

計算機ソフトウェア#6

久野 靖*

1992.5.27

本日の内容

今日も、デモを改造された方の「成果」があれば拝見したいですね。

それとまず、論文紹介の件ですが、尾崎さんが「Function Minimization and Automatic Differentiation Using C++」というのを紹介して下さいます。ただし6/17または6/24がいいそうです。ですから、本日の三橋さんの後は「中谷さん→梶田さん→尾崎さん→豊島?」ただし梶田さんと尾崎さんと相談して入れ替わってもよい、ということにいたしませう。

後半のデモですが、長らくお楽しみ頂きました一連のデモプログラムは今回で一応おしまいにしようと思います。で、IBの資料を配りますので、これを(読むのは全員読むとして)有志の方にデモしていただくようお願いしたい。それが終わったらデモプログラムをIB経由に直すというのを1回やるつもりです。

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 インターフェースビルダ

- IBとは
- IBの原理/IBは結局何をやっているか
- .nibファイル
- 使い方
- 紹介者の選定

2 クラス構造の整理

では前回と動作は同じながら、クラス構造を整理してみます。クラス階層図をまず示します。ではこれを上から順におさらいしましょう。

```
Graphic.h -----  
#import <appkit/appkit.h>  
  
@interface Graphic : Object {  
    float x, y;  
}  
- setPosX: (const float)xpos Y: (const float)ypos;
```

*筑波大学大学院経営システム科学専攻

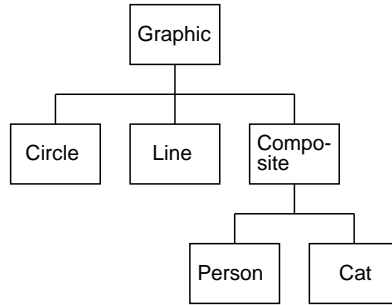


図 1: 整理されたクラス階層

```

- drawIt;
@end
Graphic.m -----
#import "Graphic.h"

@implementation Graphic
+ new {
    self = [super new];
    x = y = 0.0;
    return self;
}
- setPosX: (const float)xpos Y: (const float) ypos {
    x = xpos;
    y = ypos;
    return self;
}
- drawIt {
}
@end
Circle.h -----
#import <appkit/appkit.h>
#import "Graphic.h"

@interface Circle : Graphic {
    float r;
    id view;
}
+ newRadius: (const float)radius;
@end
Circle.m -----
#import "Circle.h"
#import <dpsclient/psops.h>

@implementation Circle
+ newRadius: (const float)radius {
    self = [super new];
    r = radius;
    return self;
}
- drawIt {
    PSnewpath();
    PSarc(x, y, r, 0.0, 360.0);
    PSsetgray(0.0);
    PSstroke();
}

```

```

@end
Line.h -----
#import "Graphic.h"

@interface Line : Graphic {
    float x1, y1;
}
+ newStartX: (float)px Y: (float) py EndX: (float)px1 Y: (float)py1;
@end
Line.m -----
#import "Line.h"
#import <dpsclient/psops.h>

@implementation Line
+ newStartX: (float)px Y: (float) py EndX: (float)px1 Y: (float)py1; {
    self = [super new];
    x = px; y = py; x1 = px1; y1= py1;
    return self;
}
- drawIt {
    PSnewpath();
    PSmoveto(x, y);
    PSlineto(x1, y1);
    PSsetgray(0.0);
    PSstroke();
}
@end
Composite.h -----
#import "Graphic.h"

@interface Composite : Graphic {
    id part[20];
    int count;
}
- addPart: p;
@end
Composite.m -----
#import "Composite.h"

@implementation Composite
+ newAtX: (float)px Y: (float)py {
    self = [super new];
    x = px; y = py; count = 0;
    return self;
}
- addPart: p {
    part[count++] = p;
}
- drawIt {
    int i;
    PSgsave();
    PStranslate(x, y);
    for(i = 0; i < count; ++i) [part[i] drawIt];
    PSgrestore();
}
@end
Person.h -----
#import <appkit/appkit.h>
#import "Composite.h"

```

```

@interface Person : Composite {
}
+ newAtX: (float)px Y: (float) py Height: (float) ph;
@end

Person.m -----
#import "Person.h"
#import "Circle.h"
#import "Line.h"

@implementation Person
+ newAtX: (float)px Y: (float)py Height: (float)ph {
    float d = ph / 8.0;
    self = [[super new] setPosX: px Y: py];
    [self addPart: [[Circle newRadius: d*2.0] setPosX: 0.0 Y: d*4.0]];
    [self addPart: [Line newStartX: 0.0 Y: d*2.0 EndX: 0.0 Y: -d*2.0]];
    [self addPart: [Line newStartX: 0.0 Y: 0.0 EndX: d*2.0 Y: d]];
    [self addPart: [Line newStartX: 0.0 Y: 0.0 EndX: -d*2.0 Y: d]];
    [self addPart: [Line newStartX: 0.0 Y: -d*2.0 EndX: d*2.0 Y: -d*4.0]];
    [self addPart: [Line newStartX: 0.0 Y: -d*2.0 EndX: -d*2.0 Y: -d*4.0]];
    return self;
}
@end

Cat.h -----
#import <appkit/appkit.h>
#import "Composite.h"

@interface Cat : Composite {
}
+ newAtX: (float)px Y: (float) py Height: (float) ph;
@end

Cat.m -----
#import "Cat.h"
#import "Circle.h"
#import "Line.h"

@implementation Cat
+ newAtX: (float)px Y: (float)py Height: (float)ph {
    float d = ph / 4.0;
    self = [[super new] setPosX: px Y: py];
    [self addPart: [[Circle newRadius: d*2.0] setPosX: -d*3.0 Y: 0.0]];
    [self addPart: [Line newStartX: -d*4.0 Y: 0.0 EndX: -d*3.0 Y: 0.0]];
    [self addPart: [Line newStartX: -d*4.0 Y: 0.0 EndX: -d*3.0 Y: -d/2.0]];
    [self addPart: [Line newStartX: -d*4.0 Y: 0.0 EndX: -d*3.0 Y: -d]];
    [self addPart: [Line newStartX: -d Y: 0.0 EndX: d*3.0 Y: 0.0]];
    [self addPart: [Line newStartX: 0.0 Y: 0.0 EndX: d Y: -d*3.0]];
    [self addPart: [Line newStartX: 0.0 Y: 0.0 EndX: -d Y: -d*3.0]];
    [self addPart: [Line newStartX: d*3.0 Y: 0.0 EndX: d*2.0 Y: -d*3.0]];
    [self addPart: [Line newStartX: d*3.0 Y: 0.0 EndX: d*4.0 Y: -d*3.0]];
    [self addPart: [Line newStartX: d*3.0 Y: 0.0 EndX: d*4.0 Y: d]];
    return self;
}
@end

```

TestView.[hm] と main.m は変わりませんが一応掲載。

```

TestView.h -----
#import <appkit/View.h>
#import "Graphic.h"

```

```

@interface TestView : View {
    id objs[100];
    int count;
}
+ newFrame: (const NXRect*)frameRect;
- drawSelf: (const NXRect*)rects : (int)rectCount;
- addOne: (id)anObj;
@end
TestView.m -----
#import "TestView.h"
#import "Circle.h"
#import <dpsclient/psops.h>

@implementation TestView
+ newFrame: (const NXRect*)frameRect {
    self = [super newFrame: frameRect];
    count = 0;
    return self;
}
- addOne: (id)anObj {
    objs[count++] = anObj;
}
- drawSelf: (const NXRect*)rects :(int)rectCount {
    int i;
    NXEraseRect(&bounds);
    for(i = 0; i < count; ++i)
        [objs[i] drawIt];
}
@end
Main.m -----
#import <appkit/appkit.h>
#import "TestView.h"
#import "Person.h"
#import "Cat.h"

main() {
    id myWin, myPanel, myMenu, myView;
    id per1, cat2;
    NXRect aRect;
    [Application new];
    NXSetRect(&aRect, 100.0, 300.0, 300.0, 300.0);
    myWin = [Window newContent: &aRect
                        style: NX_TITLEDSTYLE
                        backing: NX_BUFFERED
                        buttonMask: NX_MINIATURIZEBUTTONMASK
                        defer: NO];
    [myWin setTitle: "CompSoftDemo 5.0"];
    NXSetRect(&aRect, 0.0, 0.0, 300.0, 300.0);
    myView = [TestView newFrame: &aRect];
    [[myWin contentView] addSubview: myView];
    per1 = [Person newAtX: 100.0 Y: 100.0 Height: 100.0];
    [myView addOne: per1];
    cat2 = [Cat newAtX: 200.0 Y: 100.0 Height: 40.0];
    [myView addOne: cat2];
    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 動かす!

以上で「きれいに整理する」のが終わったので、最後にもう1つだけ、またまたスライダーを復活させて動かしてみましょう。そのための追加クラスを含んだ階層図です。

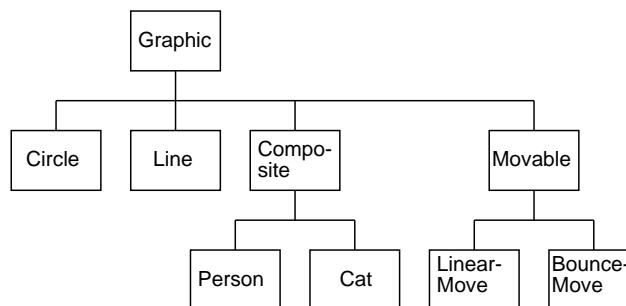


図 2: 動く機能を追加したクラス階層

では追加されたクラスを見て頂きます。

```

Movable.h -----
#import "Graphic.h"

@interface Movable : Graphic {
    id part;
    float time;
}
+ initWith: p;
- setTime: (float)t;
@end
Movable.m -----
#import "Movable.h"

@implementation Movable
+ initWith: p {
    self = [super new];
    part = p;
    return self;
}

```

```

- setTime: (float) t {
    time = t;
    if([part isKindOfClass: [Movable class]])
        [part setTime: t];
}
@end

LinearMove.h -----
#import "Movable.h"

@interface LinearMove : Movable {
    float vx, vy;
}
+ newWith: p Vx: (float)x Vy: (float)y;
@end

LinearMove.m -----
#import "LinearMove.h"
#import <dpsclient/psops.h>

@implementation LinearMove
+ newWith: p Vx: (float)px Vy: (float)py {
    self = [super newWith: p];
    vx = px; vy = py;
    return self;
}
- drawIt {
    PSgsave();
    PStranslate(vx*time, vy*time);
    [part drawIt];
    PSgrestore();
}
@end

BounceMove.h -----
#import "Movable.h"

@interface BounceMove : Movable {
    float length, freq;
}
+ newWith: p Len: (float)l Freq: (float)f;
@end

BounceMove.m -----
#import "BounceMove.h"
#import <dpsclient/psops.h>
#import <math.h>

@implementation BounceMove
+ newWith: p Len: (float)l Freq: (float)f {
    self = [super newWith: p];
    length = l; freq = f;
    return self;
}
- drawIt {
    float d = length * sin(2.0*3.1416*freq*time);
    if(d < 0) d = -d;
    PSgsave();
    PStranslate(0, d);
    [part drawIt];
    PSgrestore();
}
@end

```

さて、これらを使った TestView.[hm] と main.m です。

```
main.m -----
#import <appkit/appkit.h>
#import "TestView.h"
#import "LinearMove.h"
#import "BounceMove.h"
#import "Person.h"
#import "Cat.h"

main() {
    id myWin, myPanel, myMenu, myView;
    id per1, cat2, sli1;
    NXRect aRect;
    [Application new];
    NXSetRect(&aRect, 100.0, 300.0, 300.0, 350.0);
    myWin = [Window newContent: &aRect
              style: NX_TITLEDSTYLE
              backing: NX_BUFFERED
              buttonMask: NX_MINIATURIZEBUTTONMASK
              defer: NO];
    [myWin setTitle: "CompSoftDemo 4.2"];
    NXSetRect(&aRect, 0.0, 50.0, 300.0, 300.0);
    myView = [TestView newFrame: &aRect];
    [[myWin contentView] addSubview: myView];
    per1 = [Person newAtX: 100.0 Y: 100.0 Height: 100.0];
    per1 = [LinearMove newWith: per1 Vx: 1.0 Vy: 0.0];
    [myView addOne: per1];
    cat2 = [Cat newAtX: 200.0 Y: 100.0 Height: 40.0];
    cat2 = [BounceMove newWith: cat2 Len: 50.0 Freq: 0.02];
    [myView addOne: cat2];
    NXSetRect(&aRect, 10.0, 10.0, 280.0, 30.0);
    sli1 = [[Slider new] initWithFrame: &aRect];
    [sli1 setMinValue: 0.0];
    [sli1 setMaxValue: 200.0];
    [sli1 setFloatValue: 0.0];
    [[sli1 cell] setTarget: myView];
    [[sli1 cell] setAction: @selector(setTimeAndDraw:)];
    [sli1 sizeToFit];
    [[myWin contentView] addSubview: sli1];
    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];
    }
TestView.h -----
#import <appkit/View.h>
#import "Graphic.h"

@interface TestView : View {
    id objs[100];
    int count;
}
+ newFrame: (const NXRect*)frameRect;
- drawSelf: (const NXRect*)rects : (int)rectCount;
- addOne: (id)anObj;
- setTimeAndDraw: (id)aControl;
@end
TestView.m -----
#import "TestView.h"
#import "Movable.h"
#import <dpsclient/psops.h>

@implementation TestView
+ newFrame: (const NXRect*)frameRect {
    self = [super newFrame: frameRect];
    count = 0;
    return self;
}
- addOne: (id)anObj {
    objs[count++] = anObj;
}
- drawSelf: (const NXRect*)rects :(int)rectCount {
    int i;
    NXEraseRect(&bounds);
    for(i = 0; i < count; ++i)
        [objs[i] drawIt];
}
- setTimeAndDraw: aControl {
    int i;
    for(i = 0; i < count; ++i)
        if([objs[i] isKindOfClass: [Movable class]])
            [objs[i] setTime: [aControl floatValue]];
    [self display];
}
@end

```

さて、練習問題です。

練習 人が手や足を振りあげながらびよんびよんし、猫も尻尾を振りながらびよんびよんするように直してみよ。

これができれば、梶田さんのリクエストは完成したことになるわけでしょう？ これ以上やってしまうと皆様のやる事がなくなるから、デモプログラムの拡張は一応これまでとします。めでたしめでたし。