

松山大学 経営学部

# 情報処理論（応用）



第18回 グラフィックス処理（7）



講師 檀 裕也

<http://www.cc.matsuyama-u.ac.jp/~dan/application/>

2006年11月28日

# 出席確認

- 出席確認フォームから学籍番号および氏名を送信せよ。

## 出席確認フォーム

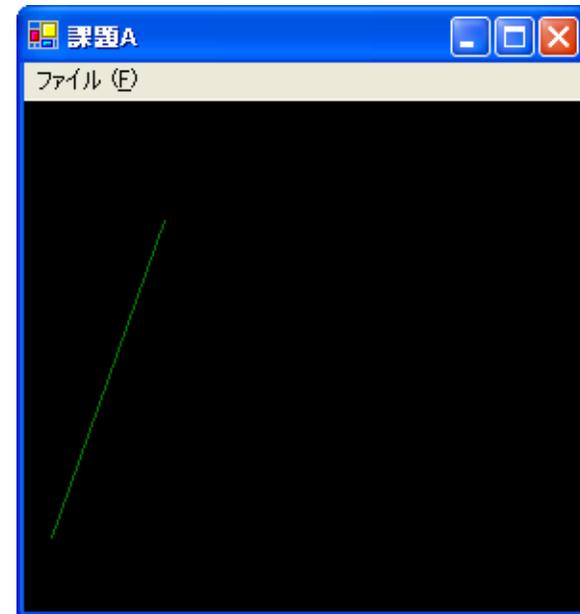
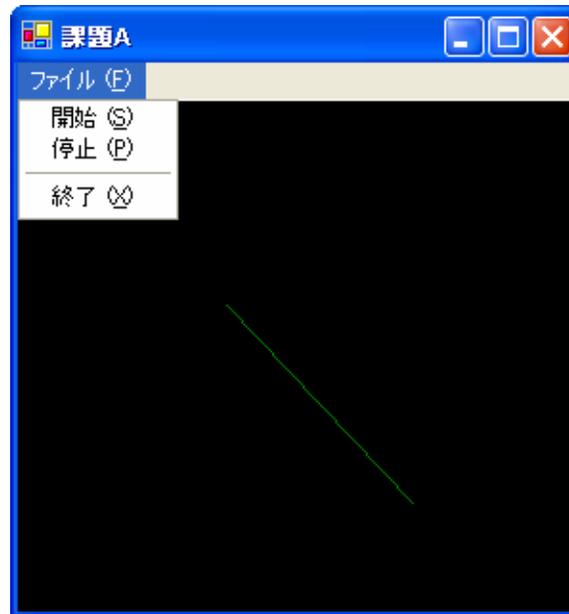
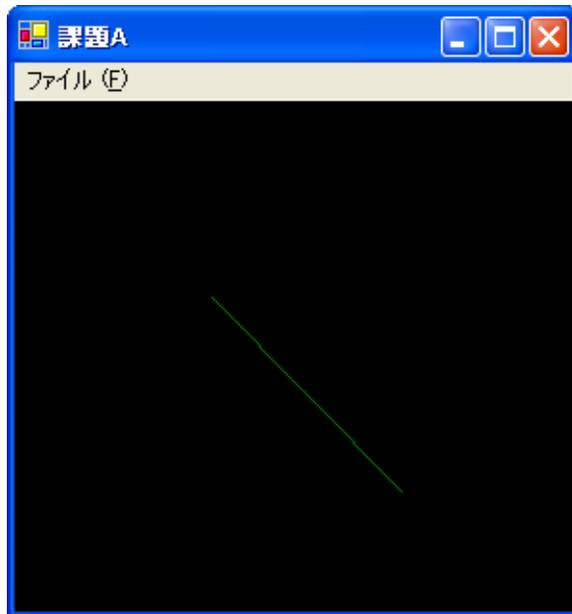
<http://www.cc.matsuyama-u.ac.jp/~dan/application/attendance.html>

# 今回の予定

- 演習課題（解答例）
  - 動く線分
  - グラデーション
  - 回転する正三角形
- 到達目標
  - Windows アプリケーションにおいて、与えられた仕様のグラフィックス処理を実装できる。

# 課題A

- 画面上を線分が動くアニメーションプログラム
  - [開始] コマンドで動作を開始する
  - [停止] コマンドで動作を停止する



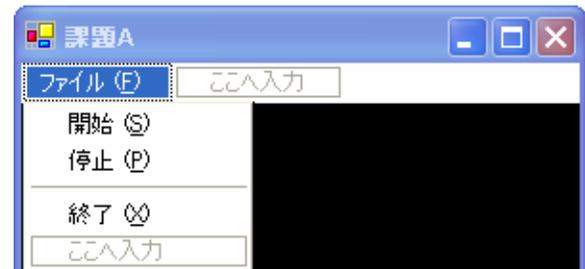
# フォームの設計

- PictureBox を配置し、 MainMenu および Timer を追加する。



## PictureBox1

- Dock: Fill
- BackColor: Black



## MainMenu1

- MenuItem1: ファイル (&F)
- MenuItem2: 開始 (&S)
- MenuItem3: 停止 (&P)
- MenuItem4: -
- MenuItem5: 終了 (&X)

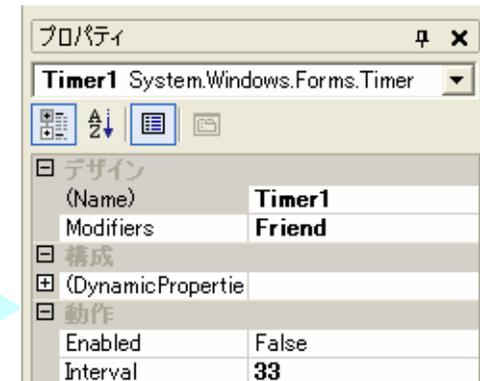
# Timer の利用

- Timer コンポーネントを追加し、Interval プロパティに値（33ミリ秒）を設定する。



Timer1

•Interval: 33



# コード (前半)

```
Public Class Form1
    Inherits System.Windows.Forms.Form
    ' グローバル変数の宣言
    Dim x1, y1 As Integer
    Dim x2, y2 As Integer
    Dim vx1, vy1 As Integer
    Dim vx2, vy2 As Integer
End Class

Windows フォーム デザイナで生成されたコード

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    ' グローバル変数の初期化
    x1 = 100 : y1 = 100
    x2 = 200 : y2 = 200
    vx1 = 3 : vy1 = 2
    vx2 = 2 : vy2 = 3
End Sub

Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem2.Click
    ' [開始] コマンドの処理
    Timer1.Start()
End Sub

Private Sub MenuItem3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem3.Click
    ' [停止] コマンドの処理
    Timer1.Stop()
End Sub

Private Sub MenuItem5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem5.Click
    ' [終了] コマンドの処理
    Me.Dispose()
End Sub

Private Sub PictureBox1_Paint(ByVal sender As Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles PictureBox1.Paint
    ' 線分の描画
    e.Graphics.DrawLine(New Pen(Color.Green, 1), x1, y1, x2, y2)
End Sub
```

コロン記号で命令文を区切る

# コード (後半)

- 座標の計算は Timer1 の Tick イベントハンドラに記述する。

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    ' 端点1のx方向の衝突判定および移動の処理
    If x1 + vx1 <= 0 Or x1 + vx1 >= PictureBox1.Width Then
        vx1 = -vx1
    End If
    x1 = x1 + vx1

    ' 端点1のy方向の衝突判定および移動の処理
    If y1 + vy1 <= 0 Or y1 + vy1 >= PictureBox1.Height Then
        vy1 = -vy1
    End If
    y1 = y1 + vy1

    ' 端点2のx方向の衝突判定および移動の処理
    If x2 + vx2 <= 0 Or x2 + vx2 >= PictureBox1.Width Then
        vx2 = -vx2
    End If
    x2 = x2 + vx2

    ' 端点2のy方向の衝突判定および移動の処理
    If y2 + vy2 <= 0 Or y2 + vy2 >= PictureBox1.Height Then
        vy2 = -vy2
    End If
    y2 = y2 + vy2

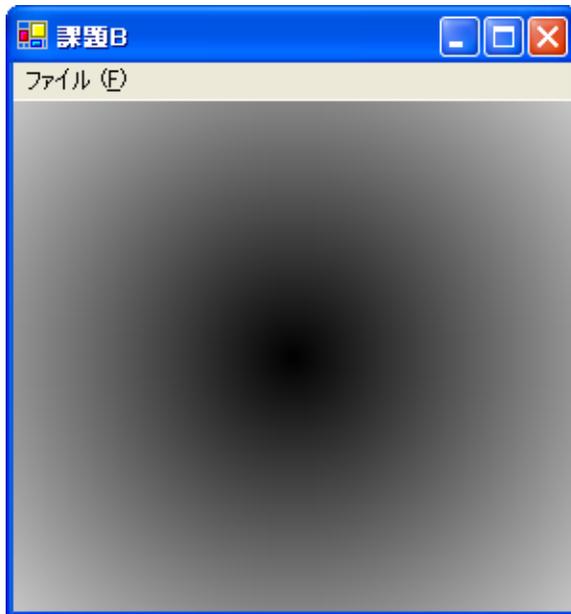
    ' 最新の画面情報に更新
    PictureBox1.Refresh()
End Sub
End Class
```

# コードの解説

- 一方の点の座標を  $(x1, y1)$ 、他方の点の座標を  $(x2, y2)$  で表し、これらの点が画面の表示領域からはみ出さないように処理する。
- なお、速度はそれぞれ  $(vx1, vy1)$  および  $(vx2, vy2)$  によって表す。
- これらの変数をグローバル変数として宣言することで、複数のプロシージャから読み書きできる。

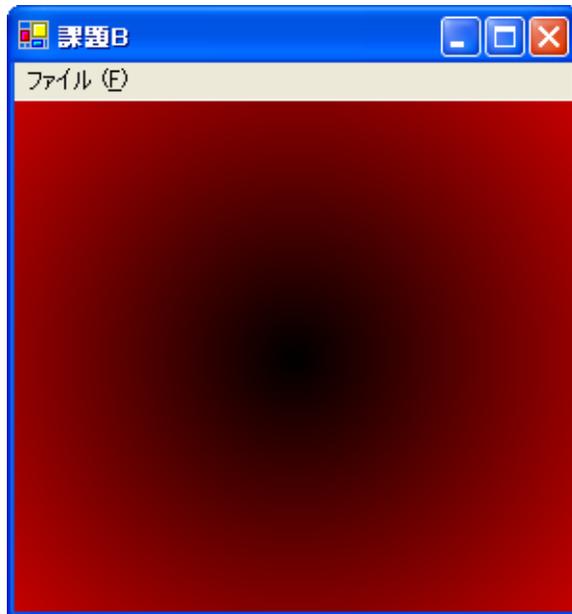
# 課題 B

- 光の三原色のグラデーション表示
  - プログラム起動時はグレーのグラデーションを表示
  - 各色のコマンドで赤・緑・青のグラデーションを表示



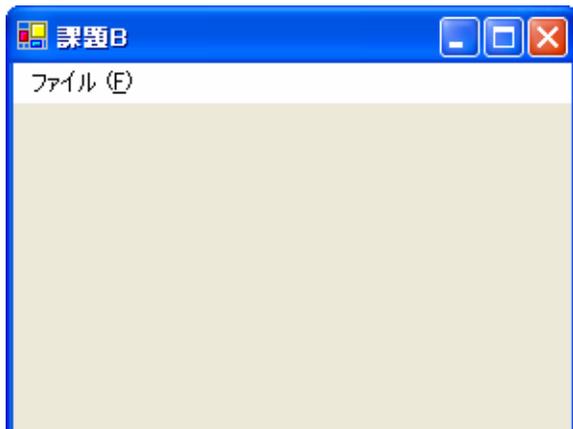
# 課題 B

## 三色の表示例



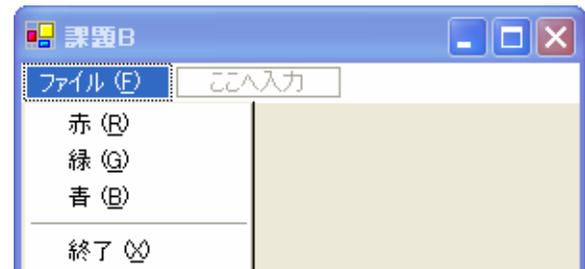
# フォームの設計

- PictureBox を配置し、 MainMenu を追加する。



PictureBox1

- Dock: Fill



MainMenu1

- MenuItem1: ファイル (&F)
- MenuItem2: 赤 (&R)
- MenuItem3: 緑 (&G)
- MenuItem4: 青 (&B)
- MenuItem5: -
- MenuItem6: 終了 (&X)

# コード (前半)

- RGB各色の割合でグラデーションの色を指定

```
Public Class Form1
    Inherits System.Windows.Forms.Form
    ' グローバル変数の宣言
    Dim Red, Green, Blue As Integer
End Class

Windows フォーム デザイナで生成されたコード

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    ' プログラム起動時の処理 (灰色グラデーション)
    Red = 1
    Green = 1
    Blue = 1
End Sub

Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem2.Click
    ' [赤] コマンドの処理
    Red = 1
    Green = 0
    Blue = 0
    PictureBox1.BackColor = Color.Red
    PictureBox1.Refresh()
End Sub

Private Sub MenuItem3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem3.Click
    ' [緑] コマンドの処理
    Red = 0
    Green = 1
    Blue = 0
    PictureBox1.BackColor = Color.Lime
    PictureBox1.Refresh()
End Sub
```

# コード (後半)

- Paint イベントでグラフィックスの描画処理

```
Private Sub MenuItem4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem4.Click
    ' [青] コマンドの処理
    Red = 0
    Green = 0
    Blue = 1
    PictureBox1.BackColor = Color.Blue
    PictureBox1.Refresh()
End Sub

Private Sub MenuItem6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem6.Click
    ' [終了] コマンドの処理
    Me.Dispose()
End Sub

Private Sub PictureBox1_Paint(ByVal sender As Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles PictureBox1.Paint
    Dim i As Integer
    Dim cx, cy As Integer
    Dim brush As Brush

    ' グラフィックス表示領域の中心座標を求める
    cx = PictureBox1.Width / 2
    cy = PictureBox1.Height / 2

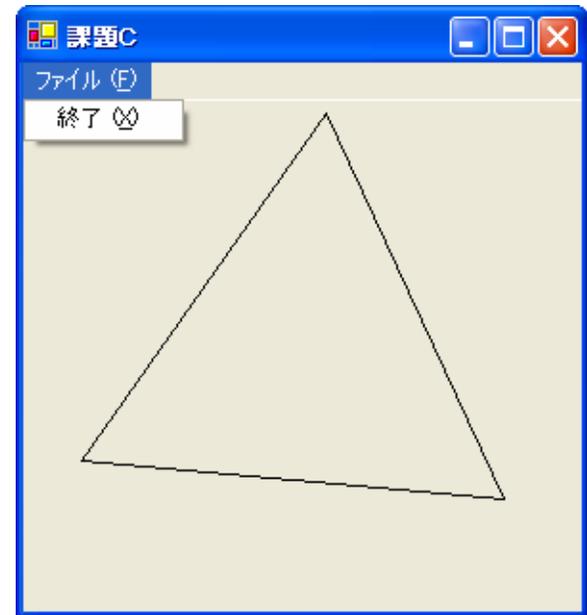
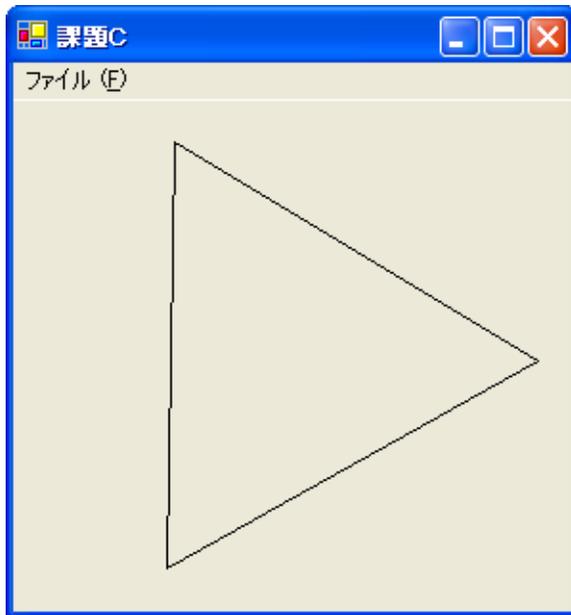
    ' グラデーションの描画
    For i = 255 To 0 Step -1
        brush = New SolidBrush(Color.FromArgb(Red * i, Green * i, Blue * i))
        e.Graphics.FillEllipse(brush, cx - i, cy - i, i * 2, i * 2)
        brush.Dispose()
    Next
End Sub
End Class
```

# コードの解説

- 赤み成分を Red、緑色の成分を Green、青み成分を Blue で表す。(各値は0または1)
- カウンタ変数 i の値を 0 から 255 までの範囲に変えることで、グラデーションを実現する。
  - `Color.FromArgb (Red * i, Green * i, Blue * i)`
- 描画の順序に注意して、ブラシを使って外側の領域から円で塗りつぶす。
  - ペンを使った `DrawEllipse` ではどうなるか？

# 課題C

- 回転する正三角形のアニメーションプログラム



# フォームの設計

- PictureBox を配置し、 MainMenu を追加する。



PictureBox1

- Dock: Fill



MainMenu1

- MenuItem2: 終了 (&X)

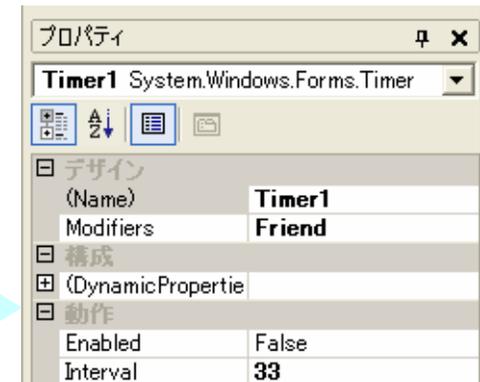
# Timer の利用

- Timer コンポーネントを追加し、Interval プロパティに値（33ミリ秒）を設定する。



Timer1

•Interval: 33



# コード (前半)

- プログラム全体を通して必要な変数は、回転角を表す `theta`、正三角形の3つの頂点の座標 `(x1, y1)`、`(x2, y2)`、`(x3, y3)` である。

```
Public Class Form1
    Inherits System.Windows.Forms.Form
    ' グローバル変数の宣言
    Dim theta As Double ' 回転角の大きさ
    Dim x1, y1 As Integer ' 頂点1の座標
    Dim x2, y2 As Integer ' 頂点2の座標
    Dim x3, y3 As Integer ' 頂点3の座標

    Windows フォーム デザイナで生成されたコード

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        ' タイマー開始
        Timer1.Start()
    End Sub

    Private Sub PictureBox1_Paint(ByVal sender As Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles P
        ' 正三角形の3つの辺を描画する
        e.Graphics.DrawLine(New Pen(Color.Black, 10), x2, y2, x3, y3)
        e.Graphics.DrawLine(New Pen(Color.Black, 10), x3, y3, x1, y1)
        e.Graphics.DrawLine(New Pen(Color.Black, 10), x1, y1, x2, y2)
    End Sub
```

# コード (後半)

- 一定の時間間隔で少しずつ回転させる。

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    Dim cx, cy As Integer ' 描画領域の中心座標

    ' 回転角の計算
    theta += Math.PI / 128
    If theta > 2 * Math.PI Then
        theta = theta - 2 * Math.PI
    End If

    ' 中心座標を求める
    cx = PictureBox1.Width / 2
    cy = PictureBox1.Height / 2

    ' 3つの頂点の座標を計算する
    x1 = cx + 128 * Math.Cos(theta)
    y1 = cy + 128 * Math.Sin(theta)
    x2 = cx + 128 * Math.Cos(theta + 2 * Math.PI / 3)
    y2 = cy + 128 * Math.Sin(theta + 2 * Math.PI / 3)
    x3 = cx + 128 * Math.Cos(theta + 4 * Math.PI / 3)
    y3 = cy + 128 * Math.Sin(theta + 4 * Math.PI / 3)

    ' 画面を描画する
    PictureBox1.Refresh()
End Sub

Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem2.Click
    Me.Dispose()
End Sub
End Class
```

# まとめ

- これまでに学習した Visual Studio .NET の機能を使って、与えられた仕様に基づくプログラムを作成した。

# 次回予定

- グラフィックス ( 8 )
  - 画像ファイルの処理 ( 予定 )