[OpenGL:1]　OpenGL とは

１．OpenGL とは何か？

米国 Silicon Graphics 社(SGI)が中心となって開発した 3 次元グラフィックスライブラリである．SGI はグラフィックスに特化したワークステーション（GWS）の開発を積極的に行い，自社の GWS 上で稼動するグラフィックスライブラリ IRIS GL を開発したが，その後，この仕様を公開したライブラリとして OpenGL が開発され，1992 年に，OpenGL Architecture Review Board(ARB)により監修されることとなる．（現在は標準化団体 The Khronos Group，OpenGL ARB Working Group で管理されている）

OpenGL は 3 次元コンピュータグラフィックス環境を容易に構築するために開発された言語であり，高度なリアルタイムでの 3 次元 CG の描画を実現している．また Open なライブラリとして様々なプラットフォーム上でのライブラリが提供されている．その利用範囲は広く，CAD，3 次元 CG ソフトウェア，3D ゲーム，技術計算の可視化等のさまざまな分野で利用されている．

※OpenGL に関する公式サイト　http://www.opengl.org/

2.　VB による OpenGL の開発環境

(1) Windows 上での OpenGL 関連ライブラリ

・opengl32.dll　　OpenGL 本体　※Windows に標準で添付
・glu32.dll　　OpenGL Utility Library　※Windows に標準で添付
・glut32.dll　　OpenGL Programming Guide Auxiliary Library
　または freeglut.dll　　同上

(2)OpenGL の開発環境

OpenGL は API (Application Programming Interface)として提供され，C, C++, Fortran, Java 等の様々な言語からの操作が可能である．この講義では，VisualBasic の環境からの OpenGL を操作することを試みる．

なお，VB から OpenGL を操作するために，さらにライブラリを介する必要があり，今回は The Tao Framework によるライブラリを利用する．

3. OpenGL を利用するための準備作業

1) Tao Framework をダウンロードしてインストールする.

http://sourceforge.net/projects/taoframework/

Files -> 2.1.0 へ移動し，taoframework-2.1.0-setup.exe を
ダウンロードし，実行する.
（TaoFramework がインストールされる.）
学内の場合は，http://w3.campus.myu.ac.jp/~makanae/tao/ からでも可能

2) PC の OS（Windows）が 32 ビットであるか，64 ビットであるかを確認する.

コントロールパネル -> システムとセキュリティ -> システム で
システムの種類を確認 (XX ビットオペレーティングシステム)

3) ライブラリファイルのコピー

インストールしたディレクトリ（通常は C:¥Program Files(x86)¥TaoFramework）の中の lib ディレクトリから freeglut.dll をコピーして C:¥Windows¥SysWOW64 に貼り付ける.

コピー: C:¥Program Files(x86)¥TaoFramework¥lib¥freeglut.dll
貼り付け:
【OS が 64 ビットの場合】C:¥Windows¥SysWOW64
【OS が 32 ビットの場合】C:¥Windows¥System32

4) VB2010 での設定方法　（新しいプログラムを作成する場合，毎回行う必要がある）

a) 新しいプロジェクトを作成する.

b) プロジェクト >> 参照の追加 >> 参照 で,

C:¥Program Files¥TaoFramework¥bin から以下の DLL を選択し OK.

Tao.OpenGl.dll, Tao.FreeGlut.dll

※ オ ブジェクトブラウザ (F2 キーを押す)上で,

Tao.OpenGl, Tao.Freeglut

が表示されていれば参照は正しく追加されている.

４．OpenGL プログラムプログラム

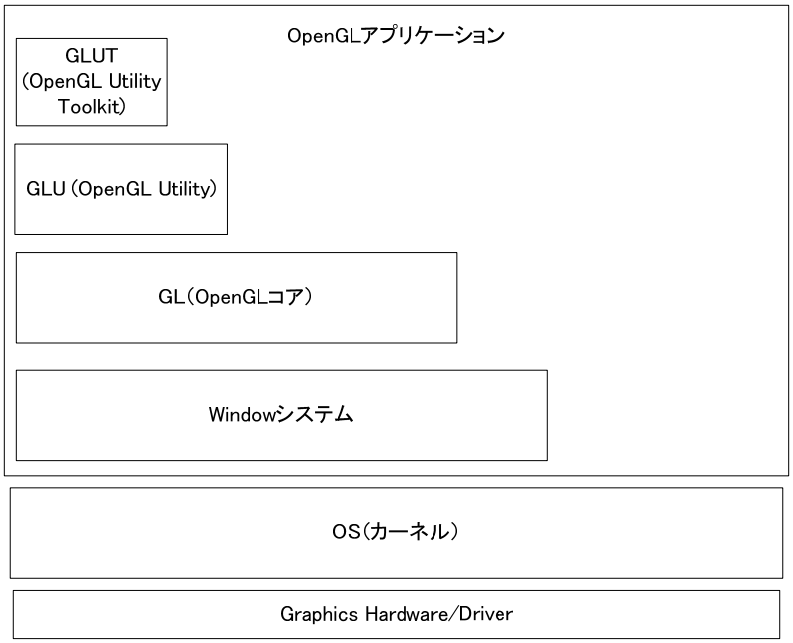
```
Imports Tao.OpenGl.Gl
Imports Tao.OpenGl.Glu
Imports Tao.FreeGlut.Glut

Public Class Form1
    Private Structure Axis
        Dim x As Double
        Dim y As Double
        Dim z As Double
    End Structure

    Dim tr As Axis
    Dim rot As Axis
    Dim sc As Axis

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As  _
                System.EventArgs) Handles MyBase.Load


        glutInitDisplayMode(GLUT_RGB Or GLUT_DOUBLE Or GLUT_DEPTH)
        glutInit()

        glutInitWindowPosition(200, 200)
        glutInitWindowSize(500, 500)
        glutCreateWindow("smallest")

        glutDisplayFunc(New DisplayCallback(AddressOf Display))
        glutReshapeFunc(New ReshapeCallback(AddressOf Reshape))
        glutMotionFunc(New MotionCallback(AddressOf MotionEvent))
        glutMainLoop()
    End Sub

    Sub Display()
        glClear(GL_COLOR_BUFFER_BIT)
        glColor3f(1.0#, 1.0#, 1.0#)
        glLoadIdentity()
```

```
        glTranslatef(0.0#, 0.0#, -5.0#)
        glRotatef(rot.y, 0, 1, 0)
        glRotatef(rot.x, 1, 0, 0)
        glScalef(1.0#, 1.0#, 1.0#)
        glutWireCube(1.0#)
        glutSwapBuffers()
    End Sub

    Sub Reshape(ByVal nWidth As Integer, ByVal nHeight As Integer)
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()
        gluPerspective(50, nWidth / nHeight, 1, 100)
        glMatrixMode(GL_MODELVIEW)
        glLoadIdentity()
        glViewport(0, 0, nWidth, nHeight)
    End Sub

    Sub MotionEvent(ByVal x As Integer, ByVal y As Integer)
        Static old As Axis

        rot.x = rot.x + (y - old.y)
        rot.y = rot.y + (x - old.x)
        Display()

        old.x = x : old.y = y
    End Sub
End Class
```

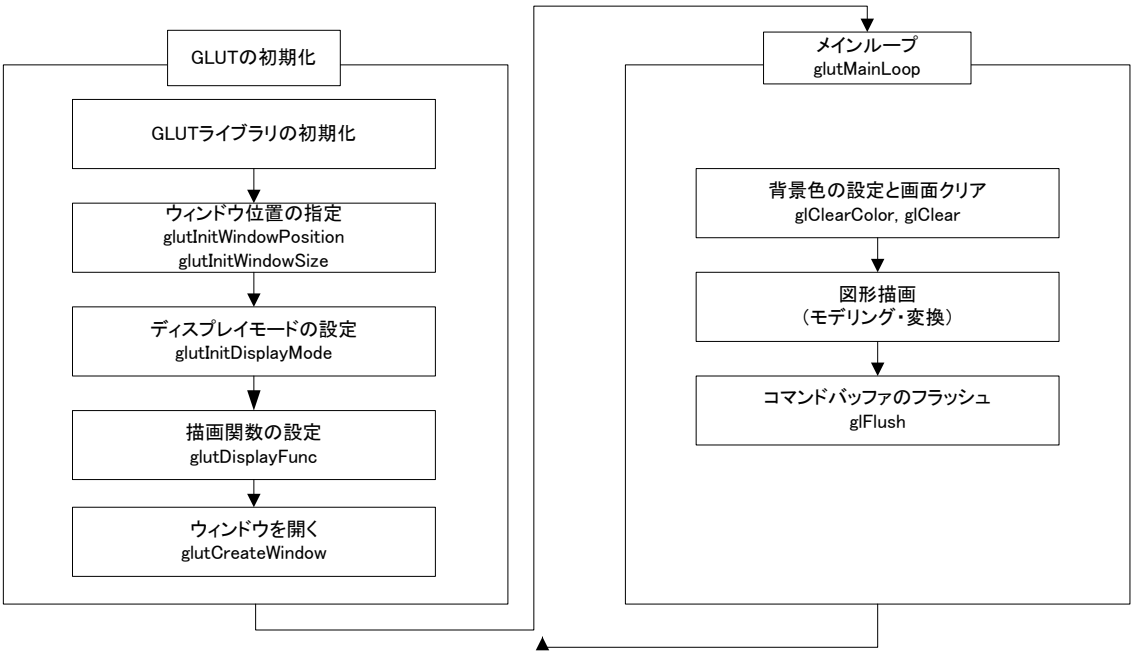
（演習）
上のサンプルプログラムを入力し，自分のPCでOpenGLが正常に動作することを確認しなさい．
（必ずやっておくこと！）

[OpenGL:2]　OpenGL の制御とプログラミング

1．Windows 上での OpenGL の制御



2. OpenGL のプログラム構造



3. OpenGL の命令および定数について

(1)命令の接頭子

gl ： OpenGL 自体の命令を表す(opengl32.dll)

glu： OpenGL のユーティリティの命令を表す(glu32.dll)

glut: OpenGL Utilitiy Toolkit の命令を表す

(glut32.dll あるいは freeglut.dll にリンク)

(2)命令の接尾子

引数の数と型を表す接尾子を設けている.

b: signed char, ub: unsigned char, s: short us: unsigned short

i: in, ut: unsigned int, f:float, d: double

(3)OpenGL の定数

OpenGL には定義済みの定数をもっており，接頭子 GL_（全て大文字）で始まる.

（補足）C 言語への移植に関する参考資料

http://fly.cc.fer.hr/~unreal/theredbook/

http://www.exa-corp.co.jp/solutions/ubiquitous/ubiquitous-solution/

4. OpenGL プログラミング[簡単なプログラミング]（平行投影，透視投影）

(1)平行投影

```
Imports Tao.OpenGl.Gl
Imports Tao.OpenGl.Glu
Imports Tao.FreeGlut.Glut
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _
                                        System.EventArgs) Handles
                                        MyBase.Load
        glutInit()
        glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH)
        glutInitWindowSize(500, 500)
        glutInitWindowPosition(100, 100)
        glutCreateWindow("ortho")

        glutDisplayFunc(New DisplayCallback(AddressOf Draw))
        glutReshapeFunc(New ReshapeCallback(AddressOf ViewSet))
        glutMainLoop()
    End Sub

    Private Sub ViewSet()
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()
        glOrtho(-1, 1, -1, 1, -1, 1)
        glMatrixMode(GL_MODELVIEW)
    End Sub

    Public Sub Draw()
        glClearColor(0, 0, 0, 0)
        glClear(GL_COLOR_BUFFER_BIT)
        glColor3f(1.0#, 1.0#, 1.0#)
        glLoadIdentity()

        glBegin(GL_POLYGON)
            glVertex2f(-0.5, -0.5)
            glVertex2f(-0.5, 0.5)
            glVertex2f(0.5, 0.5)
            glVertex2f(0.5, -0.5)
        glEnd()

        glFlush()
    End Sub
End Class
```

※VB のエディタの設定：

ツール->オプション->テキストエディタ　インデントをブロックに変更しておく

(2)透視投影

```
Imports Tao.OpenGl.Gl
Imports Tao.OpenGl.Glu
Imports Tao.FreeGlut.Glut

Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _
                                   System.EventArgs) Handles MyBase.Load
        Dim width As Integer = 500
        Dim height As Integer = 500

        glutInit()
        glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH)
        glutInitWindowSize(width, height)
        glutInitWindowPosition(100, 100)
        glutCreateWindow("perspective")
        glutDisplayFunc(New DisplayCallback(AddressOf Draw))
        ViewSet(50, width / height, 1, 100)
        glutMainLoop()
    End Sub

    Private Sub ViewSet(ByVal fov As Single, ByVal aspect As Single,
                                ByVal near As Single, ByVal far As Single)
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()
        gluPerspective(fov, aspect, near, far)
        glMatrixMode(GL_MODELVIEW)
    End Sub

    Public Sub Draw()
        glClearColor(0, 0, 0, 0)
        glClear(GL_COLOR_BUFFER_BIT)
        glColor3f(1.0#, 1.0#, 1.0#)
        glLoadIdentity()
        glTranslatef(0.0#, 0.0#, -5.0#)
        glScalef(1.0#, 1.0#, 1.0#)
        glutWireCube(1)
        glFlush()
    End Sub
End Class
```

（問題）
1．4.(1)を基に，glBegin，glVertex，glEnd を用いて，任意の三角形を描画するプログラムを作成しなさい． GL_POLYGON の代わりに GL_LINE_LOOP についても試してみること．
（探求） glRotatef glscalef を試す．
2．(2)により描かれる立方体をその中心で Y 軸周りに 30 度回転させるようにコードを変更しなさい．
3．(2)の立方体の高さを 2 とした直方体を glScalef により描画しなさい．
4．任意の 3 次元図形を作成し，透視投影を用いて描画しなさい（ワイヤフレームでよい）．

上記 1， 2， 3， 4 で作成したプログラムと出力画像を PDF ファイルにまとめ，Moodle を通じて提出すること．

glBegin..glVertex..glEnd による図形描画に関しては，下記資料を参考にすること．
http://www.glprogramming.com/red/chapter02.html

[OpenGL:3]射影変換とモデリング変換

1. 変換行列(Matrix)の設定

glMatrixMode を設定する >> 使用する行列を設定する.

射影変換をする場合　　glMatrixMode(GL_PROJECTION)

モデリング変換をする場合　　glMatrixMode(GL_MODELVIEW)

2. 射影変換

平行投影（正射影）

glOrtho(left, right, bottom, top, near, far)

例) glOrtho(-2,2,-2,2,-1,1)

透視投影

glFrustum(left, right, bottom, top, near, far)

left,right,bottom,top はクリッピング面の座標

near,far は遠近クリッピング面の視点からの距離

gluPerspective(fovy, aspect,zNear,zFar)

fovy　　: 視界角度 0～180°

aspect　　: width / height

zNear, zFar: クリッピング面の負の z 軸に沿った距離

ビューポートの設定

定義されたウィンドウに対するビューポートの大きさを定義する.

glViewport(x, y, width, height)

例) glViewport(0, 0, width, height)

３．モデリング変換

移動　　glTranslate{fd}(x, y, z)

例) glTranslatef(0.0, 0.5, 0.5)

回転　　glRotate{fd}(angle, x, y, z)

例) glRotatef(45.0, 0.0, 1.0, 0.0)　'Y 軸まわりに 45 度回転

拡大・縮小　glScale{fd}(x, y, z)

例) glScalef(2.0, -0.5, 1.0)

注意)移動・回転・縮小の順序により，描画される図形は異なるので要注意

視界変換：視点位置の設定と回転は，すべての物体の移動・回転と考えればよい.

４．モデリング変換の実際

(0)基本プログラム

```
' Form1 上に ListBox1, Button1 を配置する
Imports Tao.OpenGl.Gl
Imports Tao.OpenGl.Glu
Imports Tao.FreeGlut.Glut
Public Class Form1
    Private Sub ViewSet(ByVal width As Integer, ByVal height As Integer)
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()
        gluPerspective(50, width / height, 1, 100)
        glViewport(0, 0, width, height)
        glMatrixMode(GL_MODELVIEW)
    End Sub

    Public Sub Draw()
        Draw0()  ' 実行する Draw メソッドを選択 (Draw0 - Draw8)
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, _
                       ByVal e As System.EventArgs) Handles MyBase.Load
        Button1.Text = "Draw1"
        Button1.Enabled = False

        glutInit()
        glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH)
        glutInitWindowSize(500, 500)
        glutInitWindowPosition(100, 100)
        glutCreateWindow("gl04")
        glutDisplayFunc(New DisplayCallback(AddressOf Draw))
        glutReshapeFunc(New ReshapeCallback(AddressOf ViewSet))

        Me.Show()
        Button1.Enabled = True
        glutMainLoop()
    End Sub

    ' 0. 基本プログラム
    Public Sub Draw0()
        glClearColor(0, 0, 0, 0)
        glClear(GL_COLOR_BUFFER_BIT)
        glColor3f(1.0#, 1.0#, 1.0#)
        glLoadIdentity()
        glTranslatef(0.0#, 0.0#, -3.0#)
        glutWireCube(0.5)
        glFlush()
    End Sub

End Class
```

(1) コマンドボタンを押すことにより物体を移動する

```
Dim tz As Single

Public Sub Button1_Click() Handles Button1.Click
    tz += 0.5
    Draw1()
End Sub

 ' 1. コマンドボタンを押すことにより物体を移動する
Public Sub Draw1()
    glClearColor(0, 0, 0, 0)
    glClear(GL_COLOR_BUFFER_BIT)
    glColor3f(1.0#, 1.0#, 1.0#)
    glLoadIdentity()
    glTranslatef(0.0#, 0.0#, -10 + tz)
    glutWireCube(0.5)
    glFlush()
End Sub
```

(2)等間隔で物体を配置する

```
Public Sub Draw2()
    glClearColor(0, 0, 0, 0)
    glClear(GL_COLOR_BUFFER_BIT)
    glColor3f(1.0#, 1.0#, 1.0#)
    glLoadIdentity()
    glTranslatef(0.0#, 0.1, -5.0#)
    glutWireCube(0.5)
    For n As Integer = 1 To 100
        glTranslatef(0.3, 0.2, 0)
        glutWireCube(0.5)
    Next
    glFlush()
End Sub
```

(3)円周上に物体を配置する

```
Public Sub Draw3()
    glClearColor(0, 0, 0, 0)
    glClear(GL_COLOR_BUFFER_BIT)
    glColor3f(1.0#, 1.0#, 1.0#)
    glLoadIdentity()
    glRotatef(10, 1, 0, 0)
    glTranslatef(0.0#, -3, -5.0#)
    glutWireCube(0.5)
    For n As Integer = 1 To 100
        glRotatef(2, 0, 1, 0)
        glTranslatef(0, 0, -0.5)
        glutWireCube(0.5)
    Next
    glFlush()
End Sub
```

(4)図形を任意の座標を中心に回転させる[1]

```
Public Sub Draw4()
    glClearColor(0, 0, 0, 0)
    glClear(GL_COLOR_BUFFER_BIT)
    glColor3f(1.0#, 1.0#, 1.0#)
    glLoadIdentity()
    glTranslatef(0.0#, 0, -5.0#)
    glutWireCube(1)
    glRotatef(45, 0, 0, 1)
    glScalef(0.2, 2, 0.2)
    glutWireCube(0.5)
    glFlush()
End Sub
```

(5)図形を任意の座標を中心に回転させる[2]

```
Public Sub Draw5()
    glClearColor(0, 0, 0, 0)
    glClear(GL_COLOR_BUFFER_BIT)
    glColor3f(1.0#, 1.0#, 1.0#)
    glLoadIdentity()
    glTranslatef(0.0#, 0, -5.0#)
    glutWireCube(1)
    glTranslatef(0, -0.5, 0)  '物体
    glRotatef(45, 0, 0, 1)
    glTranslatef(0, 0.5, 0)
    glScalef(0.2, 2, 0.2)
    glutWireCube(0.5)
    glFlush()
End Sub
```

５．Pushmatrix と Popmatrix

```
glPushMatrix        現在の座標変換行列を記憶させる.
glPopMatrix         記憶していた座標変換行列に戻す.
```

(1) PushMatrix，PopMatrix を用いない場合の問題点

```
Public Sub Draw6()
    glClearColor(0, 0, 0, 0)
    glClear(GL_COLOR_BUFFER_BIT)
    glColor3f(1.0#, 1.0#, 1.0#)
    glLoadIdentity()
    glRotatef(10, 1, 0, 0)
    glTranslatef(0.0#, -3, -5.0#)
    glScalef(0.2, 2, 0.2)  'この時点で座標系のスケーリングが行なわれる.
    glutWireCube(0.5)
```

```
        'スケール変換に応じた座標系の中でのモデリングが行なわれることになる.
        For n As Integer = 1 To 100
            glRotatef(2, 0, 1, 0)
            glTranslatef(0, 0, -0.5)
            glutWireCube(0.5)
        Next

        glFlush()
    End Sub
```

(2)glPushMatrix，glPopMatrixの利用(2)

```
    Public Sub Draw7()
        glClearColor(0, 0, 0, 0)
        glClear(GL_COLOR_BUFFER_BIT)
        glColor3f(1.0#, 1.0#, 1.0#)
        glLoadIdentity()
        glRotatef(10, 1, 0, 0)
        glTranslatef(0.0#, -3, -5.0#)

        glPushMatrix()
            glScalef(0.2, 2, 0.2)
            glutWireCube(0.5)
        glPopMatrix()

        For n As Integer = 1 To 100
            glRotatef(2, 0, 1, 0)
            glTranslatef(0, 0, -0.5)
            glutWireCube(0.5)
        Next

        glFlush()
    End Sub
```

(3)glPushMatrix，glPopMatrixの利用(3)

```
    Public Sub Draw8()
        glClearColor(0, 0, 0, 0)
        glClear(GL_COLOR_BUFFER_BIT)
        glColor3f(1.0#, 1.0#, 1.0#)
        glLoadIdentity()
        glRotatef(10, 1, 0, 0)
        glTranslatef(0.0#, -3, -5.0#)

        glPushMatrix()
            glScalef(0.2, 2, 0.2)
            glutWireCube(0.5)
        glPopMatrix()


        For n As Integer = 1 To 100
            glRotatef(2, 0, 1, 0)
```

```
            glTranslatef(0, 0, -0.5)

            glPushMatrix()
                glScalef(0.2, 2, 0.2)
                glutWireCube(0.5)
            glPopMatrix()
        Next

        glFlush()
    End Sub
```

【補足】Draw0 から Draw8 をまとめたプログラムの作り方
Form1 に ListBox1 と Button1 を置く.

```
Public Class Form1
    Delegate Sub DrawDelegate()

    Dim Draws() As DrawDelegate = { _
        AddressOf Draw0, _
        AddressOf Draw1, _
        AddressOf Draw2, _
        AddressOf Draw3, _
        AddressOf Draw4, _
        AddressOf Draw5, _
        AddressOf Draw6, _
        AddressOf Draw7, _
        AddressOf Draw8 _
    }

    Private Sub ViewSet(ByVal width As Integer, ByVal height As Integer)
        （略）
    End Sub

    Public Sub Draw()
        If ListBox1.SelectedIndex >= 0 Then
            Draws(ListBox1.SelectedIndex)()
        Else
            Draws(0)()
        End If
    End Sub

    Private Sub ListBox1_SelectedIndexChanged(ByVal sender As _
         Windows.Forms.ListBox, ByVal e As System.EventArgs) _
         Handles ListBox1.SelectedIndexChanged

        If ListBox1.SelectedIndex = 1 Then
            Button1.Enabled = True
        Else
            Button1.Enabled = False
        End If

        Draw()
    End Sub
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, _
                ByVal e As System.EventArgs) Handles MyBase.Load
        Button1.Text = "Draw1"
        Button1.Enabled = False

        glutInit()
        glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH)
        glutInitWindowSize(500, 500)
        glutInitWindowPosition(100, 100)
        glutCreateWindow("gl04")
        glutDisplayFunc(New DisplayCallback(AddressOf Draw))
        glutReshapeFunc(New ReshapeCallback(AddressOf ViewSet))

        Me.Show()
        AddDrawList()   'この行を追加

        glutMainLoop()
    End Sub

    Sub AddDrawList()
        ListBox1.BeginUpdate()

        For i As Integer = 0 To Draws.Length - 1
            ListBox1.Items.Add("Draw" + i.ToString())
        Next

        ListBox1.EndUpdate()
    End Sub

   'Draw0()から Draw8()までは４．と同じため略
    Public Sub Draw0()
      （略）
    End Sub
    ・・・・（略）・・・・・・
End Class
```

（演習問題）
glTranslatef, glRotatef 等を用いて任意の図形群を描画しなさい.

※実行画面と描画部分（Draw0()に該当）のコードを印刷し，次回講義で提出

（応用問題）
OpenGL を用いて下記をシミュレートするプログラムを考えなさい.
- 太陽系
- ロボットの腕

参考となるサイト

http://www.glprogramming.com/red/

Chapter3 惑星 Example3-6 を参照

ロボットの腕 Example 3-7 を参照

[OpenGL:4]ロボットアームを作る

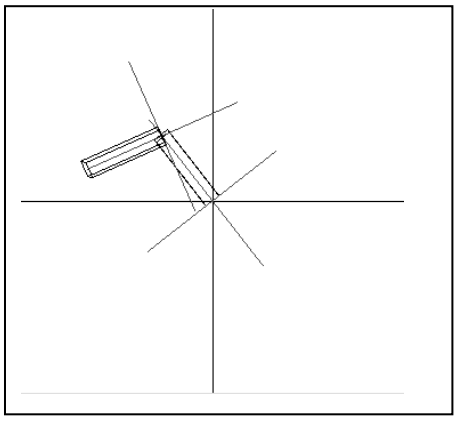
１．キネマティクス とは

キネマティクス(kinematics)とは，人間などの関節をもつ物体の動きを，階層構造により制御する方法である．物体それぞれの座標系を関連づけ，人体等の動きを表現する．

Forward Kinematics（順運動学）では，各関節点における座標系の変換の蓄積により位置の制御を行う．他方，Inverse Kinematics（逆運動学）では，複数の関節をもつ先端の位置から逆に階層をたどり，必要な制御情報を得る．

今回は，Forward Kinematics を用いてロボットアームを制御するプログラムを作成する．

２．ロボットアームを作る



コード

```
Imports Tao.OpenGl.Gl
Imports Tao.OpenGl.Glu
Imports Tao.FreeGlut.Glut

Public Class Form1
    Dim wWidth As Integer = 500
    Dim wHeight As Integer = 500
    Dim rz As Single

    Private Sub ViewSet(ByVal w As Integer, ByVal h As Integer)
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()
        gluPerspective(50, w / h, 1, 100)
        glViewport(0, 0, w, h)
        glMatrixMode(GL_MODELVIEW)
    End Sub
```

```
Public Sub Draw()
    glClearColor(0, 0, 0, 0)
    glClear(GL_COLOR_BUFFER_BIT)
    glColor3f(1.0#, 1.0#, 1.0#)
    glLoadIdentity()
    glTranslatef(0.0#, 0.0#, -5.0#)
    drawAxisGlobal()
    glRotatef(rz, 0, 0, 1)
    glPushMatrix()
       glTranslatef(0, 0.5, 0)
       glScalef(0.2, 1, 0.2)
       glutWireCube(1)
    glPopMatrix()
    drawAxis()
    glTranslatef(0, 1, 0)
    glRotatef(rz, 0, 0, 1)
    drawAxis()
    glPushMatrix()
       glTranslatef(0, 0.5, 0)
       glScalef(0.2, 1, 0.2)
       glutWireCube(1)
       glPopMatrix()
    glFlush()
End Sub

Public Sub drawAxis()
    glColor3f(1, 0, 0)
    glBegin(GL_LINES)
        glVertex3f(-1, 0, 0)
        glVertex3f(1, 0, 0)
    glEnd()
    glBegin(GL_LINES)
        glVertex3f(0, -1, 0)
        glVertex3f(0, 1, 0)
    glEnd()
    glBegin(GL_LINES)
        glVertex3f(0, 0, -1)
        glVertex3f(0, 0, 1)
    glEnd()
    glColor3f(1, 1, 1)
End Sub

Public Sub drawAxisGlobal()
    glBegin(GL_LINES)
       glVertex3f(-10, 0, 0)
       glVertex3f(10, 0, 0)
    glEnd()
    glBegin(GL_LINES)
       glVertex3f(0, -10, 0)
       glVertex3f(0, 10, 0)
    glEnd()
End Sub
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _
                               System.EventArgs) Handles MyBase.Load
        ' Timer settings
        Timer1.Enabled = False
        Timer1.Interval = 10

        glutInit()
        glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH)
        glutInitWindowSize(wWidth, wHeight)
        glutInitWindowPosition(100, 100)
        glutCreateWindow("gl05")
        glutDisplayFunc(New DisplayCallback(AddressOf Draw))
        glutReshapeFunc(New ReshapeCallback(AddressOf ViewSet))
        Me.Show()
        glutMainLoop()
    End Sub

    Private Sub Button1_MouseDown(ByVal sender As Object, ByVal e As _
                System.Windows.Forms.MouseEventArgs) Handles
Button1.MouseDown
        Timer1.Enabled = True
    End Sub

    Private Sub Button1_MouseUp(ByVal sender As Object, ByVal e As _
                System.Windows.Forms.MouseEventArgs) Handles
Button1.MouseUp
        Timer1.Enabled = False
    End Sub

    Private Sub Timer1_Tick(ByVal sender As Object, ByVal e As _
                               System.EventArgs) Handles Timer1.Tick
        rz = rz + 2
        Draw()
    End Sub

End Class
```

（演習問題）

１．robotarm.frm を基に，ロボットアームに対し，次の制御ができるようプログラムを改良しなさい

 1)逆方向にも動く（動きを戻す）ようにする.

 2)Y 軸まわりの回転ができるようにする.

２．４つの関節をもつロボットアームを作成しなさい

３．複数の指をもつロボットの手を作成しなさい.

上記 2，3について，動作がわかるように複数の実行画面と描画部分のコード（上部コードの Draw に該当）を印刷して提出すること（次回講義で回収）

[OpenGL:5]OpenGL によるアニメーション

１．ダブルバッファリングの設定

1)初期設定

GLUT 初期化時の glutInitDisplayMode に GLUT_DOUBLE を設定する.

' ダブルバッファ切替

' ダブルバッファを使わない場合

```
'glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH)
```

' ダブルバッファを使う場合

```
glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH Or GLUT_DOUBLE)
```

2) バッファの切り替え

これまで利用していた glFlush に代えて，以下のコマンドを用いる.

```
glutSwapBuffers()
```

２．シェーディング，ライティングの仮設定

1)シェーディングの設定

●Form1_Load 中に以下のコマンドを加える.

```
glDepthFunc(GL_LEQUAL )     '陰面処理における Depth の判定方法
glEnable(GL_DEPTH_TEST)     'DepthTest を ON にする.
glShadeModel(GL_FLAT)       'フラットシェーディングを行うことを宣言
```

●Draw 中の glClear を以下のように変更する.

```
glClear(GL_COLOR_BUFFER_BIT Or GL_DEPTH_BUFFER_BIT)
```

2)ライティングの設定

●Form1_Load プロシージャからの呼び出しを変更する.

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _
            System.EventArgs) Handles MyBase.Load
    Timer1.Interval = 10
    Timer1.Enabled = False
    glutInit()
    ' (中略)
    makemodel()
    SetLight()
    Me.Show()
    glutMainLoop()
End Sub
```

以下のサブプロシージャを作成する.

```
Private Sub SetLight()
   glLightfv(GL_LIGHT0, GL_AMBIENT, _
            New Single() {0.3!, 0.3!, 0.3!, 1.0!})
    glEnable(GL_LIGHTING)
    glEnable(GL_LIGHT0)
End Sub
```

３．ディスプレイリスト

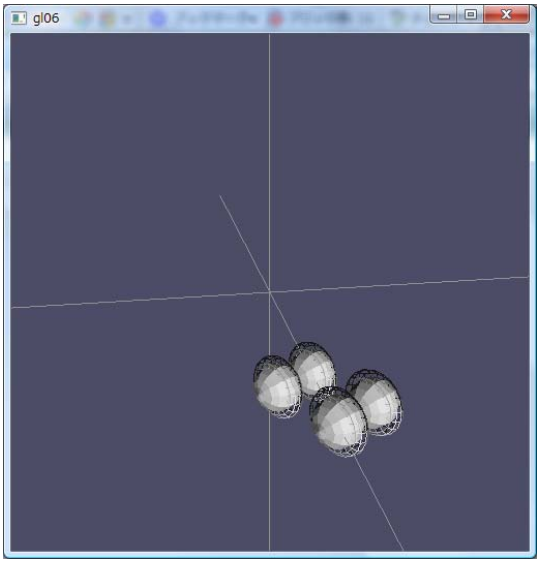
モデリングされた部品群のグループ化：

例えばタイヤとホイールのように常に一体化して動く部品をグループ化し，あらかじめ OpenGL 上でコンパイルしておくことにより，無駄な演算処理を省き，高速な描画を実現する.

```
glNewList(gluint, GL_COMPILE)
    ‘モデルを作成するコマンドを記述する
glEndList()
```

（※gluint は整数 ）

４．移動する車を作る



コード１　移動する Teapot

```
Imports Tao.OpenGl.Gl
Imports Tao.OpenGl.Glu
Imports Tao.FreeGlut.Glut


Public Class Form1
    Private Structure Point3D
        Dim X As Single
        Dim Y As Single
        Dim Z As Single
    End Structure

    Dim wWidth As Integer = 500
    Dim wHeight As Integer = 500
    Dim r As Point3D    '回転角
    Dim addr As Point3D  '回転角の増加分
    Dim pos As Point3D   '物体の位置
    Dim dist As Single   '移動距離

    Private Sub ViewSet(ByVal w As Integer, ByVal h As Integer)
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()
        gluPerspective(50, w / h, 1, 100)
        glViewport(0, 0, w, h)

        '視点の設定（今回は視点は移動しないので GL_PROJECTION で定義）
        glTranslatef(0.0#, 0.0#, -10.0#)
        glRotatef(20, 1, 0, 0)
        glRotatef(10, 0, 1, 0)

        'Matrix の切替
        glMatrixMode(GL_MODELVIEW)
    End Sub

    Public Sub Draw()
        glClearColor(0.3, 0.3, 0.4, 0)
        glClear(GL_COLOR_BUFFER_BIT Or GL_DEPTH_BUFFER_BIT)
        glColor3f(1.0#, 1.0#, 1.0#)

        glLoadIdentity()
        drawAxisGlobal()

        '物体の描画
        glTranslatef(pos.X, pos.Y, pos.Z)
        glRotatef(r.Y, 0, 1, 0)
        glutSolidTeapot(1)

        ' glFlush に代えて glutSwapBuffers を用いる
        glutSwapBuffers()
    End Sub
```

```
'座標軸を描く
Public Sub drawAxisGlobal()
    glBegin(GL_LINES)
        glVertex3f(-10, 0, 0)
        glVertex3f(10, 0, 0)
    glEnd()
    glBegin(GL_LINES)
        glVertex3f(0, 10, 0)
        glVertex3f(0, -10, 0)
    glEnd()
    glBegin(GL_LINES)
        glVertex3f(0, 0, -10)
        glVertex3f(0, 0, 10)
    glEnd()
End Sub

'ライトの設定
Private Sub setLight()
    glLightfv(GL_LIGHT0, GL_AMBIENT,
             New Single() {0.3!, 0.3!, 0.3!,1.0!})
    glEnable(GL_LIGHTING)
    glEnable(GL_LIGHT0)
End Sub

Private Sub Form1_Load1(ByVal sender As Object,
                ByVal e As System.EventArgs) Handles Me.Load
    Timer1.Interval = 10
    Timer1.Enabled = False
    glutInit()

    ' ダブルバッファリング
    glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH Or GLUT_DOUBLE)

    glutInitWindowSize(wWidth, wHeight)
    glutInitWindowPosition(300, 100)
    glutCreateWindow("gl06")
    glutDisplayFunc(New DisplayCallback(AddressOf Draw))
    glutReshapeFunc(New ReshapeCallback(AddressOf ViewSet))
    'シェーディングの設定
    glDepthFunc(GL_LEQUAL)    '陰面処理における Depth の判定方法
    glEnable(GL_DEPTH_TEST)    'DepthTest を ON にする.
    glShadeModel(GL_FLAT)      'フラットシェーディングを行うことを宣言
    setLight()
    Me.Show()
    glutMainLoop()
End Sub

Private Sub Button1_MouseDown(ByVal sender As Object,
    ByVal e As System.Windows.Forms.MouseEventArgs) _
    Handles Button1.MouseDown

    dist = If(e.Button = Windows.Forms.MouseButtons.Left, 0.01, -0.01)
    Timer1.Enabled = True
End Sub
```

```
    Private Sub Button1_MouseUp(ByVal sender As Object,
          ByVal e As System.Windows.Forms.MouseEventArgs) _
          Handles Button1.MouseUp

        Timer1.Enabled = False
        dist = 0
    End Sub

    Private Sub Button2_MouseDown(ByVal sender As Object,
        ByVal e As System.Windows.Forms.MouseEventArgs) _
        Handles Button2.MouseDown

        addr.Y = If(e.Button = Windows.Forms.MouseButtons.Left, 1, -1)
        Timer1.Enabled = True
    End Sub

    Private Sub Button2_MouseUp(ByVal sender As Object,
        ByVal e As System.Windows.Forms.MouseEventArgs) _
        Handles Button2.MouseUp

        Timer1.Enabled = False
        addr.Y = 0
    End Sub

    Private Sub Timer1_Tick(ByVal sender As Object,
                        ByVal e As System.EventArgs) _
                        Handles Timer1.Tick
        r.Y = r.Y + addr.Y
        pos.X = pos.X + dist * Math.Cos((r.Y) / 180 * Math.PI)
        pos.Y = 0
        pos.Z = pos.Z + dist * -Math.Sin((r.Y) / 180 * Math.PI)
        Draw()
    End Sub
End Class
```

コード２　移動するタイヤ（１個）

```
    Public Sub Draw()
        glClearColor(0.3, 0.3, 0.4, 0)
        glClear(GL_COLOR_BUFFER_BIT Or GL_DEPTH_BUFFER_BIT)
        glColor3f(1.0#, 1.0#, 1.0#)

        glLoadIdentity()
        drawAxisGlobal()


        '物体の描画
        glTranslatef(pos.X, pos.Y, pos.Z)
        glRotatef(r.Y, 0, 1, 0)
```

```
        glPushMatrix()
            glRotatef(-r.Z, 0, 0, 1)
            glutSolidSphere(0.3, 10, 10)
            glutWireTorus(0.1, 0.3, 10, 20)
        glPopMatrix()

        ' glFlush に代えて glutSwapBuffers を用いる
        glutSwapBuffers()
    End Sub

    Private Sub Timer1_Tick(ByVal sender As Object, ByVal e As  _
              System.EventArgs) Handles Timer1.Tick

        r.Y = (r.Y + addr.Y) Mod 360
        r.Z = (dist * 100 + r.Z) Mod 360
        pos.X = pos.X + dist * Math.Cos((r.Y) / 180 * Math.PI)
        pos.Y = 0
        pos.Z = pos.Z + dist * -Math.Sin((r.Y) / 180 * Math.PI)
        Draw()
    End Sub
```

コード３　移動する車（タイヤ４個）

```
    Private Sub Form1_Load(ByVal sender As System.Object, _
                 ByVal e As System.EventArgs) Handles MyBase.Load
        Timer1.Interval = 10
        Timer1.Enabled = False

        glutInit()

        ' ダブルバッファリング
        glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH Or GLUT_DOUBLE)

        glutInitWindowSize(wWidth, wHeight)
        glutInitWindowPosition(300, 100)
        glutCreateWindow("gl06")
        glutDisplayFunc(New DisplayCallback(AddressOf Draw))
        glutReshapeFunc(New ReshapeCallback(AddressOf ViewSet))


        'シェーディングの設定
        glDepthFunc(GL_LEQUAL)     '隠面処理における Depth の判定方法
        glEnable(GL_DEPTH_TEST)     'DepthTest を ON にする.
        glShadeModel(GL_FLAT)      'フラットシェーディングを行うことを宣言
        setLight()
        makeModel()          '←この行を追加
        Me.Show()
        glutMainLoop()
    End Sub


    Private Sub makeModel()
        glNewList(1, GL_COMPILE)
            glutSolidSphere(0.3, 10, 10)
            glutWireTorus(0.1, 0.3, 10, 20)
```

```
    glEndList()
End Sub

Public Sub Draw()
    glClearColor(0.3, 0.3, 0.4, 0)
    glClear(GL_COLOR_BUFFER_BIT Or GL_DEPTH_BUFFER_BIT)
    glColor3f(1.0#, 1.0#, 1.0#)

    glLoadIdentity()
    drawAxisGlobal()

    '物体の描画
    glTranslatef(pos.X, pos.Y, pos.Z)
    glRotatef(r.Y, 0, 1, 0)

    glPushMatrix()
        glPushMatrix()
            glTranslatef(-0.5, 0, -0.3)
            glRotatef(-r.Z, 0, 0, 1)
            glCallList(1)
        glPopMatrix()
        glPushMatrix()
            glTranslatef(0.5, 0, -0.3)
            glRotatef(-r.Z, 0, 0, 1)
            glCallList(1)
        glPopMatrix()
        glPushMatrix()
            glTranslatef(-0.5, 0, 0.3)
            glRotatef(-r.Z, 0, 0, 1)
            glCallList(1)
        glPopMatrix()
        glPushMatrix()
            glTranslatef(0.5, 0, 0.3)
            glRotatef(-r.Z, 0, 0, 1)
            glCallList(1)
        glPopMatrix()
    glPopMatrix()
    ' glFlush に代えて glutSwapBuffers を用いる
    glutSwapBuffers()
End Sub
```

（演習問題）

コード１，２と前回の課題とを組み合わせ，台車に乗って移動するロボットアームを作成しなさい．描画部分(draw)のコードと実行画面を印刷して提出すること（次回講義で回収）．

[OpenGL:6]シェーディングの設定

１．シェーディングモデル

(1)フラットシェーディング(Flat shading)：
　面の頂点の色を同一として描画する(Lambert shading)
(2)スムースシェーディング：滑らかな曲面の描画を行う．
　グ-ローシェーディング（Gouraud shading）：
　　頂点の法線ベクトルから面の頂点の色を求め，それを補間して面の色を描画する．
　フォンシェーディング（Phong shading)：
　　頂点の法線ベクトルを基に，面全体の法線ベクトルを求め，それにより面の色を決定する．

OpenGL によるシェーディングの設定
　フラットシェーディングの設定
　　glShadeModel(GL_FLAT)
　スムースシェーディング(Gouraud Shading) の設定
　　glShadeModel(GL_SMOOTH)

２．面の法線と塗りつぶし

　面の向きは法線の向きにより決定される．また法線の向きは面の色に大きく影響する．（点毎の法線も設定できる）

法線の設定
```
   glNormal3f (x, y, z)
```
ポリゴンモードの設定
```
   glPolygonMode(GL_FRONT_AND_BACK, GL_FILL)  '両面を塗りつぶす
   glPolygonMode(GL_FRONT, GL_FILL)           '表を塗りつぶす
   glPolygonMode(GL_BACK, GL_LINE)            '裏を線で描画
   glPolygonMode(GL_BACK, GL_POINTS)          '裏を点で描画
```

３．物体あるいは面の色の設定方法

以下のように glMaterialfv 関数に配列で受け渡す必要がある．

例）

```
Dim MaterialDiffuse(4) As Single
Dim MaterialSpecular(4) As Single
MaterialDiffuse(0)=.4  '赤(R)
MaterialDiffuse(1)=.3  '緑(G)
MaterialDiffuse(2)=.3  '青(B)
MaterialDiffuse(3)=1   'アルファ(A)
MaterialSpecular(0)=.2  '赤(R)
MaterialSpecular(1)=.2  '緑(G)
MaterialSpecular(2)=.2  '青(B)
MaterialSpecular(3)=1   'アルファ(A)

'配列の０番目を受け渡す（環境光及び拡散光）
glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE,, MaterialDiffuse)
'配列の０番目を受け渡す（反射光）
glMaterialfv(GL_FRONT, GL_SPECULAR, MaterialSpecular)
```

４．コード

```
Imports Tao.OpenGl.Gl
Imports Tao.OpenGl.Glu
Imports Tao.FreeGlut.Glut

Public Class Form1
    Private Structure Point3D
        Dim X As Single
        Dim Y As Single
        Dim Z As Single
    End Structure

    Dim wWidth As Integer = 500
    Dim wHeight As Integer = 500
    Dim r As Point3D
    Dim addr As Point3D
    Dim pnt(100) As Point3D  '新たに定義（ポイント座標）

    Private Sub ViewSet(ByVal w As Integer, ByVal h As Integer)
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()
        gluPerspective(50, w / h, 1, 100)
        glViewport(0, 0, w, h)
        glMatrixMode(GL_MODELVIEW)
    End Sub

    Public Sub Draw()
        Dim n As Integer
        Dim normal As Point3D '法線格納用変数

        '背景色の設定（RGBA)
        glClearColor(0.2, 0.2, 0.3, 0)
        glClear(GL_COLOR_BUFFER_BIT Or GL_DEPTH_BUFFER_BIT)
```

```
glColor3f(1.0#, 1.0#, 1.0#)
glLoadIdentity()

'ここにライトのポジション設定を与えると
'物体の移動・回転とは関係ない.
      glLightfv(GL_LIGHT0, GL_POSITION, New Single() {0.0!,
                        1.0!, _ 1.0!, 1.0!})


glRotatef(10, 1, 0, 0)    '視点回転
glTranslatef(0.0#, -1.0#, -5.0#)  '視点設定
glPushMatrix()
   glRotatef(r.Y, 0, 1, 0)
   glRotatef(r.Z, 0, 0, 1)

   'ここにライトのポジション設定を与えると物体の回転とともに
   'ライトも回転する.
   '    FillArray4f positionLight0(), 0!, 1!, 1!, 1!
   '    glLightfv GL_LIGHT0, GL_POSITION, positionLight0(0)

   '物体の色の設定１
   glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, _
                     New Single() {0.45, 0.3, 0.15, 1.0})
   glMaterialfv(GL_FRONT, GL_SPECULAR, New Single() _
                     {0.2, 0.1, 0.1, 1.0})

   glutSolidSphere(0.3, 20, 20)
   glPushMatrix()
      glTranslatef(1, 0, 0)

      '物体の色の設定２
      glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, _
                        New Single() {0.2, 0.2, 0.2, 1.0})
      glMaterialfv(GL_FRONT, GL_SPECULAR, New Single() _
                           {0.3, 0.4, 0.4, 1.0})
      glutSolidSphere(0.5, 20, 20)
   glPopMatrix()

   '任意の物体の描画
   glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, _
                        New Single() {0.5, 0.1, 0.1, 1.0!})
   glMaterialfv(GL_FRONT, GL_SPECULAR, _
                        New Single() {0.2, 0.1, 0.1, 1.0})
   glPolygonMode(GL_FRONT, GL_FILL)
   glPolygonMode(GL_BACK, GL_LINE)
   glBegin(GL_POLYGON)
      glNormal3f(0, 0, 1)  '法線の指定
      glVertex3f(-1, -1, 2)
      glVertex3f(1, -1, 2)
      glVertex3f(0, 1, 2)
   glEnd()
```

```
        '配列を用いた面の描画
        pnt(0).X = -2 : pnt(0).Y = -1 : pnt(0).Z = -4
        pnt(1).X = -1 : pnt(1).Y = -1 : pnt(1).Z = -4
        pnt(2).X = 2 : pnt(2).Y = 0 : pnt(2).Z = -4
        pnt(3).X = 1 : pnt(3).Y = 1 : pnt(3).Z = -4
        pnt(4).X = -2 : pnt(4).Y = 0 : pnt(4).Z = -4

        '法線の計算，3 点のみ与えればよい.
        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, _
                             New Single() {0.1, 0.1, 0.4, 1.0})
        glMaterialfv(GL_FRONT, GL_SPECULAR, New Single() _
                             {0.1, 0.1, 0.4, 1.0})
        normal = calcNormal(pnt(0), pnt(1), pnt(2))

        glPolygonMode(GL_FRONT_AND_BACK, GL_FILL)

        glBegin(GL_POLYGON)
          glNormal3f(normal.X, normal.Y, normal.Z)
          For n = 0 To 4
            glVertex3f(pnt(n).X, pnt(n).Y, pnt(n).Z)
          Next
        glEnd()

    glPopMatrix()
    glutSwapBuffers()
End Sub


'ポリゴンの法線を計算するプロシージャ
Private Function calcNormal(ByVal P1 As Point3D, _
              ByVal P2 As Point3D, ByVal P3 As Point3D) As Point3D
   Dim n As Point3D
   Dim length As Single

   n.X = (P2.Y - P1.Y) * (P3.Z - P2.Z) - (P2.Z - P1.Z) * (P3.Y - P2.Y)
   n.Y = (P2.Z - P1.Z) * (P3.X - P2.X) - (P2.X - P1.X) * (P3.Z - P2.Z)
   n.Z = (P2.X - P1.X) * (P3.Y - P2.Y) - (P2.Y - P1.Y) * (P3.X - P2.X)

    '長さの計算
   length = Math.Sqrt(n.X * n.X + n.Y * n.Y + n.Z * n.Z)
    '単位ベクトルにする
   If length <> 0 Then
       calcNormal.X = n.X / length
       calcNormal.Y = n.Y / length
       calcNormal.Z = n.Z / length
   End If
End Function

Private Sub SetLight()
  glLightfv(GL_LIGHT0,GL_AMBIENT,New Single(){0.3, 0.3, 0.3, 1.0})
  glEnable(GL_LIGHTING)
  glEnable(GL_LIGHT0)
End Sub
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As  _
                       System.EventArgs) Handles MyBase.Load
    Timer1.Interval = 10
    Timer1.Enabled = False

    glutInit()
    glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH Or GLUT_DOUBLE)
    glutInitWindowSize(wWidth, wHeight)
    glutInitWindowPosition(100, 100)
    glutCreateWindow("gl07")
    glutDisplayFunc(New DisplayCallback(AddressOf Draw))
    glutReshapeFunc(New ReshapeCallback(AddressOf ViewSet))

    glDepthFunc(GL_LEQUAL)
    glEnable(GL_DEPTH_TEST)

    'シェーディング設定
    'glShadeModel(GL_SMOOTH) 'スムースシェーディングの場合
    glShadeModel(GL_FLAT)   'フラットシェーディングの場合

    SetLight()
    Me.Show()
    glutMainLoop()
 End Sub

Private Sub Button1_MouseDown(ByVal sender As Object, ByVal e As  _
   System.Windows.Forms.MouseEventArgs) Handles Button1.MouseDown


  Dim add As Integer=IIf(e.Button =  _
            Windows.Forms.MouseButtons.Left, 1, -1)

  addr.Y = 0
  addr.Z = add * 10

  Timer1.Enabled = True
End Sub

Private Sub Button1_MouseUp(ByVal sender As Object, ByVal e As  _
   System.Windows.Forms.MouseEventArgs) Handles Button1.MouseUp
    Timer1.Enabled = False
End Sub

Private Sub Button2_MouseDown(ByVal sender As Object, ByVal e As  _
   System.Windows.Forms.MouseEventArgs) Handles Button2.MouseDown
    Dim add As Integer = IIf(e.Button = _
                         Windows.Forms.MouseButtons.Left, 1, -1)

    addr.Y = add * 10
    addr.Z = 0
    Timer1.Enabled = True
 End Sub
```

```
        Private Sub Button2_MouseUp(ByVal sender As Object, ByVal e As  _
            System.Windows.Forms.MouseEventArgs) Handles Button2.MouseUp
            Timer1.Enabled = False
        End Sub

        Private Sub Timer1_Tick(ByVal sender As Object, ByVal _
                            e As System.EventArgs) Handles Timer1.Tick
            r.Z = r.Z + addr.Z
            r.Y = r.Y + addr.Y
            Draw()
        End Sub
    End Class
```

任意の座標をもつ多角形を書く場合は,

```
  glBegin(GL_POLYGON)
     glNormal3f(0, 0, 1)    '法線を指定
     glVertex3f(0, 0, 0)     座標を指定
     glVertex3f(2, 3, 0)
     glVertex3f(3, 3, 0)
      ・・・・・・（略）
  glEnd
```

その他の glut の図形描画コマンド

glutSolidCube, glutSolidDodecahedron, glutSolidSphere
glutSolidTeapot, glutSolidTorus, glutSolidCone など

[OpenGL:7]照光処理（ライティング）

１．照光モデルにおける光の分類

放射光　　emission
環境光　　ambient
拡散光　　diffuse
鏡面光（鏡面反射）　specular
→物体の色は光の特性及び物体の特性により決定される.

２．光源の種類

平行光源
点光源
スポットライト
詳細は RedBook を参照 http://www.glprogramming.com/red/

３．ライトの設定

glLightfv(light, pname, param)
light は GL_LIGHT0, GL_LIGHT1, ……, GL_LIGHT7 まで
pname は設定項目　GL_AMBIENT 等
param は設定した配列等

例)

```
glLightfv(GL_LIGHT1, GL_AMBIENT, ambientLight1(0))   '環境光設定
glLightfv(GL_LIGHT1, GL_DIFFUSE, diffuseLight1(0))   '拡散光設定
glLightfv(GL_LIGHT1, GL_SPECULAR, specularLight1(0)) '鏡面反射光設定
glLightfv(GL_LIGHT1, GL_POSITION, positionLight1(0)) 'ライトの位置設定
```

光源の種類の設定
GL_POSITION で設定する照明位置の同次座標(x,y,z,w)の w の値により異なる.
w=1：点光源
w=0：平行光源

例)

```
positionLight1(0)=1
positionLight1(1)=1
positionLight1(2)=0
positionLight1(3)=0   'ここの値が 0 であれば平行光源，1 であれば点光源.
 glLightfv(GL_LIGHT1, GL_POSITION, positionLight1(0))
```

点光源の場合には照明位置は実座標として指定できる．また光は減衰する．

平行光源の場合には距離は関係無く，無限遠にある光源（太陽）と考えるため光は減衰しない．

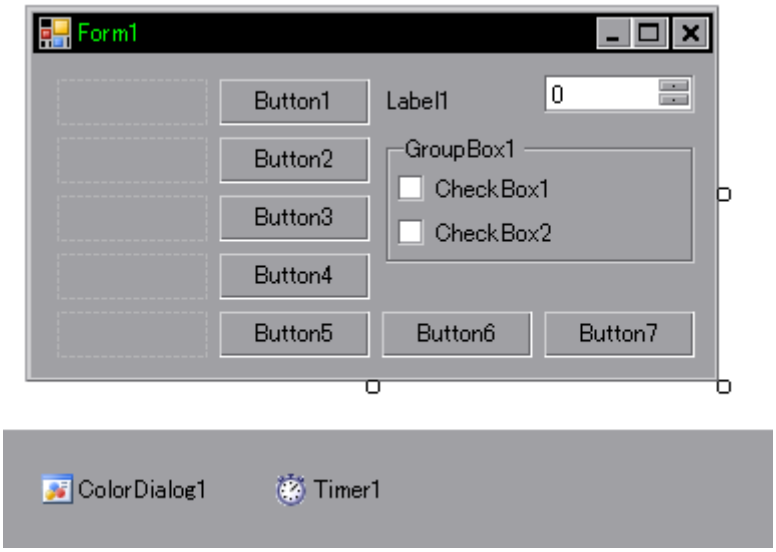
４．物体の色の指定

それぞれの物体に対して環境光，拡散光，鏡面光（場合によっては放射光）を設定する．

例)

```
glMaterialfv GL_FRONT, GL_AMBIENT, MaterialAmbient
glMaterialfv GL_FRONT, GL_DIFFUSE, MaterialDiffuse
glMaterialfv GL_FRONT, GL_SPECULAR, MaterialSpecular
```

５．コード



```
Option Explicit On
Imports Tao.OpenGl
Imports Tao.OpenGl.Gl
Imports Tao.OpenGl.Glu
Imports Tao.FreeGlut.Glut

Public Class Form1
    Dim m_hGLRC As Long

    Private Structure point3D
        Dim X As Single
        Dim Y As Single
        Dim Z As Single
    End Structure

    Dim Rot As point3D
    Dim Tr As point3D
    Dim AddR As point3D
    Dim AddT As point3D
    Dim LightPos As point3D

    Dim wWidth As Integer = 500
    Dim wHeight As Integer = 500
    Dim Initialized As Boolean = False

    Private Sub Initialize()
        glutInit()
        glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH Or GLUT_DOUBLE)
        glutInitWindowSize(300, 300)
        glutInitWindowPosition(400, 100)
        glutCreateWindow("Light")
        glutDisplayFunc(New DisplayCallback(AddressOf draw))
        glutReshapeFunc(New ReshapeCallback(AddressOf SetView))
        glDepthFunc(GL_LEQUAL)
        glEnable(GL_DEPTH_TEST)
        glShadeModel(GL_FLAT)
        glEnable(GL_BLEND)
        glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)
        Me.Show()
        Initialized = True
        SetLight()
        glutMainLoop()
    End Sub

    '初期設定
    Private Sub Form1_Load(ByVal sender As System.Object, _
                    ByVal e As System.EventArgs) Handles MyBase.Load
        Button1.Text = "Ambient"
        Button2.Text = "Diffuse"
        Button3.Text = "Specular"
        Button4.Text = "Emit"
        Button5.Text = "Back"
        Button6.Text = "Flat/Smooth"
```

```
    Button7.Text = "Move/Stop"
    CheckBox1.Text = "Parallel"
    CheckBox2.Text = "Point"
    GroupBox1.Text = "Light"
    Label1.Text = "Alpha"

    PictureBox1.BackColor = Color.FromArgb(&HFF002110)
    PictureBox2.BackColor = Color.FromArgb(&HFF00BB10)
    PictureBox3.BackColor = Color.FromArgb(&HFFBBBB20)
    PictureBox4.BackColor = Color.Black
    PictureBox5.BackColor = Color.Black

    ColorDialog1.FullOpen = True

    NumericUpDown1.TextAlign = HorizontalAlignment.Right
    NumericUpDown1.Maximum = 255
    NumericUpDown1.Minimum = 0
    NumericUpDown1.Value = 255

    LightPos.X = 2 : LightPos.Y = 2 : LightPos.Z = 0


    Timer1.Interval = 1
    Initialize()
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button1.Click
    ColorDialog1.Color = PictureBox1.BackColor
    ColorDialog1.ShowDialog()
    PictureBox1.BackColor = ColorDialog1.Color
    draw()
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button2.Click
    ColorDialog1.Color = PictureBox2.BackColor
    ColorDialog1.ShowDialog()
    PictureBox2.BackColor = ColorDialog1.Color
    draw()
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button3.Click
    ColorDialog1.Color = PictureBox3.BackColor
    ColorDialog1.ShowDialog()
    PictureBox3.BackColor = ColorDialog1.Color
    draw()
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button4.Click
    ColorDialog1.Color = PictureBox4.BackColor
    ColorDialog1.ShowDialog()
```

```
    PictureBox4.BackColor = ColorDialog1.Color
    draw()
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button5.Click
    ColorDialog1.Color = PictureBox5.BackColor
    ColorDialog1.ShowDialog()
    PictureBox5.BackColor = ColorDialog1.Color
    draw()
End Sub

Private Sub CheckBox1_CheckedChanged(ByVal sender As  _
    System.Object, ByVal e As System.EventArgs) Handles _
    CheckBox1.CheckedChanged
    If CheckBox1.Checked Then
        glEnable(GL_LIGHT0)
    Else
        glDisable(GL_LIGHT0)
    End If
    '再描画
    draw()
End Sub

Private Sub CheckBox2_CheckedChanged(ByVal sender As  _
    System.Object, ByVal e As System.EventArgs) _
    Handles CheckBox2.CheckedChanged

    If CheckBox2.Checked Then
        glEnable(GL_LIGHT1)
    Else
        glDisable(GL_LIGHT1)
    End If

    '再描画
    draw()
End Sub

' シェーディング切り替え
Private Sub Button6_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button6.Click
    Static smoothFlag As Boolean
    smoothFlag = Not smoothFlag

    If smoothFlag Then
        glShadeModel(GL_SMOOTH)
    Else
        glShadeModel(GL_FLAT)
    End If
    draw()
End Sub

'タイマー起動
Private Sub Button7_Click(ByVal sender As System.Object, _
```

```
       ByVal e As System.EventArgs) Handles Button7.Click
    AddR.Y = 1
    Timer1.Enabled = Not Timer1.Enabled
End Sub

Private Sub Timer1_Tick(ByVal sender As Object, _
       ByVal e As System.EventArgs) Handles Timer1.Tick
    Rot.Y = Rot.Y + AddR.Y
    draw()
End Sub

Private Sub NumericUpDown1_ValueChanged(ByVal sender As Object, _
ByVal e As System.EventArgs) Handles NumericUpDown1.ValueChanged
    If Me.Initialized Then
       draw()
    End If
End Sub

Private Sub SetView(ByVal nWidth As Integer, ByVal nHeight As Integer)
    glViewport(0, 0, nWidth, nHeight)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluPerspective(50, nWidth / nHeight, 0.2, 100)
    glMatrixMode(GL_MODELVIEW)
End Sub

Private Sub SetLight()
    'ライトの設定(平行光源)
    glLightfv(GL_LIGHT0, GL_AMBIENT, _
              New Single() {0.5!, 0.5!, 0.5!, 1.0!}) '環境光設定
    glLightfv(GL_LIGHT0, GL_DIFFUSE, _
              New Single() {0.5!, 0.5!, 0.5!, 1.0!}) '拡散光設定
    glLightfv(GL_LIGHT0, GL_SPECULAR, New Single() _
           {0.5!, 0.5!, 0.5!, 1.0!})   '鏡面反射光設定
    glLightfv(GL_LIGHT0, GL_POSITION, New Single() _
           {0.0!, 1.0!, 1.0!, 0.0!})   'ライトの位置設定

    'ライティングを使用可能に
    glEnable(GL_LIGHTING)
    glEnable(GL_LIGHT0)
    glDisable(GL_LIGHT1)
    CheckBox1.Checked = True
    CheckBox2.Checked = False

    'ライトの設定（点光源）
    glLightfv(GL_LIGHT1, GL_AMBIENT, _
              New Single() {0.2!, 0.2!, 0.2!, 1.0!})    '環境光設定
    glLightfv(GL_LIGHT1, GL_DIFFUSE, _
              New Single() {0.6!, 0.6!, 0.6!, 1.0!})    '拡散光設定
    glLightfv(GL_LIGHT1, GL_SPECULAR, _
              New Single() {0.2!, 0.2!, 0.2!, 1.0!})   '鏡面反射光設定

End Sub
```

```
Public Sub draw()
   glMatrixMode(GL_MODELVIEW)

   '背景色の設定
   Dim bgColor As Color = PictureBox5.BackColor
   glClearColor(bgColor.R / 255, bgColor.G / 255, bgColor.B / 255, 1)

   glClear(GL_COLOR_BUFFER_BIT Or GL_DEPTH_BUFFER_BIT)
   glLoadIdentity()

   '視点の設定
   glTranslatef(0, 0, -5)
   glRotatef(10, 1, 0, 0)
   glRotatef(Rot.Y, 0, 1, 0)

   glLightfv(GL_LIGHT1, GL_POSITION, New Single() _
        {LightPos.X, LightPos.Y, LightPos.Z, 1.0!})   'ライトの位置設定


   '図形の描画
   glPushMatrix()
      glTranslatef(0.5, 1, -5)

      'SetColor サブプロシージャを用いた色設定
      SetColor(New Single() {0.3, 0.05, 0.05, 1}, _
             New Single() {0.6, 0.5, 0.05, 1}, _
             New Single() {0.35, 0.1, 0.1, 1})
      glutSolidCube(3.0)
   glPopMatrix()

   glMaterialfv(GL_FRONT, GL_AMBIENT, GetColor(PictureBox1))
   glMaterialfv(GL_FRONT, GL_DIFFUSE, GetColor(PictureBox2))
   glMaterialfv(GL_FRONT, GL_SPECULAR, GetColor(PictureBox3))
   glMaterialfv(GL_FRONT, GL_EMISSION, GetColor(PictureBox4))

   glPushMatrix()
      glRotatef(-90, 1, 0, 0)
      glutSolidCone(0.7, 2, 10, 10)
   glPopMatrix()
   glPushMatrix()
      glTranslatef(-1, 0, -1)
      glutSolidSphere(0.7, 10, 10)
   glPopMatrix()
   glMaterialfv(GL_FRONT, GL_EMISSION, New Single() {0, 0, 0, 0})
   glutSwapBuffers()
End Sub

Sub SetColor(ByVal amb() As Single, ByVal dff() As Single, _
                                 ByVal spc() As Single)
   glMaterialfv(GL_FRONT, GL_AMBIENT, amb)
   glMaterialfv(GL_FRONT, GL_DIFFUSE, dff)
   glMaterialfv(GL_FRONT, GL_SPECULAR, spc)
End Sub
```

```
    Function GetColor(ByVal pBox As PictureBox) As Single()
        Dim c As Color = pBox.BackColor
        GetColor = New Single() {c.R / 255, c.G / 255, c.B / 255, _
                                 NumericUpDown1.Value / 255}
    End Function
End Class
```

【課題】
これまでに学んだことをもとに，OpenGL を用いたプログラムを作成しなさい． 最低限，以下の機能を有するものであること．
1．ユーザーが動きを制御・「ｐできるものであること．
2．ロボットアームを応用した機能が含まれていること．
3．照光および着色処理がなされていること．
※ゲーム性や物理シミュレーションを加えるなど各自，工夫を加えること．
1 月 10 日(木)12:50 までに E-mail 添付ファイルにてフォルダごと圧縮してメールで送信すること．
・メールの件名は 学籍番号（半角）氏名
・フォルダ名は 学籍番号.zip
・送信先メールアドレス：makarepo@myu.ac.jp

（今後の予定）

1 月 10 日（木）混合処理（ブレンディング）
1 月 17 日（木）テクスチュアマッピング(1)
1 月 24 日（木）テクスチュアマッピング(2)
1 月 31 日（木）まとめ・最終成果チェック
（※最終提出期限：2 月 5 日（木））

[OpenGL:8] 混合処理（ブレンディング）

１．混合により可能な処理

1)トランスペアレンシィ（transparency；透明度)

アルファ値を基に，物体に対して透明度を設定する（ガラスなどの表現)．

またトランスペアレンシィとマスクを用いたテクスチュアマッピングを組み合わせるアルファマッピングは，特に樹木等の膨大なポリゴンをもつ物体の描画処理に有効である．

2)アンチエイリアシング(anti-aliasing)

低解像度で画像計算した場合に発生するエイリアスを除去する技術.

注)alias とは

低解像度で画像を計算した場合に発生する現象であり，物体のエッジに生ずるジャギー（ギザギザ)， 低速度で運動する物体輪郭の震動，ピクセル以下の大きさの物体が運動する際の点滅，高密度の連続パターンに生じるモアレなど複数の意味がある。

3)フォグ(fog; 霧)

その名の通り，霧を表現する手法である．これにより，遠方のモデリングが不要となりデータ量を少なくすることができるとともに，より高い現実感を得ることができる．

２．トランスペアレンシィの設定

1)混合の関数を定める

```
glBlendFunc(sfactor, tfactor)
```

例)　 glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)

2)混合処理を有効にする

```
glEnable(GL_BLEND)
```

注)トランスペアレンシィを設定する物体に対しては，奥行判定（GL_DEPTH_TEST)が問題となる場合がある．そのため，透明度を設定する物体を最後に描画するようにすることが有効である．

３．Fog の設定

1)Fog に関するパラメータ（関数，フォグの色，濃さ，距離等）を設定する

```
glFogfv(pname, param)
```

例)

'fogclr()はユーザーの定義した配列

```
glFogfv(GL_FOG_COLOR, fogclr)
 glfogfv(GL_FOG_DENSITY, 0.3)
```

2)Fog を有効にする

```
glEnable(GL_FOG)
```

【参考】アンチエイリアシングの設定

1)混合の関数を定める

```
glBlendFunc(sfactor, tfactor)
```

例) glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)

2)混合処理を有効にする

```
glEnalbe(GL_BLEND)
```

3)アンチエイリアシングの品質を決定する

```
glHint(target, hint)
```

例) glHint(GL_POLYGON_SMOOTH_HINT,GL_FASTEST)

4)アンチエイリアシングを有効にする.

例)glEnable(GL_POLYGON_SMOOTH) 'ポリゴンの場合

```
glEnable(GL_LINE_SMOOTH)        '線の場合
```

注）アンチエイリアシングを用いる場合には奥行き判定（GL_DEPTH_TEST)は使えない.

（glDisable(GL_DEPTH_TEST) としておく必要がある.）

そのため，裏を向いているポリゴンを描画しないように，以下の設定をしておく.

```
glCullFace(GL_BACK)      '裏を向いているポリゴンの情報は廃棄される.
glEnable(GL_CULL_FACE)
```

またアンチエイリアシングにより，描画速度はかなり遅くなる.

（高速アニメーションには不適）

４．コード２（トランスペアレンシィ・フォグ）

```
Imports Tao.OpenGl
Imports Tao.OpenGl.Gl
Imports Tao.OpenGl.Glu
Imports Tao.FreeGlut.Glut

Public Class Form1
    Dim m_hGLRC As Long

    Private Structure axis
        Dim X As Single
        Dim Y As Single
        Dim Z As Single
    End Structure

    Private Structure Texture
        Dim Tex() As Byte
        Dim W As Integer
        Dim H As Integer
```

```
End Structure

Dim Rot As axis
Dim Tr As axis
Dim AddR As axis
Dim AddT As axis
Dim wWidth As Integer = 500
Dim wHeight As Integer = 500

'初期設定
Private Sub Form_Load() Handles MyBase.Load
    Timer1.Interval = 100
    Timer1.Enabled = False
    Initialize()
End Sub

'タイマー起動
Private Sub Button1_Click() Handles Button1.Click
    AddR.Y = 5
    Timer1.Enabled = Not Timer1.Enabled
End Sub

Private Sub Timer1_Tick() Handles Timer1.Tick
    Rot.Y = Rot.Y + AddR.Y
    display()
End Sub

Private Sub SetView()
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    Glu.gluPerspective(50, wWidth / wHeight, 0.2, 10000)
    glMatrixMode(GL_MODELVIEW)
    glViewport(0, 0, wWidth, wHeight)
    glTranslatef(0, -1, -3)
End Sub

Private Sub Initialize()
    ' GLUT Window
    glutInit()
    glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH Or GLUT_DOUBLE)
    glutInitWindowSize(wWidth, wHeight)
    glutInitWindowPosition(100, 100)
    glutCreateWindow("Antialias")
    glutDisplayFunc(New DisplayCallback(AddressOf display))
    glutReshapeFunc(New ReshapeCallback(AddressOf SetView))

    glDepthFunc(GL_LEQUAL)
    glEnable(GL_DEPTH_TEST)

    glShadeModel(GL_SMOOTH)

    '混合の関数を指定する.
    glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)
```

```
        '混合を有効にする.
        glEnable(GL_BLEND)

        Dim fogclr(0 To 3) As Single
        fogclr(0) = 0.8
        fogclr(1) = 0.8
        fogclr(2) = 0.7
        fogclr(3) = 1

        'Fog を有効にする
        glFogfv(GL_FOG_COLOR, fogclr)
        glFogfv(GL_FOG_DENSITY, 0.2!)
'       glEnable(GL_FOG)

        Me.Show() ' Show Form

        SetView()
        SetLight()

        glutMainLoop()
    End Sub

    Private Sub SetLight()
        glLightfv(GL_LIGHT0, GL_AMBIENT,
                New Single() {0.2!, 0.2!, 0.2!, 1.0!})  '環境光設定
        glLightfv(GL_LIGHT0, GL_DIFFUSE,
                New Single() {1.0!, 1.0!, 1.0!, 1.0!})  '拡散光設定
        glLightfv(GL_LIGHT0, GL_SPECULAR,
                New Single() {1.0!, 1.0!, 1.0!, 1.0!}) '鏡面反射光設定
        glLightfv(GL_LIGHT0, GL_POSITION,
                New Single() {0.0!, 1.0!, 1.0!, 0.0!}) 'ライトの位置配列
                                    '（平行光源の場合，配列の４番目は 0)

        'ライティングを使用可能に
        glEnable(GL_LIGHTING)
        glEnable(GL_LIGHT0)
        'glEnable GL_LIGHT1
    End Sub

    Public Sub display()
        glMatrixMode(GL_MODELVIEW)
        '背景色の設定
        glClearColor(0.8, 0.8, 0.9, 1)

        glClear(GL_COLOR_BUFFER_BIT Or GL_DEPTH_BUFFER_BIT)

        glLoadIdentity()

        '視点の設定
        glTranslatef(0, -1, -5)
        glRotatef(10, 1, 0, 0)
        glRotatef(rot.Y, 0, 1, 0)
```

```
        '図形の描画
        glPushMatrix()
        glTranslatef(0.5, 1, -5)
        SetColor(0.01, 0.01, 0.03, 0.1, 0.1, 0.3, 0.1, 0.1, 0.25, 1)
        glutSolidCube(3)
        glPopMatrix()

        glPushMatrix()
        glRotatef(-90, 1, 0, 0)
        SetColor(0.01, 0.01, 0.01, 0.2, 0.2, 0.22, 0.2, 0.2, 0.22, 1)
        glutSolidCone(0.7, 2, 10, 10)
        glPopMatrix()

        glPushMatrix()
        glTranslatef(-1, 0, -1)
        SetColor(0.3, 0.05, 0.05, 0.3, 0.05, 0.05, 0.35, 0.1, 0.1, 0.8)
        glutSolidSphere(0.7, 10, 10)
        glPopMatrix()

        glutSwapBuffers()
    End Sub

    Private Sub SetColor(ByVal ambR!, ByVal ambG!, ByVal ambB!, _
              ByVal difR!, ByVal difG!, ByVal difB!, ByVal specR!, _
                        ByVal specG!, ByVal specB!, ByVal alpha!)
        Dim MaterialAmbient(3) As Single
        Dim MaterialDiffuse(3) As Single
        Dim MaterialSpecular(3) As Single

        MaterialAmbient = New Single() {ambR, ambG, ambB, alpha}
        MaterialDiffuse = New Single() {difR, difG, difB, alpha}
        MaterialSpecular = New Single() {specR, specG, specB, alpha}
        glMaterialfv(GL_FRONT, GL_AMBIENT, MaterialAmbient(0))
        glMaterialfv(GL_FRONT, GL_DIFFUSE, MaterialDiffuse(0))
        glMaterialfv(GL_FRONT, GL_SPECULAR, MaterialSpecular(0))
    End Sub
End Class
```

演習

これまでに作成した自らのプログラムに混合処理（トランスペアレンシィおよびフォグ）の効果を加えたプログラムを作成し，その出力画像を印刷して提出しなさい．（次回講義で回収）

[OpenGL:9] テクスチュアマッピング

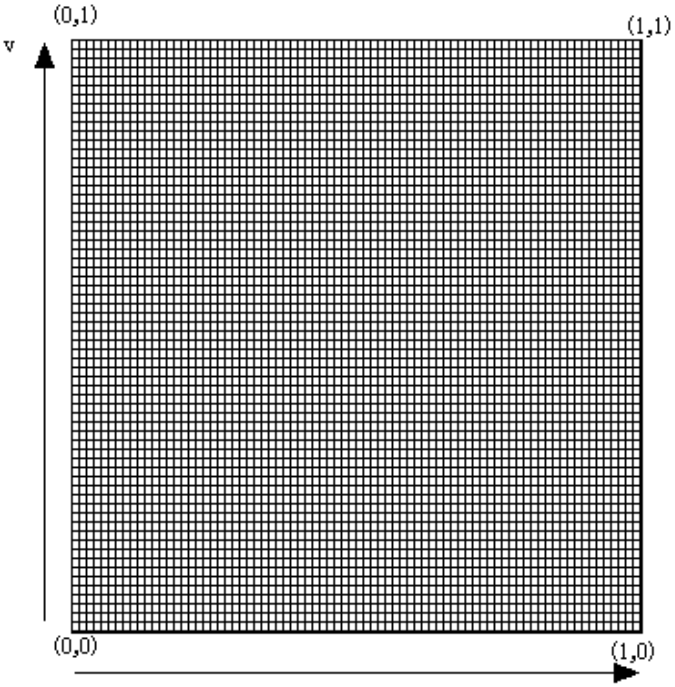
１．テクスチュアマッピングとは

3 次元オブジェクトの表面に対し，質感表現のための画像データを貼り付ける手法である．例えば煉瓦の壁を作る場合に，本来であれば 1 つ 1 つの煉瓦をモデリングする必要があるが，テクスチュアマッピングを用いることにより，容易かつ効率的にその疑似的な表現が可能となる．

※バンプマッピング：
数学的な処理（オブジェクト表面の法線ベクトルに対して揺らぎを与える）により，表面の凹凸を表現する手法である．

２．テクスチュアマッピングの座標系

テクスチュアマッピングに用いる画像は右図のような座標系(UV)をもつ．
※テクスチュアのピクセルをテクセルと呼ぶ



テクスチュア座標

３．テクスチュアマッピングを用いたプログラム

(準備)
tex1.bmp という名前で BMP 形式の画像ファイルを作成すること．

(準備)
```
Imports System.Drawing.Imaging
Imports Tao.OpenGl
Imports Tao.OpenGl.Gl
Imports Tao.OpenGl.Glu
Imports Tao.FreeGlut.Glut

Public Class Form1
    Private Structure Axis
        Dim X As Single
        Dim Y As Single
        Dim Z As Single
    End Structure

    Dim texture() As Integer = New Integer() {1}  'テクスチュアマッピング用
の配列
    Dim Rot As Axis
    Dim Tr As Axis
    Dim wWidth As Integer = 500
    Dim wHeight As Integer = 500

    Private Sub Command1_Click() Handles Button1.Click
        Timer1.Enabled = Not Timer1.Enabled
    End Sub

    Private Sub Form_Load() Handles MyBase.Load
        Timer1.Interval = 10
        Timer1.Enabled = False
        Button1.Text = "回転"
        Form_Initialize()
    End Sub

    Private Sub Form_Initialize()
        Initialize()
    End Sub

    Private Sub Timer1_Tick() Handles Timer1.Tick
        Rot.Y = Rot.Y + 5
        display()
    End Sub

    Private Sub SetView()
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()
        glFrustum(-2, 2, -2, 2, 2, 100)   ' gluPerspectiveの代替
        glMatrixMode(GL_MODELVIEW)
        glViewport(0, 0, wWidth, wHeight)
    End Sub

    Private Sub Initialize()
        glutInit()
        glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH Or GLUT_DOUBLE)
        glutInitWindowSize(wWidth, wHeight)
        glutInitWindowPosition(100, 100)
```

```
    glutCreateWindow("Texture 1")
    glutDisplayFunc(New DisplayCallback(AddressOf display))

    Me.Show()

    glDepthFunc(GL_LEQUAL)
    glEnable(GL_DEPTH_TEST)
    glShadeModel(GL_FLAT)

    'Fog
    glFogfv(GL_FOG_COLOR, New Single() {1, 1, 1, 1})
    glFogfv(GL_FOG_DENSITY, 0.1!)
    glDisable(GL_FOG)  'fog を ON にする場合には glEnable GL_FOG

    SetView()
    SetLight()
    LoadTexture()

    glutMainLoop()
End Sub

'テクスチュア画像の読み込み
Private Sub LoadTexture()
    ' 画像ファイルの読み込み
    ' 画像ファイルは自分の作成したファイル名
    ' パス名を明記するか，実行ファイルのあるところに置く
                          ([プロジェクトのディレクトリ]\bin\Debug)
    Dim image As Bitmap = New Bitmap("tex1.bmp")

    ' 画像情報の取得
    image.RotateFlip(RotateFlipType.RotateNoneFlipY) ' 上下反転
    Dim rectangle As Rectangle = New Rectangle(0, 0, _
                             image.Width, image.Height)
    Dim data As BitmapData = image.LockBits(rectangle, _
            ImageLockMode.ReadOnly, PixelFormat.Format24bppRgb)

    'テクスチュアの生成・設定
    glGenTextures(1, texture)
    glBindTexture(GL_TEXTURE_2D, texture(0))
    Gl.glTexImage2D(Gl.GL_TEXTURE_2D, 0, Gl.GL_RGB8, image.Width, _
            image.Height, 0, Gl.GL_BGR, Gl.GL_UNSIGNED_BYTE, _
            data.Scan0)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST)
End Sub

Private Sub SetLight()
    '平行光源（固定）
    Dim valueLight0(3) As Single
    Dim positionLight0(3) As Single

    'fillarray4f はユーザー定義
```

```
        valueLight0 = New Single() {0.7!, 0.7!, 0.7!, 1.0!}
        positionLight0 = New Single() {0.0!, 1.0!, 1.0!, 0.0!}

        glLightfv(GL_LIGHT0, GL_AMBIENT, valueLight0)
        glLightfv(GL_LIGHT0, GL_DIFFUSE, valueLight0)
        glLightfv(GL_LIGHT0, GL_SPECULAR, valueLight0)
        glLightfv(GL_LIGHT0, GL_POSITION, positionLight0)

        glEnable(GL_LIGHTING)
        glEnable(GL_LIGHT0)
    End Sub

    Private Sub display()
        Dim MaterialDiffuse(3) As Single
        Dim MaterialSpecular(3) As Single

        glMatrixMode(GL_MODELVIEW)
        glClear(GL_COLOR_BUFFER_BIT Or GL_DEPTH_BUFFER_BIT)
        glClearColor(0, 0, 0, 0)
        glLoadIdentity()
        glTranslatef(0, 0, -5)

        'テクスチュア ON
        glEnable(GL_TEXTURE_2D)
        glRotatef(Rot.Y, 0, 1, 0)

        MaterialDiffuse = New Single() {0.2, 0.2, 0.2, 1.0!}
        MaterialSpecular = New Single() {0.25, 0.25, 0.3, 1.0!}
        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, MaterialDiffuse)
        glMaterialfv(GL_FRONT, GL_SPECULAR, MaterialSpecular)

        glBegin(GL_QUADS)
        glNormal3f(0, 0, 1) '法線の設定
        'テクスチャア座標の設定：及び物体の作成
        glTexCoord2f(0, 0) : glVertex3f(-4, -4, 0)
        glTexCoord2f(1, 0) : glVertex3f(4, -4, 0)
        glTexCoord2f(1, 1) : glVertex3f(4, 4, 0)
        glTexCoord2f(0, 1) : glVertex3f(-4, 4, 0)
        glEnd()

        'テクスチュア OFF
        glDisable(GL_TEXTURE_2D)

        '球を描く
        glTranslatef(0, 0, 2)
        MaterialDiffuse = New Single() {0.4, 0.4, 0.1, 1.0!}
        MaterialSpecular = New Single() {0.3, 0.3, 0.1, 1.0!}
        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, _
                                        MaterialDiffuse(0))
        glMaterialfv(GL_FRONT, GL_SPECULAR, MaterialSpecular(0))
        glutSolidSphere(0.5, 20, 20)
        glutSwapBuffers()
    End Sub
End Class
```

４．テクスチュア画像の繰り返し

例えばブロック壁などを作成する場合，サイズの小さなテクスチュア画像を繰り返してマッピングすることが一般的である． ただし，マッピング画像を作成する場合には，横方向あるいは縦方向に接続できるよう作成する必要がある．

●設定方法

テクスチャアの環境設定を以下のように変える必要がある．（補間方法を線形補間にする）

```
glPixelStorei(GL_UNPACK_ALIGNMENT, 1)
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_LINEAR)
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_LINEAR)
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAP_FILTER, GL_LINEAR)
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR)
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL)
glEnable(GL_TEXTURE_2D)
glTexImage2D(GL_TEXTURE_2D, 0, 3, Tex(0).Width, Tex(0).Height, _
          0, GL_BGR, GL_UNSIGNED_BYTE, Tex(0).Data0)
```

物体上のテクスチュア座標の設定

```
glBegin(GL_QUADS)
    '法線の設定
    glNormal3f(0, 0, 1)
    'テクスチャア座標の設定：及び物体の作成
    glTexCoord2f(0, 0): glVertex3f(-1, -1, 0)
    glTexCoord2f(2, 0): glVertex3f( 1, -1, 0)
    glTexCoord2f(2, 2): glVertex3f( 1,  1, 0)
    glTexCoord2f(0, 2): glVertex3f(-1,  1, 0)
glEnd()
```

※テクスチュアマッピングに対するトランスペアレンシィ

マッピングを行なった物体に関しても，物体の色設定におけるアルファ値を１より小さくすることにより， トランスペアレンシィの設定が可能である．

５．サンプルコード

```
Imports System.Drawing.Imaging
Imports Tao.OpenGl
Imports Tao.OpenGl.Gl
Imports Tao.OpenGl.Glu
Imports Tao.FreeGlut.Glut

Public Class Form1
   Private Structure axis
      Dim X As Single
      Dim Y As Single
      Dim Z As Single
   End Structure

   ' テクスチュアデータ用クラス
   Private Class Texture
```

```
        Public Tex() As Integer
        Public Data As BitmapData

        Public Image As Bitmap
        Public Width As Integer
        Public Height As Integer
        Public Data0 As IntPtr

        Public Sub New(ByVal bmpname As String)
            Me.Tex = New Integer() {1}

            '画像ファイルの読み込み
            '（画像ファイルは自分の作成したファイル名）
            Me.Image = New Bitmap(bmpname)
            Me.Width = Me.Image.Width
            Me.Height = Me.Image.Height

            ' 画像情報の取得
            Me.Image.RotateFlip(RotateFlipType.RotateNoneFlipY) ' 上下反転
            Dim rect As Rectangle = New Rectangle(0, 0, Me.Width, Me.Height)
            Me.Data = Me.Image.LockBits(rect, ImageLockMode.ReadOnly, _
                                        PixelFormat.Format24bppRgb)
            Me.Data0 = Me.Data.Scan0

            'テクスチュアの生成・設定
            glGenTextures(1, Me.Tex)
            glBindTexture(GL_TEXTURE_2D, Me.Tex(0))
        End Sub
    End Class

    Dim Tex(10) As Texture 'テクスチュアマッピング用の配列

    Dim Rot As axis
    Dim Tr As axis
    Dim AddT As axis
    Dim AddR As axis
    Dim wWidth As Integer = 500
    Dim wHeight As Integer = 500

    Private Sub Button1_MouseDown(ByVal sender As Object, _
        ByVal e As System.Windows.Forms.MouseEventArgs) Handles
Button1.MouseDown
        Dim onLeftButton As Boolean = (e.Button =
Windows.Forms.MouseButtons.Left)

        AddT.X = 0 : AddT.Y = 0 : AddT.Z = IIf(onLeftButton, -1, 1)
        AddR.X = 0 : AddR.Y = 0 : AddR.Z = 0

        Timer1.Enabled = Not Timer1.Enabled
    End Sub

    Private Sub Button1_MouseUp() Handles Button1.MouseUp
        Timer1.Enabled = Not Timer1.Enabled
    End Sub
```

```
Private Sub Button2_MouseDown(ByVal sender As Object, _
          ByVal e As System.Windows.Forms.MouseEventArgs) _
          Handles Button2.MouseDown
   Dim onLeftButton As Boolean = (e.Button = _
             Windows.Forms.MouseButtons.Left)

   AddT.X = 0 : AddT.Y = 0 : AddT.Z = 0
   AddR.X = 0 : AddR.Y = IIf(onLeftButton, -1, 1) : AddR.Z = 0

   Timer1.Enabled = Not Timer1.Enabled
End Sub

Private Sub Button2_MouseUp() Handles Button2.MouseUp
   Timer1.Enabled = Not Timer1.Enabled
End Sub

Private Sub Form1_Load() Handles MyBase.Load
   Timer1.Interval = 10
   Timer1.Enabled = False
   Button1.Text = "移動"
   Button2.Text = "回転"
   Tr.Z = 6
   Initialize()
End Sub

Private Sub Timer1_Tick() Handles Timer1.Tick
   Rot.Y = Rot.Y + 1 * AddR.Y
   Tr.X = Tr.X + 0.2 * Math.Sin((Rot.Y) / 180 * Math.PI) * AddT.Z
   Tr.Z = Tr.Z + 0.2 * Math.Cos((Rot.Y) / 180 * Math.PI) * AddT.Z
   display()
End Sub

Private Sub SetView()
   glMatrixMode(GL_PROJECTION)
   glLoadIdentity()
   glFrustum(-2, 2, -2, 2, 2, 100)   '透視投影変換設定(gluPerspectiveの代替)
   glMatrixMode(GL_MODELVIEW)
   glViewport(0, 0, wWidth, wHeight)
End Sub

Private Sub Initialize()
   glutInit()
   glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH Or GLUT_DOUBLE)
   glutInitWindowSize(wWidth, wHeight)
   glutInitWindowPosition(100, 100)
   glutCreateWindow("Texture 2")
   glutDisplayFunc(New DisplayCallback(AddressOf display))
   Me.Show()
   glDepthFunc(GL_LEQUAL)
   glEnable(GL_DEPTH_TEST)
   glShadeModel(GL_FLAT)

   'Fog
   Dim fogclr(4) As Single
```

```
    fogclr(0) = 1 : fogclr(1) = 1 : fogclr(2) = 1 : fogclr(3) = 1
    glFogfv(GL_FOG_COLOR, fogclr(0))
    glFogfv(GL_FOG_DENSITY, 0.1!)
    glDisable(GL_FOG)  'fog を ON にする場合には glEnable GL_FOG
    glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)
    glEnable(GL_BLEND)

    SetView()
    SetLight()

    Tex(0) = New Texture("tex1.bmp")
    Tex(1) = New Texture("brick1.bmp")
    Tex(2) = New Texture("cloud1.bmp")

    glutMainLoop()
End Sub

Private Sub SetLight()
    '平行光源（固定）
    Dim valueLight0() As Single = New Single() {0.7!, 0.7!, 0.7!, 1.0!}
    glLightfv(GL_LIGHT0, GL_AMBIENT, valueLight0(0))
    glLightfv(GL_LIGHT0, GL_DIFFUSE, valueLight0(0))
    glLightfv(GL_LIGHT0, GL_SPECULAR, valueLight0(0))
    glLightfv(GL_LIGHT0, GL_POSITION, New Single() {0.0!, 1.0!, 1.0!, 0.0!})
    glEnable(GL_LIGHTING)
    glEnable(GL_LIGHT0)
End Sub

Private Sub Display()
    glMatrixMode(GL_MODELVIEW)

    glClearColor(0.3, 0.3, 0.4, 1)
    glClear(GL_COLOR_BUFFER_BIT Or GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    glTranslatef(-Tr.X, 0, -Tr.Z)
    glRotatef(Rot.Y, 0, 1, 0)

    glPushMatrix()
        glTranslatef(0, 0, 2)
        glMaterialfv(GL_FRONT, GL_AMBIENT, _
                             New Single() {0.17, 0.17, 0.17, 1})
        glMaterialfv(GL_FRONT, GL_DIFFUSE, _
                             New Single() {0.17, 0.17, 0.17, 1.0!})
        glMaterialfv(GL_FRONT, GL_SPECULAR, _
                             New Single() {0.3, 0.3, 0.3, 1.0!})
        glutSolidSphere(0.5, 20, 20)
    glPopMatrix()


    glPushMatrix()
        glTranslatef(-2, 0, 0)

        glMaterialfv(GL_FRONT, GL_AMBIENT, _
                             New Single() {0.17, 0.17, 0.17, 1})
```

```
    glMaterialfv(GL_FRONT, GL_DIFFUSE, _
                                New Single() {0.17, 0.17, 0.17, 1.0!})
    glMaterialfv(GL_FRONT, GL_SPECULAR, _
                                New Single() {0.3, 0.3, 0.3, 1.0!})

    'テクチュアの環境設定
    glPixelStorei(GL_UNPACK_ALIGNMENT, 1)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_LINEAR)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_LINEAR)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR)
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL)

    'テクスチュアを実行可能にする
    glEnable(GL_TEXTURE_2D)

    'テクスチュアイメージの設定
    glTexImage2D(GL_TEXTURE_2D, 0, 3, Tex(0).Width, Tex(0).Height, 0, _
                            GL_BGR, GL_UNSIGNED_BYTE, Tex(0).Data0)
    glBegin(GL_QUADS)
        '法線の設定
        glNormal3f(0, 0, 1)
        'テクスチャア座標の設定：及び物体の作成
        glTexCoord2f(0, 0) : glVertex3f(-1, -1, 0)
        glTexCoord2f(1, 0) : glVertex3f(1, -1, 0)
        glTexCoord2f(1, 1) : glVertex3f(1, 1, 0)
        glTexCoord2f(0, 1) : glVertex3f(-1, 1, 0)
    glEnd()
glPopMatrix()

glPushMatrix()
    glTranslatef(2, 0, 0)
    glMaterialfv(GL_FRONT, GL_AMBIENT, _
                                New Single() {0.17, 0.17, 0.17, 1})
    glMaterialfv(GL_FRONT, GL_DIFFUSE, _
                                New Single() {0.17, 0.17, 0.17, 1.0!})
    glMaterialfv(GL_FRONT, GL_SPECULAR, _
                                New Single() {0.3, 0.3, 0.3, 1.0!})
    glTexImage2D(GL_TEXTURE_2D, 0, 3, Tex(1).Width, Tex(1).Height, _
                        0, GL_BGR, GL_UNSIGNED_BYTE, Tex(1).Data0)
    glBegin(GL_QUADS)
        '法線の設定
        glNormal3f(0, 0, 1)
        'テクスチャア座標の設定：及び物体の作成
        glTexCoord2f(0, 0) : glVertex3f(-1, -1, 0)
        glTexCoord2f(2, 0) : glVertex3f(1, -1, 0)
        glTexCoord2f(2, 2) : glVertex3f(1, 1, 0)
        glTexCoord2f(0, 2) : glVertex3f(-1, 1, 0)
    glEnd()
glPopMatrix()
```

```
        '混合する物体は最後に描画
        glPushMatrix()
            glMaterialfv(GL_FRONT, GL_AMBIENT, New Single() {0.3, 0.3, 0.3, 0.5})
            glMaterialfv(GL_FRONT, GL_DIFFUSE, New Single() {0.2, 0.2, 0.2, 0.5})
            glMaterialfv(GL_FRONT, GL_SPECULAR, _
                                        New Single() {0.2, 0.2, 0.2, 0.5})
            glTexImage2D(GL_TEXTURE_2D, 0, 3, Tex(2).Width, Tex(2).Height, _
                                    0, GL_BGR, GL_UNSIGNED_BYTE, Tex(2).Data0)
            glBegin(GL_QUADS)
                '法線の設定
                glNormal3f(0, 0, 1)
                'テクスチャア座標の設定：及び物体の作成
                glTexCoord2f(0, 0) : glVertex3f(-3, -3, 2)
                glTexCoord2f(1, 0) : glVertex3f(3, -3, 2)
                glTexCoord2f(1, 1) : glVertex3f(3, 3, 2)
                glTexCoord2f(0, 1) : glVertex3f(-3, 3, 2)
            glEnd()
        glPopMatrix()

        'テクスチュア OFF
        glDisable(GL_TEXTURE_2D)
        glutSwapBuffers()
    End Sub
End Class
```

宿題：
テクスチュアマッピングを用いたプログラムを作成し，その実行画面をキャプチュアして提出すること．
次回の講義の際に回収する．

【最終課題（予告）】

これまでに学んできた OpenGL の諸機能を活用し，インタラクティブなシステム（これまでに作ってきたものを拡張したものでもよい）を作品として完成させなさい．

作品自体のインタラクティブな機能，完成度，面白さ等から総合的に評価する．

（テクスチュアマッピングの機能は必ず使用すること．）

期限：１月３１日（木）１２：５０まで

確認方法：１月３１日（木）３時限に個別に動作の確認を行う．

[OpenGL:9] アルファマッピング

１．アルファマッピングとは

アルファチャンネル(A)に設定された透過度を基に，テクスチュア画像を部分的に透過させるテクニックである．

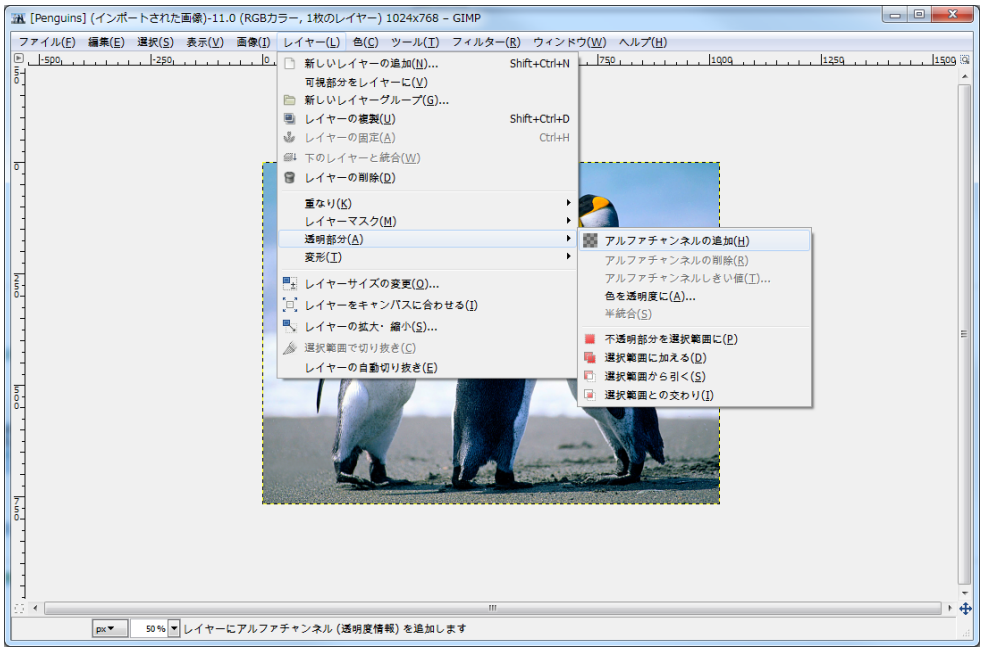
ポリゴンを用いずに画像データから擬似立体的な表現が可能であり，樹木等を擬似的に作成する手法として広く用いられている．

２．アルファマッピングのための画像（PNG 形式）をつくる（GIMP の場合）

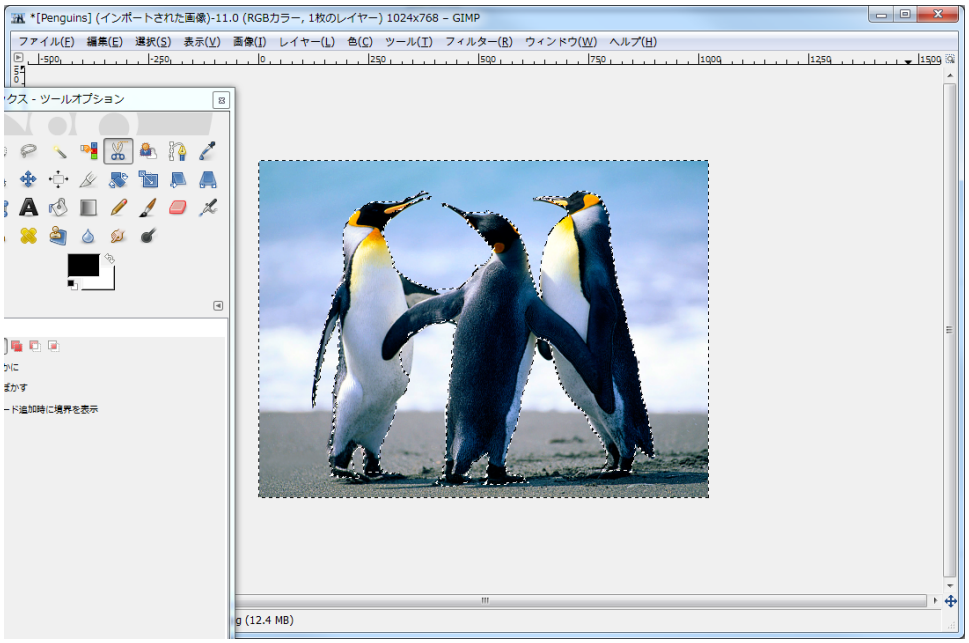
1)任意のカラー画像を開き，

レイヤー　->　透明部分　->　アルファチャンネルの追加

を選択する．

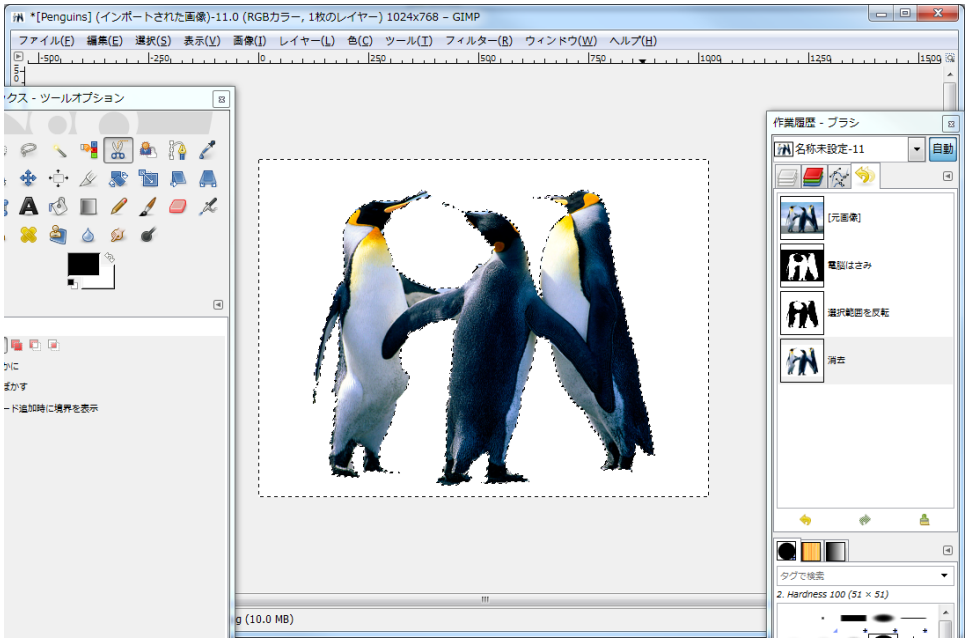


2)「電脳はさみ」ツールなどを使って不透明領域を選択.（確定は Enter キー）



3) 選択->選択範囲の反転　　（※ 2)で透明領域を選択した場合には反転は不要）

4) 選択領域を削除する（Delete キーを押す）

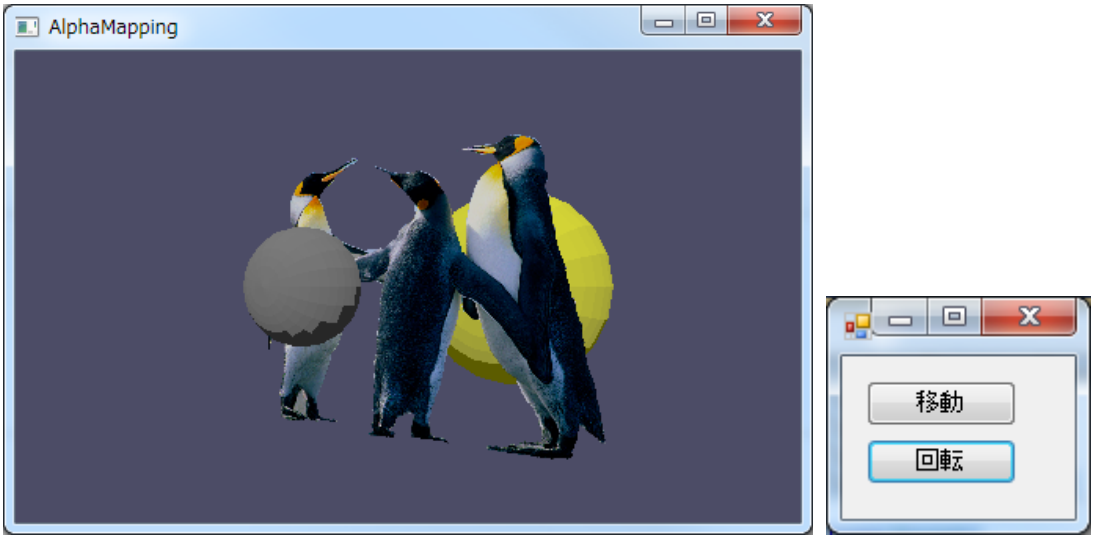


5)画像を PNG 形式でエクスポートする.

（ファイル->エクスポートで拡張子.png のファイル名で保存　例 Penguins.png）

6)作業を保存する（ファイル -> 名前を付けて保存 xcf 形式）

３．アルファマッピングのプログラム




```
Imports System.Drawing.Imaging
Imports Tao.OpenGl
Imports Tao.OpenGl.Gl
Imports Tao.OpenGl.Glu
Imports Tao.FreeGlut.Glut

Public Class Form1
    Private Structure axis
        Dim X As Single
        Dim Y As Single
        Dim Z As Single
    End Structure

    ' テクスチュアデータ用クラス
    Private Class Texture
        Public Tex() As Integer
        Public Data As BitmapData

        Public Image As Bitmap
        Public Width As Integer
        Public Height As Integer
        Public Data0 As IntPtr

        Public Sub New(ByVal bmpname As String)
            Me.Tex = New Integer() {1}

            '画像ファイルの読み込み
            '（画像ファイルは自分の作成したファイル名）
            Me.Image = New Bitmap(bmpname)
            Me.Width = Me.Image.Width
            Me.Height = Me.Image.Height
```

```
        ' 画像情報の取得
        Me.Image.RotateFlip(RotateFlipType.RotateNoneFlipY) ' 上下反転
        Dim rect As Rectangle = New Rectangle(0, 0, Me.Width, Me.Height)
        Me.Data = Me.Image.LockBits(rect, ImageLockMode.ReadOnly, _
                                    PixelFormat.Format32bppPArgb)
        Me.Data0 = Me.Data.Scan0

        'テクスチュアの生成・設定
        glGenTextures(1, Me.Tex)
        glBindTexture(GL_TEXTURE_2D, Me.Tex(0))
    End Sub
End Class

Dim Tex(10) As Texture 'テクスチュアマッピング用の配列

Dim Rot As axis
Dim Tr As axis
Dim AddT As axis
Dim AddR As axis
Dim wWidth As Integer = 500
Dim wHeight As Integer = 300

Private Sub Button1_MouseDown(ByVal sender As Object, _
    ByVal e As System.Windows.Forms.MouseEventArgs) Handles _
             Button1.MouseDown

    Dim onLeftButton As Boolean = _
     (e.Button = Windows.Forms.MouseButtons.Left)

    AddT.X = 0 : AddT.Y = 0 : AddT.Z = IIf(onLeftButton, -1, 1)
    AddR.X = 0 : AddR.Y = 0 : AddR.Z = 0

    Timer1.Enabled = Not Timer1.Enabled
End Sub

Private Sub Button1_MouseUp() Handles Button1.MouseUp
    Timer1.Enabled = Not Timer1.Enabled
End Sub

Private Sub Button2_MouseDown(ByVal sender As Object, _
          ByVal e As System.Windows.Forms.MouseEventArgs) _
          Handles Button2.MouseDown
    Dim onLeftButton As Boolean = (e.Button = _
            Windows.Forms.MouseButtons.Left)

    AddT.X = 0 : AddT.Y = 0 : AddT.Z = 0
    AddR.X = 0 : AddR.Y = IIf(onLeftButton, -1, 1) : AddR.Z = 0

    Timer1.Enabled = Not Timer1.Enabled
End Sub

Private Sub Button2_MouseUp() Handles Button2.MouseUp
    Timer1.Enabled = Not Timer1.Enabled
End Sub
```

```
    Private Sub Form1_Load() Handles MyBase.Load
        Timer1.Interval = 10
        Timer1.Enabled = False
        Button1.Text = "移動"
        Button2.Text = "回転"
        Tr.Z = 6
        Initialize()
    End Sub

    Private Sub Timer1_Tick() Handles Timer1.Tick
        Rot.Y = Rot.Y + 1 * AddR.Y
        Tr.X = Tr.X + 0.2 * Math.Sin((Rot.Y) / 180 * Math.PI) * AddT.Z
        Tr.Z = Tr.Z + 0.2 * Math.Cos((Rot.Y) / 180 * Math.PI) * AddT.Z
        display()
    End Sub

    Private Sub SetView()
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()
        gluPerspective(50, Width / Height, 1, 100)
        glMatrixMode(GL_MODELVIEW)
        glLoadIdentity()
        glViewport(0, 0, wWidth, wHeight)
    End Sub

    Private Sub Initialize()
        glutInit()
        glutInitDisplayMode(GLUT_RGBA Or GLUT_DEPTH Or GLUT_DOUBLE)
        glutInitWindowSize(wWidth, wHeight)
        glutInitWindowPosition(100, 100)
        glutCreateWindow("AlphaMapping")
        glutReshapeFunc(New ReshapeCallback(AddressOf SetViewReshape))
        glutDisplayFunc(New DisplayCallback(AddressOf display))

        Me.Show()

        glDepthFunc(GL_LEQUAL)
        glEnable(GL_DEPTH_TEST)
        glShadeModel(GL_FLAT)

        'Fog
        Dim fogclr(4) As Single
        fogclr(0) = 1 : fogclr(1) = 1 : fogclr(2) = 1 : fogclr(3) = 1
        glFogfv(GL_FOG_COLOR, fogclr(0))
        glFogfv(GL_FOG_DENSITY, 0.1!)
        glDisable(GL_FOG)  'fog を ON にする場合には glEnable GL_FOG
        glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)
        glEnable(GL_BLEND)

        SetView()
        SetLight()

        Tex(0) = New Texture("Penguins.png")
```

```
    glutMainLoop()
End Sub

 Sub Reshape(ByVal nWidth As Integer, ByVal nHeight As Integer)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluPerspective(50, nWidth / nHeight, 1, 100)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()
    glViewport(0, 0, nWidth, nHeight)
End Sub


Private Sub SetLight()
    '平行光源（固定）
    Dim valueLight0() As Single = New Single() {0.7!, 0.7!, 0.7!, 1.0!}

    glLightfv(GL_LIGHT0, GL_AMBIENT, valueLight0(0))
    glLightfv(GL_LIGHT0, GL_DIFFUSE, valueLight0(0))
    glLightfv(GL_LIGHT0, GL_SPECULAR, valueLight0(0))
    glLightfv(GL_LIGHT0, GL_POSITION,
                    New Single() {0.0!, 1.0!, 1.0!, 0.0!})

    glEnable(GL_LIGHTING)
    glEnable(GL_LIGHT0)
End Sub

Private Sub Display()
    glMatrixMode(GL_MODELVIEW)

    glClearColor(0.3, 0.3, 0.4, 1)
    glClear(GL_COLOR_BUFFER_BIT Or GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    glTranslatef(-Tr.X, 0, -Tr.Z)
    glRotatef(Rot.Y, 0, 1, 0)

    glPushMatrix()
       glTranslatef(0, 0, 2)
       glMaterialfv(GL_FRONT, GL_AMBIENT,
          New Single() {0.17, 0.17, 0.17, 1.0})
       glMaterialfv(GL_FRONT, GL_DIFFUSE,
          New Single() {0.17, 0.17, 0.17, 1.0!})
       glMaterialfv(GL_FRONT, GL_SPECULAR,
          New Single() {0.3, 0.3, 0.3, 1.0!})
       glutSolidSphere(0.5, 20, 20)
    glPopMatrix()

    glPushMatrix()
       glTranslatef(0, 0, -5)
       glMaterialfv(GL_FRONT, GL_AMBIENT,
          New Single() {0.4, 0.4, 0.0, 1.0})
       glMaterialfv(GL_FRONT, GL_DIFFUSE,
          New Single() {0.4, 0.4, 0.0, 1.0!})
```

```
            glMaterialfv(GL_FRONT, GL_SPECULAR,
                New Single() {0.3, 0.3, 0.3, 1.0!})
            glutSolidSphere(2, 20, 20)
        glPopMatrix()


        glPushMatrix()
            glTranslatef(0, 0, 0)

            glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE,
                New Single() {0.5, 0.5, 0.5, 1.0})
            glMaterialfv(GL_FRONT, GL_SPECULAR,
                New Single() {0.2, 0.2, 0.2, 1.0})


            'テクチュアの環境設定
            glPixelStorei(GL_UNPACK_ALIGNMENT, 4)
            glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP)
            glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP)
            glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER,
                    GL_NEAREST)
            glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,
                    GL_NEAREST)


            glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE)
            glAlphaFunc(GL_GREATER, 0.5)

            'テクスチュアを実行可能にする
            glEnable(GL_TEXTURE_2D)
            glEnable(GL_ALPHA_TEST)


            'テクスチュアイメージの設定
            glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, Tex(0).Width,
                Tex(0).Height, 0,GL_BGRA, GL_UNSIGNED_BYTE, Tex(0).Data0)
            glBegin(GL_QUADS)
                '法線の設定
                glNormal3f(0, 0, 1)
                'テクスチャア座標の設定：及び物体の作成
                glTexCoord2f(0, 0) : glVertex3f(-3, -2, 0)
                glTexCoord2f(1, 0) : glVertex3f(3, -2, 0)
                glTexCoord2f(1, 1) : glVertex3f(3, 2, 0)
                glTexCoord2f(0, 1) : glVertex3f(-3, 2, 0)
            glEnd()
        glPopMatrix()

        'テクスチュア OFF
        glDisable(GL_TEXTURE_2D)
        glDisable(GL_ALPHA_TEST)   'この行を追加（AlphaTest の終了）

        glutSwapBuffers()
    End Sub
End Class
```

４．球体へのマッピングプログラム（変更部分のみ）

```
' テクスチュアデータ用クラス
Private Class Texture
    Public Tex() As Integer
    Public Data As BitmapData

    Public Image As Bitmap
    Public Width As Integer
    Public Height As Integer
    Public Data0 As IntPtr

    Public Sub New(ByVal bmpname As String)
        Me.Tex = New Integer() {1}

        '画像ファイルの読み込み
        '（画像ファイルは自分の作成したファイル名）
        Me.Image = New Bitmap(bmpname)
        Me.Width = Me.Image.Width
        Me.Height = Me.Image.Height

        ' 画像情報の取得
        Me.Image.RotateFlip(RotateFlipType.RotateNoneFlipY) ' 上下反転
        Dim rect As Rectangle = New Rectangle(0, 0, Me.Width, Me.Height)
        Me.Data = Me.Image.LockBits(rect, ImageLockMode.ReadOnly, _
                                    PixelFormat.Format32bppPArgb)
        Me.Data0 = Me.Data.Scan0

        'テクスチュアの生成・設定
        glGenTextures(1, Me.Tex)
        glBindTexture(GL_TEXTURE_2D, Me.Tex(0))
    End Sub
End Class


Private Sub Initialize()
    （略）
    Tex(0) = New Texture("penguins.png")
    Tex(1) = New Texture("PenguinsNon.png")
    glutMainLoop()
End Sub


Private Sub Display()
    glMatrixMode(GL_MODELVIEW)

    glClearColor(0.3, 0.3, 0.4, 1)
    glClear(GL_COLOR_BUFFER_BIT Or GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    glTranslatef(-Tr.X, 0, -Tr.Z)
    glRotatef(Rot.Y, 0, 1, 0)

    glPushMatrix()
        glTranslatef(0, 0, 2)
```

```
    glMaterialfv(GL_FRONT, GL_AMBIENT,
        New Single() {0.17, 0.17, 0.17, 1.0})
    glMaterialfv(GL_FRONT, GL_DIFFUSE,
        New Single() {0.17, 0.17, 0.17, 1.0!})
    glMaterialfv(GL_FRONT, GL_SPECULAR,
        New Single() {0.3, 0.3, 0.3, 1.0!})
    glutSolidSphere(0.5, 20, 20)
glPopMatrix()

glPushMatrix()
    glTranslatef(0, 0, -5)
    glMaterialfv(GL_FRONT, GL_AMBIENT,
        New Single() {0.4, 0.4, 0.0, 1.0})
    glMaterialfv(GL_FRONT, GL_DIFFUSE,
        New Single() {0.4, 0.4, 0.0, 1.0!})
    glMaterialfv(GL_FRONT, GL_SPECULAR,
        New Single() {0.3, 0.3, 0.3, 1.0!})
    glutSolidSphere(2, 20, 20)
glPopMatrix()


glPushMatrix()
    glTranslatef(0, 0, 0)

    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE,
        New Single() {0.5, 0.5, 0.5, 1.0})
    glMaterialfv(GL_FRONT, GL_SPECULAR,
        New Single() {0.2, 0.2, 0.2, 1.0})


    'テクチュアの環境設定
    glPixelStorei(GL_UNPACK_ALIGNMENT, 4)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER,
                                                GL_NEAREST)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,
                                                GL_NEAREST)
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE)
    glAlphaFunc(GL_GREATER, 0.5)


    'テクスチュアを実行可能にする
    glEnable(GL_TEXTURE_2D)
    glEnable(GL_ALPHA_TEST)

    'ティーポットへのテクスチュア画像の貼りつけ
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, Tex(1).Width,
        Tex(1).Height, 0, GL_BGRA, GL_UNSIGNED_BYTE, Tex(1).Data0)
    glutSolidTeapot(1)

    glTranslatef(2, 0, 0)
    glRotatef(Rot.Y * 2, 0, 1, 0)
    glRotatef(-90, 1, 0, 0)
```

```
        '球体への画像貼りつけ（GluSphere でのマッピング座標生成）
        Dim Sph As New GLUquadric
        Sph = gluNewQuadric()
        gluQuadricDrawStyle(Sph, GLU_FILL)
        gluQuadricNormals(Sph, GLU_SMOOTH)
        gluQuadricTexture(Sph, GL_TRUE)

        glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, Tex(0).Width,
            Tex(0).Height, 0, GL_BGRA, GL_UNSIGNED_BYTE, Tex(0).Data0)
        gluSphere(Sph, 1, 10, 10)
    glPopMatrix()

    '以下追加，テクスチュア，アルファテスト OFF
    glDisable(GL_TEXTURE_2D)
    glDisable(GL_ALPHA_TEST)

End Sub
```

【最終課題】

これまでに学んできた OpenGL の諸機能を活用し，インタラクティブなシステム（これまでに作ってきたものを拡張したものでもよい）を作品として完成させなさい．

作品自体のインタラクティブな機能，完成度，面白さ等から総合的に評価する．

（テクスチュアマッピングの機能は必ず使用すること．）

期限：１月３１日（木）１２：５０まで

確認方法：１月３１日（木）３時限に個別にチェックする．

（すぐに動作できるように準備をしておくこと！）

提出方法：プログラム等のデータは指定した PC に入れること（USB メモリ等を利用）

（作成したフォルダを Zip 形式等でまとめて提出する．

画像ファイルは bin¥debug フォルダの中に入れておき，フォルダ内の exe ファイルを起動したときに，自動的に読み込めるようにしておくこと．）

コンピュータグラフィックスのまとめ

到達目標：

[1]3 次元コンピュータグラフィックス（CG）の基礎理論を理解する.

[2]3 次元 CG を用いたインタラクティブシステム構築のための基礎技術を修得する.

[3]3 次元 CG の応用技術を理解する.

講義内容：

[01]3 次元 CG の基礎(1)－3 次元データモデルの構築－

[02]3 次元 CG の基礎(2)－3 次元モデルの 2 次元平面への投影－

[03]3 次元 CG の基礎(3)－隠線処理の手法－

[04]OpenGL の制御とプログラミング

[05]射影変換

[06]モデリング変換

[07]ロボットアームの制御

[08]OpenGL によるアニメーション

[09]シェーディング

[10]照光処理

[11]混合（透過・アンチエイリアシング・フォグ）

[12]テクスチュアマッピング

[13]アルファマッピング

[14]3 次元 CG の応用技術

コンピュータグラフィックスの応用技術

(1)　バーチャルリアリティ（VR）：仮想現実（感）
(2)　ミックスドリアリティ（MR）：複合現実（感）
(3)　オーグメンティドリアリティ（AR）：拡張現実（感）

AR Toolkit

(1) AR TOOLKIT のインストール

a) Sourceforge から artoolkit の最新版をダウンロードする.
（現時点で Release 2.72.1 が最新）
http://sourceforge.net/projects/artoolkit/files/
ARToolKit-2.72.1-bin-win32.zip をダウンロードし,
任意のフォルダ(ディレクトリ)に展開する.

b) ARToolKit というフォルダができるので，そのフォルダを C ドライブ直下に移動する。

(2)GLUT のインストール

a)下記のサイトから glut-3.7.6-bin.zip (117 KB)をダウンロードする.
http://www.xmission.com/~nate/glut.html

b)適当なフォルダに展開し，フォルダ内の glut-3.7.6-bin を
c:¥Program Files¥ARToolKit 下に移動する.

c) glut32.dll を C:¥Windows¥System32 下に移動する.
(64bitOS の場合は，C:¥Windows¥SysWOW64)

(3) AR TOOLKIT の実行

C:¥Program Files¥ARToolkit¥bin 下に実行可能ファイルが入っている.
例えば SimpleLite.exe を実行してみる.

[補足]カメラが正常に起動しない場合

　他のカメラ・スキャナ等のデバイスなどが動作していると正常に起動しない場合がある.

(1)カメラ名を確認する.

コントロールパネル->システム->デバイスマネージャ

イメージングデバイス下から接続したカメラ名を探す.

表示されている名前がカメラ名である.

(2) ARToolKit¥bin￥Data の中の WDM_camera_flipV.xml をテキストエディタ（メモ帳）で

開き，下の行を探す.

```
<camera show_format_dialog="true" friendly_name="PGR">
```

friendly_name の属性を自分のカメラ名 XXX に変更する.

```
<camera show_format_dialog="true" friendly_name="XXX">
```


Hiro