

計算機ソフトウェア#3

久野 靖*

1992.5.13

本日の内容

本日はまず岡野さんの論文紹介からお願いしたいと思います。その後少々時間を頂いて、NeXT Stepの例題の「宿題」の回答とちょっとした拡張を説明してみたいと思います。なお、本日もミニアンケートを(あとで)出しますので、2・3日中に答を投稿してください。

1 前回のデモプログラムの構造変更

復習しておく、前回の宿題とは、mouseDownでいきなり大きさ等を変更してしまうのではなく、自分自身に「大きくなれ」とか「右に回れ」というメッセージを送ることによって機能を実現せよ、というものでした。(なぜそうしたいのでしたっけ?) ともかくそういうわけで、変更されるのはTestView.[hm]のみです。

```
#import <appkit/View.h>

@interface TestView : View {
    NXCoord size;
    NXCoord angle;
}
+ newFrame: (const NXRect*)frameRect;
- mouseDown: (NXEvent*)theEvent;
- drawSelf: (const NXRect*)rects : (int)rectCount;
- enlarge;
- shrink;
- rotateL;
- rotateR;
@end

#import "TestView.h"

@implementation TestView

+ newFrame: (const NXRect*)frameRect {
    self = [super newFrame: frameRect];
    size = 100.0;
    angle = 0.0;
    return self;
}

- mouseDown: (NXEvent*)theEvent {
    if(theEvent->flags & NX_CONTROLMASK)
```

*筑波大学大学院経営システム科学専攻

```

        [self rotateR];
    else if(theEvent->flags & NX_ALTERNATEMASK)
        [self rotateL];
    else if(theEvent->flags & NX_SHIFTMASK)
        [self shrink];
    else
        [self enlarge];
}

- drawSelf: (const NXRect*)rects :(int)rectCount {
    NXRect aRect;
    [self translate: 150.0 : 150.0];
    [self rotate: angle];
    NXSetRect(&aRect, -size/2.0, -size/2.0, size, size);
    NXEraseRect(&aRect);
    PSsetgray(NX_BLACK);
    NXFrameRect(&aRect);
    [self rotate: -angle];
    [self translate: -150.0 : -150.0];
}

- enlarge { size = size + 10.0; [self display]; }
- shrink { size = size - 10.0; [self display]; }
- rotateR { angle = angle + 3.0; [self display]; }
- rotateL { angle = angle - 3.0; [self display]; }

@end

```

別に、簡単だったでしょう？

2 例題 2.0: 対話のための「部品」 — ボタン

さて、せっかく TestView に「大きくなれ」とかいえるようになったのですから、マウスボタン以外からそういつてみたいですね。そこで、件の窓にいくつか「ボタン」を追加して、そのボタンを押すとボタンが TestView に「大きくなれ」などのメッセージを送るようにしてみませう。ということは、今度の変更するのは main.m のみです。

```

#import <appkit/appkit.h>
#import "TestView.h"

main() {
    id myWin, myPanel, myMenu, myView;
    id btn1, btn2, btn3, btn4;
    NXRect aRect;
    [Application new];
    NXSetRect(&aRect, 100.0, 350.0, 300.0, 350.0);
    myWin = [Window newContent: &aRect
                    style: NX_TITLEDSTYLE
                    backing: NX_BUFFERED
                    buttonMask: NX_MINIATURIZEBUTTONMASK
                    defer: NO];
    [myWin setTitle: "CompSoftDemo 2.0"];
    NXSetRect(&aRect, 0.0, 50.0, 300.0, 300.0);
    myView = [TestView newFrame: &aRect];
    [[myWin contentView] addSubview: myView];
    NXSetRect(&aRect, 10.0, 10.0, 60.0, 30.0);
    btn1 = [[Button new] initWithFrame: &aRect title: "Enlarge" tag: 1

```

```

        target: myView action: @selector(enlarge) key: ' ' enabled: TRUE];
[[myWin contentView] addSubview: btn1];
NXSetRect(&aRect, 85.0, 10.0, 60.0, 30.0);
btn2 = [[Button new] initWithFrame: &aRect title: "Shrink" tag: 2
        target: myView action: @selector(shrink) key: ' ' enabled: TRUE];
[[myWin contentView] addSubview: btn2];
NXSetRect(&aRect, 160.0, 10.0, 60.0, 30.0);
btn3 = [[Button new] initWithFrame: &aRect title: "Left" tag: 3
        target: myView action: @selector(rotateL) key: ' ' enabled: TRUE];
[[myWin contentView] addSubview: btn3];
NXSetRect(&aRect, 235.0, 10.0, 60.0, 30.0);
btn4 = [[Button new] initWithFrame: &aRect title: "Right" tag: 4
        target: myView action: @selector(rotateR) key: ' ' enabled: TRUE];
[[myWin contentView] addSubview: btn4];
NXSetRect(&aRect, 100.0, 700.0, 300.0, 40.0);
myPanel = [Panel newContent: &aRect
           style: NX_TITLEDSTYLE
           backing: NX_BUFFERED
           buttonMask: NX_CLOSEBUTTONMASK
           defer: YES];
[myPanel setTitle: "About CompSoftDemo"];
[myPanel removeFromEventMask: (NX_KEYDOWNMASK|NX_KEYUPMASK)];
myMenu = [Menu newTitle: "CS Demo"];
[[myMenu addItem: "Info..."
  action: @selector(orderFront:)
  keyEquivalent: 'i'] setTarget: myPanel];
[myMenu addItem: "Hide" action: @selector(hide:) keyEquivalent: 'h'];
[myMenu addItem: "Quit" action: @selector(terminate:) keyEquivalent: 'q'];
[myMenu sizeToFit];
[NXApp setMainMenu: myMenu];
[myWin display];
[myWin orderFront: nil];
[myWin makeKeyWindow];
[NXApp run];
[NXApp free];
}

```

3 例題 2.1: 対話のための「部品」 — スライダ

ボタンを 1 回つつくとちょっと動く、というのはそろそろ飽きましたから、もっと「連続的に」動かしてみようと思います。そこで「ボタン」の代わりに「スライダ」を使うようにします。

```

#import <appkit/appkit.h>
#import "TestView.h"

main() {
    id myWin, myPanel, myMenu, myView;
    id sli1, sli2;
    NXRect aRect;
    [Application new];
    NXSetRect(&aRect, 100.0, 350.0, 300.0, 350.0);
    myWin = [Window newContent: &aRect
           style: NX_TITLEDSTYLE
           backing: NX_BUFFERED
           buttonMask: NX_MINIATURIZEBUTTONMASK
           defer: NO];
    [myWin setTitle: "CompSoftDemo 2.1"];
    NXSetRect(&aRect, 0.0, 50.0, 300.0, 300.0);

```

```

myView = [TestView newFrame: &aRect];
[[myWin contentView] addSubview: myView];
NXSetRect(&aRect, 10.0, 10.0, 135.0, 30.0);
sli1 = [[Slider new] initWithFrame: &aRect];
[sli1 setMinValue: 10.0];
[sli1 setMaxValue: 200.0];
[sli1 setFloatValue: 100.0];
[[sli1 cell] setTarget: myView];
[[sli1 cell] setAction: @selector(sizeTo:)];
[sli1 sizeToFit];
[[myWin contentView] addSubview: sli1];
NXSetRect(&aRect, 155.0, 10.0, 135.0, 30.0);
sli2 = [[Slider new] initWithFrame: &aRect];
[sli2 setMinValue: 0.0];
[sli2 setMaxValue: 360.0];
[[sli2 cell] setTarget: myView];
[[sli2 cell] setAction: @selector(angleTo:)];
[sli2 sizeToFit];
[[myWin contentView] addSubview: sli2];
NXSetRect(&aRect, 100.0, 700.0, 300.0, 40.0);
myPanel = [Panel newContent: &aRect
            style: NX_TITLEDSTYLE
            backing: NX_BUFFERED
            buttonMask: NX_CLOSEBUTTONMASK
            defer: YES];
[myPanel setTitle: "About CompSoftDemo"];
[myPanel removeFromEventMask: (NX_KEYDOWNMASK|NX_KEYUPMASK)];
myMenu = [Menu newTitle: "CS Demo"];
[[myMenu addItem: "Info..."
  action: @selector(orderFront:)
  keyEquivalent: 'i'] setTarget: myPanel];
[myMenu addItem: "Hide" action: @selector(hide:) keyEquivalent: 'h'];
[myMenu addItem: "Quit" action: @selector(terminate:) keyEquivalent: 'q'];
[myMenu sizeToFit];
[NXApp setMainMenu: myMenu];
[myWin display];
[myWin orderFront: nil];
[myWin makeKeyWindow];
[NXApp run];
[NXApp free];
}

```

するとスライダは掴んで動かされるごとに TestView に向かって「sizeTo: self」とか「angleTo: self」とかいうメッセージを送ってくれます。そこでこれらに対処するように TestView.[hm] も修正します。

```

#import <appkit/View.h>

@interface TestView : View {
    NXCoord size;
    NXCoord angle;
}
+ newFrame: (const NXRect*)frameRect;
- mouseDown: (NXEvent*)theEvent;
- drawSelf: (const NXRect*)rects : (int)rectCount;
- enlarge;
- shrink;
- rotateL;
- rotateR;

```

```

- sizeTo: (id)aControl;
- angleTo: (id)aControl;
@end

#import "TestView.h"
#import <appkit/Control.h>

@implementation TestView

+ newFrame: (const NXRect*)frameRect {
    self = [super newFrame: frameRect];
    size = 100.0;
    angle = 0.0;
    return self;
}

- mouseDown: (NXEvent*)theEvent {
    if(theEvent->flags & NX_CONTROLMASK)
        [self rotateR];
    else if(theEvent->flags & NX_ALTERNATEMASK)
        [self rotateL];
    else if(theEvent->flags & NX_SHIFTMASK)
        [self shrink];
    else
        [self enlarge];
}

- drawSelf: (const NXRect*)rects :(int)rectCount {
    NXRect aRect;
    NXEraseRect(&bounds);
    [self translate: 150.0 : 150.0];
    [self rotate: angle];
    NXSetRect(&aRect, -size/2.0, -size/2.0, size, size);
    NXEraseRect(&aRect);
    PSsetgray(NX_BLACK);
    NXFrameRect(&aRect);
    [self rotate: -angle];
    [self translate: -150.0 : -150.0];
}

- enlarge { size = size + 10.0; [self display]; }
- shrink { size = size - 10.0; [self display]; }
- rotateR { angle = angle + 3.0; [self display]; }
- rotateL { angle = angle - 3.0; [self display]; }

- sizeTo: (id)aControl {
    size = [aControl floatValue]; [self display];
}
- angleTo: (id)aControl {
    angle = [aControl floatValue]; [self display];
}
@end

```