완성된 LIST

제 강좌를 한번이라도 보시고 기다리시는 분들에게 너무 죄송합니다. 솔직히 기다리시는 분들이 있을지가 더 의문입니다만.. 한동안 무지 바쁜지라..강좌가 머고 되는게 하나도 없었습니다. 솔직히 마음대로 되는거 하나 없습니다. 그나마 이번에 리스트를 완성시켰다는데..감동입니다. 완벽한 버전으로 나마 보답해 드리기 위해서 좋은 기능들을 많이 넣었습니다. ㅋㅋㅋ 아시는 분들은 다 아시는 기능들이지만요. 그나마 무리를 해서라도 소스를 공개합니다. 요즘들어 위피가 나오는 바람에 모바일 프로그래머만 작살입니다. 머이리 플랫폼이 많은지 참... 제대로 하는거 하나 없는데 배울 것은 산더미고..

제가 그냥 생각해 보건데..많은 분들이 네트워크 사용방법이라든지 혹은 컴포넌트 만드는 방법에 대해서 부족함을 많이들 느끼시는 것 같습니다. 이러한 부분들을 제가 조금이나마 도와드리기 위해서 실전에서 쓰이는 여러가지 기술들을 공유하는 하기로 맘 먹고 있습니다. 언제나 말씀드리지만 실전은 UI가 80%는 먹고 들어갑니다. 이런 UI들을 어떻게 프로그램으로 구현하느냐가 바로 노하우가 되는 것이지요..그래서 가끔씩 초보분들이 많이 어려워하시는 것 같습니다. 하지만..초보분들은 조금만 더 노력하시면 제 강의가 많이 도움이 되리라 생각됩니다. 집에 가서 조금씩 밤마다 돌리면서 소스를 완성했습니다. mm;

역시 처음은 저의 한을 푸는 문장으로 시작하는 군요..기분 와따 입니다..ㅋㅋㅋ

PinkMidlet.java

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;

import com.xce.lcdui.*;

public class PinkMidlet extends MIDlet
{
/******************************************************************************/
// Display instance
        private Display display;                                        // Display 변수
        private PinkCanvas pinkCanvas;

/******************************************************************************/
        public void startApp()
        {
```

```
                display = Display.getDisplay(this);
                pinkCanvas = new PinkCanvas();
                display.setCurrent(pinkCanvas);
        }

        public void pauseApp() {}
        public void resumeApp()         {}
        public void destroyApp(boolean con){}
}
```

설명 안드려도 미들릿은 전부 이해하시겠지요..무슨 미들릿이 이렇게 형편없냐고 하지만..미들릿을 복잡하게 할 의무도 없기 때문이죠..보기도 좋고 정리도되고…
Canvas는 제가 예전에 말슴드렸던..그러한 방식입니다. 현재는 리스트 하나밖에 없기 때문에 상태를 두어서 페인트를 하지는 않았습니다. 그점은 참고 하시기 바랍니다.

PinkCanvas.java

```
import javax.microedition.lcdui.*;
import com.xce.lcdui.*;
import com.skt.m.*;

public class PinkCanvas extends Canvas
{
        PinkThread              pinkThread; // 스레드
        PinkList    pinkList;       //    리스트
        PinkTicker              pinkTicker;     // 티커

        static int FONT_HEIGHT;     // 폰트 높이
        static int LCD_HEIGHT;        // LCD 넓이
        static int LCD_WIDTH;         // LCD 높이

        String[] pinkLecture = {"첫번째 강좌일까요. 고기를 낚으십당", "두번째 강좌", "세번째 강좌", "네번째 강좌", "다섯번째 강좌", "여섯번째 강좌", "일곱변째 강좌", "여덟번째 강좌", "아홉번째 강좌", "열번째 강좌"};

        PinkCanvas()
        {
```

```
                FONT_HEIGHT = Font.getDefaultFont().getHeight(); // 폰트의 Height
                LCD_HEIGHT = this.getHeight() + 16;     // 밑의 커맨드 영역도 포함
                LCD_WIDTH = this.getWidth();

                pinkThread = new PinkThread(this);
                pinkList = new PinkList(this);
                pinkTicker = new PinkTicker(this);

                pinkList.setStrings(pinkLecture, pinkLecture.length);     // 리스트의 내용을 삽입
                pinkList.setBounds(10, 10, 90, 100);       // 리스트 영역 크기 잡기
        }
        // 싱크를 걸어줘야지..페인트가 잘맞습니다.
        public synchronized void paint(Graphics g)
        {
                g.setClip(0, 0, LCD_WIDTH, LCD_HEIGHT);
                // 이 방법이 애니매이션의 키 포인트 입니다.
                if(pinkThread.getAni())              // 애니매이션일 경우
                {
                        pinkTicker.paint(g);          // 티커를 그려줌
                        return;
                }

                g.setColor(0xffffff);                        // 애니매이션이 아닐경우
                g.fillRect(0, 0, LCD_WIDTH, LCD_HEIGHT);
                pinkList.paint(g, true);                      // 리스트를 그림
                pinkThread.setAni(true);                      // 애니매이션을 TRUE로
        }

        public void keyPressed(int keyCode)                          // 모든 키 이벤트는 리스트로
        {
                pinkList.keyPressed(keyCode);
        }

        public void keyRepeated(int keyCode){}
        public void keyReleased(int keyCode){}
}
```

Canvas도 앞에서 강좌했던 부분과 크게 다를바 없습니다. 최대한 앞의 강좌의 맥을 이어가기 위해서 노력했씁니다.

Thread는 애니가 있을 경우에만 쓰입니다. 보통 애니가 안들어가는 경우는 없죠. 아무리 단순한 컨텐츠라고 해도 기본적으로 하나의 애니 정도는 들어갑니다.

PinkThread.java

```
import javax.microedition.lcdui.*;

import com.xce.lcdui.*;
import com.skt.m.*;

public class PinkThread extends Thread
{
	private boolean isAni;	// 애니매이션인지 아닌지
	private boolean isStop;	// 멈추기

	PinkCanvas pinkCanvas;
/*************************************************************************/
	PinkThread(PinkCanvas _pinkCanvas)
	{
		pinkCanvas = _pinkCanvas;
		this.start();
	}

/*************************************************************************/
	public void run()
	{
		while(!isStop)
		{
			if(isAni)	// 애니매이션 처리할 경우에 repaint()를 호출한다.
				pinkCanvas.repaint();
			// 만일 repaint속도를 빠르게 하고 싶다면
			// serviceRepaints()를 사용하여 속도를 빠르게 해주어도 됨
			try{	Thread.sleep(200); }catch(Exception e){}	// 슬립타임을 줌
		}
```

```
		}

/*************************************************************************/
		public void setStop(boolean _isStop)
		{
			isStop = _isStop;
		}

		public void setAni(boolean _isAni)
		{
			isAni = _isAni;
		}

		public boolean getAni()
		{
			return isAni;
		}
}
```

애니도 예상외로 간단합니다. 간단히 Thread만 돌리면 되거든요..제가 이렇게 모듈을 분리하는 이유는 나중에 너무 많은 파일이나 용량이 커지면 도대체 관리가 안됩니다. 그래서 최대한 한 모듈을 파일로 구분해서 처리할려고 하고 있습니다. 일단 객체 지향을 목적으로 하는 것은 아닙니다. 모바일에서 객체 지향 프로그래밍은 그다지 효율적이지 못하다고 생각합니다. 단지 소스를 수정하기 편하게 하기 위해서 모듈을 나눌 뿐입니다. 재활용의 목적은 그렇게 크지 않습니다. 제가 어떤 컨테츠 소스를 본적이 있는데..완전히 PC용 자바프로그램을 했더군요..상속에 interface에 작살이더군요...파일도 많고 용량도 크고..모바일 프로그램은 작고 간단하면서 해당 기능을 전부 가지고 있어야 합니다.

가장 긴 소스인 List입니다. 예전에 정말 간단한 List를 보여드렸습니다. 아주 기능도 거의 없는 것이지요…이번에는 이곳에 점 투자를 했습니다. 숫자키 이벤트도 넣고 Ticker도 넣고 하는 만큼은 했습니다. 솔직히 더 최적화 할 수 있지만 이정도면 다음은 고기를 직접 잡으시는 것도 좋을 듯 합니다. 현재 이 리스트는 각각의 리스트 간격도 조절합니다. 폰의 폰트와 LCD사이즈가 다르기 때문에 리스트 간격도 조금씩 수정되어야지 점 이쁜 리스트가 나오기 때문에 리스트 간격도 조절하게 되어있습니다.

PinkList.java

```
import javax.microedition.lcdui.*;
import java.util.*;

import com.xce.lcdui.*;
import com.skt.m.*;

public class PinkList
{
/*************************************************************************/
// 폰트 크기에 의한 줄간격
        public int FONT_HEIGHT;
// 숫자 딜레이
        private static final long DELAY = 500;

        int x;
        int x1;
        int y;
        int width;
        int height;

        int indexFirst;                                 // 현재 화면에 보이는 상단 리스트
        int indexSelect;                                // 현재 선택된 리스트
        int listScreenNum;                      // 현재 height에 보이는 최대 개수

        int list_num;                                           //리스트 개수

        char[][] list;                                          // 리스트 목록
        char[] listNum = {'①', '②', '③', '④', '⑤', '⑥', '⑦', '⑧', '⑨', '⑩', '⑪', '⑫', '⑬', '⑭', '⑮'};

        PinkCanvas pinkCanvas;

        private Timer timer = new Timer();
        private TimerTask numTimerClient = new NumTimerClient();
        int oldPressedKey = 0;                  // 오래 누르고 있을 때의 처리
        boolean isReleased;
```

```
/*****************************************************************************/
//
	public PinkList(PinkCanvas _pinkCanvas)
	{
		pinkCanvas = _pinkCanvas;
	}

/*****************************************************************************/
// 숫자 키로 리스트를 받기 위한 TimerTask
	class NumTimerClient extends TimerTask
	{
		public final void run()
		{
			isReleased = true;

			if(oldPressedKey > 0 && oldPressedKey <= list.length)
			{
				setIndex(oldPressedKey - 1);
				pinkCanvas.keyPressed(131);
			}

			oldPressedKey = 0;
			isReleased = false;
			numTimerClient.cancel();
		}
	}

/*****************************************************************************/
// 리스트의 크기
	public void setBounds(int _x, int _y, int _width, int _height)
	{
		x = _x + 1;
		x1 = _x - 12;
		y = _y;
		width = _width;
		height = _height;
```

```
                    // 현재 height에 보이는 최대 개수
                    listScreenNum = height / (PinkCanvas.FONT_HEIGHT + 1);
                    FONT_HEIGHT = height / listScreenNum; // 리스트 줄 간격 조절
            }

// String값으로 리스트 입력
            public void setStrings(String[] _list, int _totListNum)
            {
                    indexSelect = 0;
                    indexFirst = 0;
                    list = null;
                    list_num = _totListNum;

                    list = new char[_totListNum][];

                    for(int i = 0; i < _totListNum; i++)
                            list[i] = _list[i].toCharArray();
            }



// Char 배열로 리스트값 입력
            public void setChars(char[][] _list, int _totListNum)
            {
                    indexSelect = 0;
                    indexFirst = 0;

                    list_num = _totListNum;

                    list = null;
                    list = _list;
            }
/*************************************************************************/
// 선택된 리스트 인덱스 반환
            public int getSelectIndex()
            {
```

```
                return indexSelect;
        }
/*************************************************************************/
// 해당 인덱스로 리스트 선택하기
        public void setIndex(int _indexSelect)
        {
                indexSelect =_indexSelect;
        }

/*************************************************************************/
//
        public void paint(Graphics g, boolean isScroll)
        {
                int indexLast = (list_num > listScreenNum)?indexFirst + listScreenNum:list_num;

                for(int i = indexFirst , k = 0; i < indexLast; i++, k++)
                {       // 현재 선택된 리스트는 Ticker처리한다.
                        if(i == indexFirst)
                        {
                                // Ticker
                                pinkCanvas.pinkTicker.setBounds(x, y + FONT_HEIGHT
                                 * (indexSelect -indexFirst), width - 4, PinkCanvas.FONT_HEIGHT);
                                pinkCanvas.pinkTicker.setChars(list[indexSelect]);
                        }
                        g.setColor(0x000000);

                        g.setClip(x1, y, width - 4, height);
                        g.drawChar(listNum[i], x1, y + FONT_HEIGHT * k , g.TOP|g.LEFT);

                        g.setClip(x, y, width - 5, height);
                        g.drawChars(list[i] , 0, list[i].length, x, y + FONT_HEIGHT * k , g.TOP|g.LEFT);
                }

                g.setClip(0, 0, PinkCanvas.LCD_WIDTH, PinkCanvas.LCD_HEIGHT);
                // 우측의 스크롤 유무
                if(isScroll)
```

```
                                drawScrollBar(g, x + width - 3);
        }

/*************************************************************************/
// 오른쪽 스코롤 그리기
        public void drawScrollBar(Graphics g, int _x)
        {
                g.setColor(171, 181, 127);
                g.drawRect(_x, y + 1, 5, height - 2);

                g.setColor(230, 235, 209);
                g.drawRect(_x + 1, y + 1 + 1, 3, height - 4);

                g.setColor(203, 211, 171);
                g.drawRect(_x + 2, y + 1 + 2, 1, height - 6);

                int _y = y + 2 + getScrollBarLoc();

                g.setColor(152, 163, 14);
                g.drawLine(_x + 1, _y, _x + 1, _y + 4);
                g.drawLine(_x + 1, _y, _x + 4, _y);
                g.setColor(161, 194, 0);
                g.fillRect(_x + 2, _y + 1, 2, 3);
                g.setColor(117, 126, 14);
                g.drawLine(_x + 4, _y + 1, _x + 4, _y + 3);
                g.drawLine(_x + 1, _y + 4, _x + 4, _y + 4);
        }
// 스크롤 바 위치 얻어오기
        private int getScrollBarLoc()
        {
                return (list_num <= 1)?0:(height - 8) * indexSelect    / (list_num - 1);
        }

/*************************************************************************/
// 키인벤트
        public void keyPressed(int keyCode)
```

```
{
        switch(keyCode)
        {
                case Canvas.KEY_UP:             // 위
                        pinkCanvas.pinkThread.setAni(false);

                        if(indexSelect <= 0)
                        {
                                indexSelect = list_num - 1;
                                indexFirst = (list_num <= listScreenNum)?0:
                                        list_num - listScreenNum;
                        } else
                        {
                                if(indexFirst > indexSelect - 1)
                                        indexFirst--;

                                indexSelect--;
                        }

                        pinkCanvas.repaint();
                        break;
                case Canvas.KEY_DOWN:       // 아래
                        pinkCanvas.pinkThread.setAni(false);

                        if(indexSelect >= list_num - 1)
                                indexFirst = indexSelect = 0;
                        else
                        {
                                if(indexFirst + listScreenNum - 1 <= indexSelect)
                                        indexFirst++;

                                indexSelect++;
                        }
                        pinkCanvas.repaint();
                        break;
        }
```

```
        }
// 숫자키로 리스트를 선택할 경우
        public void keyReleased(int keyCode)
        {
                switch (keyCode)
                {
                        case Canvas.KEY_NUM0:
                        case Canvas.KEY_NUM1:
                        case Canvas.KEY_NUM2:
                        case Canvas.KEY_NUM3:
                        case Canvas.KEY_NUM4:
                        case Canvas.KEY_NUM5:
                        case Canvas.KEY_NUM6:
                        case Canvas.KEY_NUM7:
                        case Canvas.KEY_NUM8:
                        case Canvas.KEY_NUM9:
                                if(!isReleased)
                                {
                                        oldPressedKey = keyCode - Canvas.KEY_NUM0;
                                        numTimerClient.cancel();
                                        timer.schedule(numTimerClient, DELAY);
                                        isReleased = true;
                                        break;
                                }

                                oldPressedKey = oldPressedKey * 10 +
                                                (keyCode - Canvas.KEY_NUM0);
                                break;
                }
        }
}
```

소스 보시고 궁금한 점도 무지 많을까요? 많은 부분을 추가 시켜서 혼란스러운 분들도 계실 것 같습니다. 여하튼 궁금한거 있으시면 언제든지 질문해 주십시오..참고로 해당 소스는 제가 컨텐츠를 만들때쓰는 리스트 입니다 솔직히 잘하는 코딩도 아니지만 많은 도움이 되었으면 합니다. 언제든지 맘에 안드시는 부분은 태클 들어 오십시오. 몸으로 받게습니다.

이번 Ticker소스는 예전에 배포한것과 동일합니다. 고친 부분도 거의 없죠. 그냥 리스트가 선택되면 해당 리스트 목록을 티커한테 넘기고 쓰레드가 열라 돌면서 repaint()를 걸어줍니다. Repaint()에 영역을 설정하면 빠르기도 하겠죠..또한 setClip()이 paint()문장이 끝날때까지 영역이 잡혀 있다면 repaint()시에는 해당 영역만 repaint()합니다. Repaint()의 영역을 설정한것과 똑 같은 효과이지요.

PinkTicker.java

```
import javax.microedition.lcdui.*;
import com.xce.lcdui.*;

public class PinkTicker
{
	private final static int DELAY_TURN		= 1;		// delay turn
	private final static int MOVESTRING_WIDTH	= 10;

	private char[] text;			// 문자

	private int x;				// X 축
	private int y;				// Y 축
	private int width;			// 넓이
	private int height;			// 높이

	private int indexLeft;			// 첫번째 문자 움직이기
	private int indexRight;			// 두번째 문자 움직이기

	private int initDelay;
	private int delay;
	private int widthString;

	private boolean isPause;
	private boolean isStop;

	private int colorBack;

	private PinkCanvas pinkCanvas;
/*************************************************************************/
```

```
//
	PinkTicker(PinkCanvas _pinkCanvas)
	{
		pinkCanvas = _pinkCanvas;

		isPause = true;
		isStop = false;
		this.setColor(0xFF4105);
	}

/*************************************************************************/
// 티커 문자 설정하기
	public void setString(String _text)
	{
		setChars(_text.toCharArray());
	}
//
	public void setChars(char[] _text)
	{
		System.out.println("_text : " + new String(_text));

		text = null;
		text = _text;

		indexLeft = 0;
		indexRight = 0;
		delay = 0;



		widthString = Font.getDefaultFont().charsWidth(text, 0, text.length) ;
	}

/*************************************************************************/
// 티커 영역 설정하기
	public void setBounds(int _x, int _y, int _width, int _height)
	{
```

```
                this.x = _x;
                this.y = _y;
                this.width = _width;
                this.height = _height;
        }

/*************************************************************************/
//
        public void setColor(int _colorBack)
        {
                colorBack = _colorBack;
        }

        public void paint(Graphics g)
        {
                g.setClip(x, y, width, height);

                // Color 상자
                g.setColor(colorBack);
                g.fillRect(x, y, width, height);

                if(widthString >= width)
                {
                        if(delay++ > DELAY_TURN)
                        {
                                if(widthString+ initDelay >= indexLeft)
                                        indexLeft += MOVESTRING_WIDTH;

                                if(widthString + initDelay >= indexRight)
                                        indexRight += MOVESTRING_WIDTH;

                                if(widthString + initDelay < indexLeft)
                                {
                                        indexLeft = indexRight - widthString - 20;
                                        delay = 0;
                                }
```

```
                        if(widthString + initDelay < indexRight)
                                indexRight = indexLeft - widthString - 20;
                } else
                {
                        initDelay = indexLeft = -x;

                        indexRight = indexLeft - widthString - 20;
                }

                // 문자 그리기
                g.setColor(0xffffff);
                g.drawChars(text, 0, text.length, -indexLeft, y +
                                (height - PinkCanvas.FONT_HEIGHT) /2, g.LEFT|g.TOP);
                g.drawChars(text, 0, text.length, -indexRight, y +
                                (height - PinkCanvas.FONT_HEIGHT) /2, g.LEFT|g.TOP);
        }
        else
        {
                g.setColor(0xffffff);
                g.drawChars(text, 0, text.length, x, y, g.LEFT|g.TOP);
        }
    }
}
```

이번에는 열라 긴 소스를 넣어서 그런지 뿌듯하게 페이지가 왕창입니다. 다음에는 다른 컴포넌트들에 대해서 알려드리도록 하겠습니다. 언제나 말씀드리지만 디자이너가 생각한 모든 것을 프로그램으로 구현할 수 있어야지만이 진정한 모바일 프로그래머가 될 수 있지 않을까 생각됩니다. 모바일은 UI 80%가 먹고 들어갑니다. 디자이너의 상상을 손바닥위로 올리십시오. 그리고 네트워크나 각종 최적화 기법들은 당근 몸에 익숙해야 합니다. 거진 2달만에 올리는 강좌라서 할말도 많고 잡다구리는 열라 많습니다. 이번 강좌로 점 열분들에게 깍인 점수를 만해 할수 있는 기회가 될는지..ㅋㅋㅋ

닉네임 : 핑크레드
성격 : 왕소심, 왕삐짐, 이중인격, 수다쟁이

이메일 : pinkred@hanafos.com

추신 : 강좌를 다른 사이트에 올리시는 것은 자유입니다..단지 저한테 메일 한 통 이라도 보내주시고 올리시면 감사하겠습니다. 언제나 좋은 하루 되십시오.