバイス非依存ピクセルとなる。
- 96DPIの環境で、96デバイス非依存ピクセルの直線は96物理ピクセル
- 120DPIの環境で、96デバイス非依存ピクセルの直線は120物理ピクセル
- フォントのサイズ指定がこれまでと違う！
- Silverlight は相変わらず1単位=1物理ピクセル

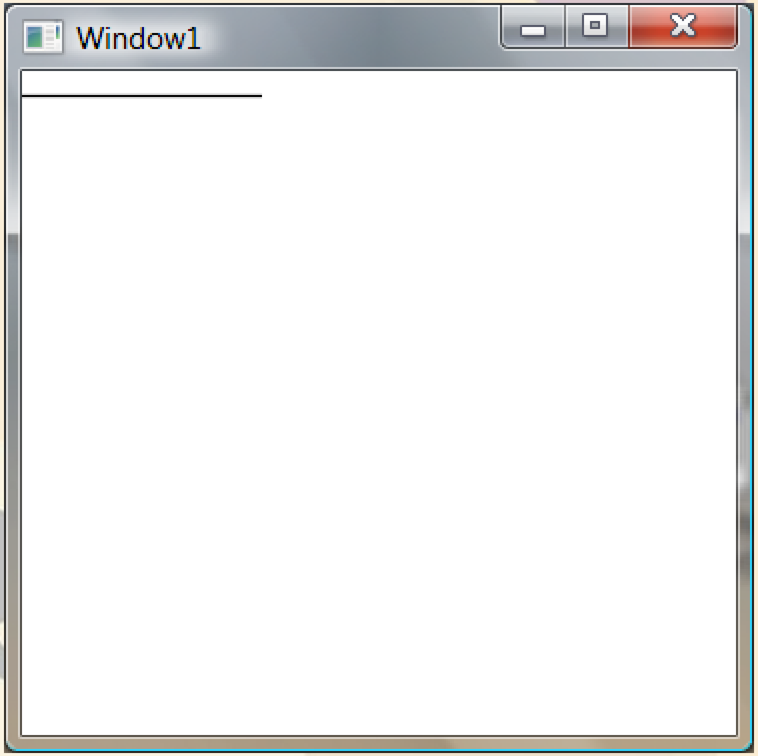
WPF で直線を描画

```
<Window x:Class="WpfApplication1.Window1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Window1" Height="300" Width="300">
    <Grid>
          <Line X1="0" X2="96" Y1="10" Y2="10" Stroke="Black" />
    </Grid>
</Window>
```

WPF DPI の違い



96DPI



120DPI