```
//pageflip version 2.13 (200504272050)
//
//0427: - removeML() function added, see details at frame 15
//      - new page definition system, hope it is easier to use!
//      - no need to adjust the mask size!
//0418: now it is AS2/FLash7 Compatible (The bug was the variable flipOff/flipoff)
//skip page option added to gotoPage function
//(for the latest version, always check at http://www.iparigrafika.hu/pageflip)
//
//this code is written by Macc 2004.11.01
//abel@iparigrafika.hu
//
//this code is opensource
//
//free use!
//but a small credit would be nice:)



//these values can be changed: ------------------------------------------------------------------
-----

pw = 390;           //page width in pixel
ph = 490;           //page height in pixel
                    //pages in the library must have the same size, the script will not resize them

//page data... (export names/tearing flag)
addPage("start");       //Always start with "start"!!!
addPage("page1"); // counts as pg 1
addPage("page2"); // counts as pg 2
addPage("page3"); // counts as pg 3
addPage("page6"); // counts as pg 4
addPage("page7"); // counts as pg 5
addPage("page8"); // counts as pg 6
addPage("page9"); // counts as pg 7
addPage("page10"); // counts as pg 8
addPage("page11"); // counts as pg 9
addPage("page12"); // counts as pg 10
addPage("page13"); // counts as pg 11
addPage("page16"); // counts as pg 12
addPage("page17"); // counts as pg 13
addPage("page4"); // counts as pg 14
addPage("page5"); // counts as pg 15
addPage("page18"); // counts as pg 16
addPage("page19"); // counts as pg 17
addPage("page20"); // counts as pg 18
addPage("page21"); // counts as pg 19
addPage("page22"); // counts as pg 20
addPage("page23"); // counts as pg 21
addPage("page24"); // counts as pg 22
addPage("end");         //always end with "end"!!!
/* IMPORTANT ABOUT BUTTONS WORKING . . .
IF YOU MOVE OR REMOVE PAGES BUTTONS WILL BREAK.
YOU NEED TO ALSO EDIT/UPDATED THE CODE TO ENABLE BUTTONS WHEN THE PAGE IS SHOWING
FIND IT ON THE FIRST FRAME OF EACH PAGE FROM THE LIBRARY.
*/

page=0;             //first page (normally it is 0 = the page before the cover = blank page)
```

```
hcover = true;      //hard cover on/off

clickarea = 60;     //pixel width of the click sensitive area at the edges.. 64
afa = 60;           //width of the autoflip starter square. 56

gs = 2;             //goto page flip speed
ps = 2;             //mouse pursuit speed
es = 3;             //flip speed after mouse btn release

canflip = true;     //page flipping enabled

transparency = false;   //use transparent pages or not (1 level transparency)

lcover = false;     //large cover on/off
lcaddx = 0;     //width difference
lcaddy = 0;     //height difference on top/bottom

SoundOn = true;     //use page sounds


//do not change these values:  ------------------------------------------------------------------
------------------------------------------------

_global.mcnt = 0;   //counter (used on a page where is an animation)
gpage = 0;          //gotoPage No
gflip = false;      //gotoPage flip
gdir = 0;           //goto direction...
gskip = false;      //skip pages    ***
gtarget = 0;        //target when skipping

aflip = false;      //auto flip
flip = false;       //pageflip
flipOff = false;    //terminateflip
flipOK = false;     //good flip
hflip = false;      //hardflip (the cover of the book)
rotz = -30;         //hardflip max y difference

preflip = false;    //corner flip status
ctear = false;      //actual page status
tear = false;
teard = 0;
tlimit = 80;
removedPages = new Array(); //list of removed pages!

mpx = 0, mpy = 0;   //mousepos at click
sx = sy = 0;        //startpoint when flipping
x = 0;  //mouse x,y
y = 0;
ax = 0; //auto x,y
ay = 0;
acnt = 0;
aadd = 0;
aamp = 0;

AM = Math.PI/180;

//pageflip code by Macc -------------------------------------------------------------------------
------------------------------------------------
```

```
function addPage(ename,tear) {      //defining pages... ----------------------------------------
----------------------------------------
    if(ename=="start") {
        pageOrder = new Array();
        pageCanTear = new Array();
        page=0;
        ename="blankpage";
    }
    if(ename=="end") {
        maxpage = page-1;
        ename="blankpage";
    }
    tear = tear==undefined? false: tear;
    pageOrder[page] = ename;
    pageCanTear[page] = tear;
    page++
}

function reset() {                  //initializing pages... ------------------------------------
----------------------------------
    pages.p4.page._x = -pw;
    pages.p4._x = pw;
    pages.p1.page._x = -pw;
    pages.p1._x = 0;

    pages.flip.p2.page._x = -pw;
    pages.flip.p2._x = pw;
    pages.flip.p3.page._x = -pw;
    pages.flip.p3._x = 0;

    pages.p0.page._x = -pw;
    pages.p0._x = 0;
    pages.p5.page._x = -pw;
    pages.p5._x = pw;

    pages.pLL.page._x = -pw;
    pages.pLL._x = 0;
    pages.pLR.page._x = -pw;
    pages.pLR._x = pw;


    pages.pgrad._visible = pages.mask._visible = pages.flip._visible = false;
    pages.flip.p3mask._width = pages.pgmask._width = pw*2;
    pages.flip.p3mask._height = pages.pgmask._height = ph;
    pages.center._height = ph+2*lcaddy;
    pages.flip.fmask.page.pf._width = pw;
    pages.center._width = 6;
    pages.flip.fmask.page.pf._height = ph;

    pages.mask._height = pages.pgrad._height = pages.flip.p3shadow._height = pages.flip.flipgrad._h
ht = 2*Math.sqrt(ph*ph+pw*pw);

    pageNumber = new Array();
    for(i=0;i<=(maxpage+1);i++) pageNumber[i] = i;
}

mousecontroll = new Object();       //initializing mouse click handler --------------------------
----------------------------------------
```

```
mousecontroll.onMouseDown = function() {
    if(flip && !aflip) {//  && !preflip) {
        flipOK = false;
        if(sx<0 && pages._xmouse>0) flipOK = true;
        if(sx>0 && pages._xmouse<0) flipOK = true;
        flipOff = true;
        flip = false;
    } else if((flipOff || aflip || !canflip) && !preflip) {
        trace("donothing");
    } else {
        var oox = ox;
        var ooy = oy;
        var osx = sx;
        var osy = sy;
        var hit = hittest();        //hittest
        if(hit) {
            startsnd(1);        //Sound
            anim._visible = false;
            flip = true;
            flipOff = false;
            tear = false;           //not tearing yet...
            ox = sx = hit*pw;
            if(preflip) {
                aflip = preflip = false;
                ox = oox, oy = ooy;
                sx = osx, sy = osy;
            }
            pages.flip.setMask(pages.mask);
            mpx = pages._xmouse, mpy = pages._ymouse;
            oef();
            //_quality = "MEDIUM";  //it is the place to degrade image quality while turning pages
formance is too low.
        }
    }
}
mousecontroll.onMouseUp = function() {
    if(flip && !tear) {
        if((Math.abs(pages._xmouse)>(pw-afa) && Math.abs(pages._ymouse)>(ph/2-afa) && Math.abs(page
use-mpx)<afa) || preflip) {
            flip = false;
            preflip = false;
            autoflip();
            startsnd(2);    //sound
        } else if(!preflip) {
            preflip = false;
            flipOK = false;
            if(sx<0 && pages._xmouse>0) flipOK = true;
            if(sx>0 && pages._xmouse<0) flipOK = true;
            flipOff = true;
            flip = false;
            if(flipOK) startsnd(2); //sound
        }
    }
}

function hittest() {    //hittest at mouse clicks, if click is over the book -> determining turning
rection ----------------------------------
    var x=pages._xmouse;
```

```
    var y=pages._ymouse;
    var pmh = ph/2;

    if(y<=pmh && y>=-pmh && x<=pw && x>=-pw) {  //ha a megadott intervallumban klikkelunk, akkor la
atunk
        var r = Math.sqrt(x*x+y*y);
        var a = Math.asin(y/r);
        var y = Math.tan(a)*pw;
        if(y>0 && y>ph/2) y = ph/2;
        if(y<0 && y<-ph/2) y = - ph/2;
        oy = sy = y;
        r0 = Math.sqrt((sy+ph/2)*(sy+ph/2)+pw*pw);
        r1 = Math.sqrt((ph/2-sy)*(ph/2-sy)+pw*pw);

        pageN = eval("pages.flip.p2.page");
        pageO = eval("pages.flip.p3");
        offs = -pw;
        pages.flip.fmask._x = pw;

        if(x<-(pw-clickarea) && page>0) {       //>-----> flip backward
            pages.flip.p3._x = 0;
            hflip = checkCover(page,-1);
            setPages(page-2,page-1,page,page+1);
            ctear = pageCanTear[page-1];
            return -1;
        }
        if(x>(pw-clickarea) && page<maxpage) {  //<-----< flip forward
            pages.flip.p3._x = pw;
            hflip = checkCover(page,1);
            setPages(page,page+2,page+1,page+3);
            ctear = pageCanTear[page+2];
            return 1;
        }
    } else return 0;    //wrong click
}
function checkCover(p,dir) {
    if(hcover) {
        if(dir>0) {
            if(p==(maxpage-2) || p==0) return true;
        } else {
            if(p==maxpage || p==2) return true;
        }
    }
    return false;
}
function corner() {
    var x = Math.abs(pages._xmouse);
    var y = Math.abs(pages._ymouse);
    if(x>(pw-afa) && x<pw && y>(ph/2-afa) && y<(ph/2)) {
        return true;
    }
    return false;
}


function oef() {
    _global.mcnt++; //main counter incrase (need for some page effect);
```

```
    if(!flip && corner()) { //corner mouseover
        preflip = true;
        if(!autoflip()) preflip = false;
    }
    if(preflip && !corner()) {
        preflip = false;
        flip = false;
        flipOK = false;
        flipOff = true;
    }
    getm();
    if(aflip && !preflip) {
        y = (ay += (sy-ay)/(gflip? gs: ps ));
        acnt += aadd;
        ax -= aadd;
        if(Math.abs(acnt)>pw) {
            flipOK = true;
            flipOff = true;
            flip = false;
            aflip = false;
        }
    }
    if(flip) {                          //page turning is in progress...
        if(tear) {
            x = tox;
            y = (toy += teard);
            teard *= 1.2;
            if(Math.abs(teard)>1200) {
                flipOff = true;
                flip = false;
            }
        } else {
            x = (ox += (x-ox)/(gflip? gs: ps ));
            y = (oy += (y-oy)/(gflip? gs: ps ));
        }
        calc(x,y);                      //positioning pages and shadows
    }
    if(flipOff) {                       //terminating page turning effect... (comlplete turning...
side)
        if(flipOK || tear) {
            x = (ox += (-sx-ox)/(gflip? gs: es ));
            y = (oy += (sy-oy)/(gflip? gs: es ));
            calc(x,y);
            if(x/-sx > 0.99 || tear) {          //we are done with turning, so stop all turning iss
                flip = false;
                flipOK = flipOff = false;
                pages.pgrad._visible = pages.flip._visible = false;
                //_quality = "BEST";    //if quality is decrased during turning effect, you must re
value!

                if(tear) {              //if tear: remove page!!!
                    removePage((sx<0)? page: page+1);
                    page += (sx<0)? -2: 0;
                } else {
                    page += (sx<0)? -2: 2;  //and tourning pages at pagenumber level...
                }
                if(gskip) page = gtarget;
                setPages(page,0,0,page+1);
```

```
                tear = false;

                if(gpage>0 && !gskip) {         //gotoflip active -> is there another flipping left
                    gpage--;
                    autoflip();
                    startsnd(0);    //sound
                } else gflip = gskip = false;
            }
        } else {                        //terminating page turning effect... (incomlplete turning..
 side)
            x = (ox += (sx-ox)/3);
            y = (oy += (sy-oy)/3);
            calc(x,y);
            if(x/sx > 0.99) {           //we are done with turning, so stop all turning issue...
                flip = false;
                flipOff = false;
                aflip = false;
                pages.pgrad._visible = pages.flip._visible = false;
                //_quality = "HIGH";    //if quality is decrased during turning effect, you must re
t value!
                setPages(page,0,0,page+1);  //no change at pagenumbers..
            }
        }
    }
}

function calc(x,y) {                    //positioning pages and shadows by x,y reference points ---
-------------------------------
    if(hflip) { //hardflip...
        var xp = (sx<0)? -x: x;
        if(xp>0) {
            sp2._visible = false;
            sp3._visible = true;
            scalc(sp3,x);
        } else {
            sp3._visible = false;
            sp2._visible = true;
            scalc(sp2,x);
        }
        pages.flip.setMask(null);
        pages.flip._visible = true;
        pages.flip.fgrad._visible = false;
        pages.flip.p2._visible = pages.flip.p3._visible = false;
        return;
    } else pages.flip.fgrad._visible = true;

    //normal flipping process-----------------------------------------------------------------------
    var rr0 = Math.sqrt((y+ph/2)*(y+ph/2)+x*x);
    var rr1 = Math.sqrt((ph/2-y)*(ph/2-y)+x*x);
    if((rr0>r0 || rr1>r1) && !tear) {   // we can tear off pages now:)
                            // so reference points must be recalculated!
        if(y<sy) {  // k1-gyel kell osszehasonlitani!
            var a = Math.asin((ph/2-y)/rr1);
            y = (ph/2-Math.sin(a)*r1);
            x = (x<0)? -Math.cos(a)*r1: Math.cos(a)*r1;
            if(y>sy) {
                if((sx*x)>0) y = sy, x = sx;
                else y = sy, x = -sx;
```

```
        }
        if((rr1-r1)>tlimit && ctear) {
            teard = -5;
            tear = true;
            tox = ox = x;
            toy = oy = y;
        }
    } else {    // k0-val kell osszehasonlitani!
        var a = Math.asin((y+ph/2)/rr0);
        y = Math.sin(a)*r0-ph/2;
        x = (x<0)? -Math.cos(a)*r0: Math.cos(a)*r0;
        if(y<sy) {
            if((sx*x)>0) y = sy, x = sx;
            else y = sy, x = -sx;
        }
        if((rr0-r0)>tlimit && ctear) {
            teard = 5;
            tear = true;
            tox = ox = x;
            toy = oy = y;
        }
    }
}
if((sx<0 && (x-sx)<10) || (sx>0 && (sx-x)<10)) {
    if(sx<0) x = -pw+10;
    if(sx>0) x = pw-10;
}
//calculating flipping process
pages.flip._visible = true;
pages.flip.p3shadow._visible = pages.pgrad._visible = !tear;
pages.flip.p2._visible = pages.flip.p3._visible = true;
//equation of the line
var vx = x-sx;
var vy = y-sy;
var a1 = vy/vx;
var a2 = -vy/vx;

cx = sx+(vx/2);
cy = sy+(vy/2);
//trigonometriai szamitasok
//calculating rotation of the page, and the masks
var r = Math.sqrt((sx-x)*(sx-x)+(sy-y)*(sy-y));
var a = Math.asin((sy-y)/r);
if(sx<0) a = -a;
ad = a/AM;  //in degree
pageN._rotation = ad*2;
r = Math.sqrt((sx-x)*(sx-x)+(sy-y)*(sy-y));
rl = (pw*2);
if(sx>0) {              //flip forward
    pages.mask._xscale = 100;
    nx = cx-Math.tan(a)*(ph/2-cy);
    ny = ph/2;
    if(nx>pw) {
        nx = pw;
        ny = cy+Math.tan(Math.PI/2+a)*(pw-cx);
    }
    pageN.pf._x = -(pw-nx);
    pages.flip.fgrad._xscale = (r/rl/2)*pw;
```

```
        pages.pgrad._xscale = -(r/rl/2)*pw;
        pages.flip.p3shadow._xscale = (r/rl/2)*pw;
    } else {                //flip backward
        pages.mask._xscale = -100;
        nx = cx-Math.tan(a)*(ph/2-cy);
        ny = ph/2;
        if(nx<-pw) {
            nx = -pw;
            ny = cy+Math.tan(Math.PI/2+a)*(-pw-cx);
        }
        pageN.pf._x = -(pw-(pw+nx));
        pages.flip.fgrad._xscale = -(r/rl/2)*pw;
        pages.pgrad._xscale = (r/rl/2)*pw;
        pages.flip.p3shadow._xscale = -(r/rl/2)*pw;
    }
    pages.mask._x = cx;
    pages.mask._y = cy;
    pages.mask._rotation = ad;
    pageN.pf._y = -ny;
    pageN._x = nx+offs;
    pageN._y = ny;
    pages.flip.fgrad._x = cx;
    pages.flip.fgrad._y = cy;
    pages.flip.fgrad._rotation = ad;
    pages.flip.fgrad._alpha = (r>(rl-50))? 100-(r-(rl-50))*2: 100;
    pages.flip.p3shadow._x = cx;
    pages.flip.p3shadow._y = cy;
    pages.flip.p3shadow._rotation = ad;
    pages.flip.p3shadow._alpha = (r>(rl-50))? 100-(r-(rl-50))*2: 100;
    pages.pgrad._x = cx;
    pages.pgrad._y = cy;
    pages.pgrad._rotation = ad+180;
    pages.pgrad._alpha = (r>(rl-100))? 100-(r-(rl-100)): 100;
    pages.flip.fmask.page._x = pageN._x;
    pages.flip.fmask.page._y = pageN._y;
    pages.flip.fmask.page.pf._x = pageN.pf._x;
    pages.flip.fmask.page.pf._y = pageN.pf._y;
    pages.flip.fmask.page._rotation = pageN._rotation;
}

function scalc(obj,x) { //hardflip calc
    if(x<-pw) x=-pw;
    if(x>pw) x=pw;
    var a = Math.asin( x/pw );
    var rot = a/AM/2;
    var xs = 100;
    var ss = 100*Math.sin( rotz*AM );
    x = x/2;
    var y = Math.cos(a)*(pw/2)*(ss/100);
    placeImg(obj, rot, ss, x, y)
    pages.pgrad._visible = pages.flip._visible = true;
    pages.pgrad._xscale = x;
    pages.pgrad._alpha = pages.flip.p3shadow._alpha = 100;
    pages.flip.p3shadow._xscale = -x;
    pages.flip.p3shadow._x = 0;
    pages.flip.p3shadow._y = 0;
    pages.flip.p3shadow._rotation = 0;
    pages.pgrad._x = 0;
```

```
    pages.pgrad._y = 0;
    pages.pgrad._rotation = 0;
}

function placeImg(j, rot, ss, x, y) {
    var m = Math.tan( rot*AM );
    var f = Math.SQRT2/Math.sqrt(m*m+1);
    var phxs = 100*m;
    var phRot = -rot;;
    var xs = 100*f;
    var ys = 100*f;
    j.ph.pic._rotation = 45;
    j.ph.pic._xscale = (phxs<0)? - xs: xs;
    j.ph.pic._yscale = ys*(100/ss);
    j.ph._rotation = phRot;
    j.ph._xscale = phxs;
    j._yscale = ss;
    j._x = x;
    j._y = y;
    j._visible = true;
}

function setPages(p1,p2,p3,p4) {    //attach the right page "image" at the right place:)
    p0 = p1-2;                      //pages for transparency...
    p5 = p4+2;
    if(p0<0) p0=0;
    if(p5>maxpage) p5=0;

    if(p1<0) p1=0;  //visible pages
    if(p2<0) p2=0;
    if(p3<0) p3=0;
    if(p4<0) p4=0;
    trace("setpages ->"+p1+","+p2+","+p3+","+p4);

    pleft = pages.p1.page.pf.ph.attachMovie(pageOrder[p1],"pic",0);
    pages.p1.page.pf.ph._y = -ph/2;

    if(transparency) {
        pleftb = pages.p0.page.pf.ph.attachMovie(pageOrder[p0],"pic",0);
        pages.p0.page.pf.ph._y = -ph/2;
    } else pages.p0._visible = false;
    if(hflip) { //hardflip pages are specials!!!
        var tm = pages.flip.hfliph.attachMovie("sph","sp2",0);
        sp2 = tm.ph.pic.attachMovie(pageOrder[p2],"pic",0);
        sp2._y = -ph/2, sp2._x = -pw/2;
        sp2 = eval("pages.flip.hfliph.sp2");
        var tm = pages.flip.hfliph.attachMovie("sph","sp3",1);
        sp3 = tm.ph.pic.attachMovie(pageOrder[p3],"pic",0);
        sp3._y = -ph/2, sp3._x = -pw/2;
        sp3 = eval("pages.flip.hfliph.sp3");
    } else {
        pages.flip.hfliph.sp2.removeMovieClip();
        pages.flip.hfliph.sp3.removeMovieClip();
        sp2 = pages.flip.p2.page.pf.ph.attachMovie(pageOrder[p2],"pic",0);
        pages.flip.p2.page.pf.ph._y = -ph/2;
        sp3 = pages.flip.p3.page.pf.ph.attachMovie(pageOrder[p3],"pic",0);
        pages.flip.p3.page.pf.ph._y = -ph/2;
    }
```

```
    pright = pages.p4.page.pf.ph.attachMovie(pageOrder[p4],"pic",0);
    pages.p4.page.pf.ph._y = -ph/2;
    if(transparency) {
        prightb = pages.p5.page.pf.ph.attachMovie(pageOrder[p5],"pic",0);
        pages.p5.page.pf.ph._y = -ph/2;
    } else pages.p5._visible = false;
    if(lcover) {
        var lpl = transparency? p1-4: p1-2;
        var lpr = transparency? p4+4: p4+2;
        var limit = transparency? 0: -2;
        if(lpl>limit) {
            pages.pLL.page.pf.ph.attachMovie(pageOrder[2],"pic",0);
            pages.pLL.page.pf.ph._y = -ph/2;
            pages.pLL._visible = true;
        } else pages.pLL._visible = false;
        if(lpr<(maxpage-limit)) {
            pages.pLR.page.pf.ph.attachMovie(pageOrder[maxpage-1],"pic",0);
            pages.pLR.page.pf.ph._y = -ph/2;
            pages.pLR._visible = true;
        } else pages.pLR._visible = false;
    }
}

function resetPages() {
    setPages(page,0,0,page+1);
}

function autoflip() {                                   //start auto flip!
    if(!aflip && !flip && !flipOff && canflip) {    //only when all conditions fits our needs...
        acnt = 0
        var pmh = ph/2;
        aamp = Math.random()*pmh-(ph/4);
        var x= gflip? (gdir*pw)/2: ((pages._xmouse<0)? -pw/2: pw/2);

        var y= pages._ymouse;

        if(y>0 && y>pmh) y = pmh;
        if(y<0 && y<-pmh) y = - pmh;

        oy = sy = y;
        ax = (pages._xmouse<0)? -pmh: pmh;
        ay = y*Math.random();               //page turnig style randomizing

        offs = -pw;
        var hit = 0;
        if(x<0 && page>0) {
            pages.flip.p3._x = 0;
            hflip = (hcover && gskip)? (page==maxpage || gtarget==0): checkCover(page,-1);
            if(!(preflip && hflip)) {
                if(gskip) setPages(gtarget,gtarget+1,page,page+1);
                else setPages(page-2,page-1,page,page+1);
            }
            hit = -1;
        }
        if(x>0 && page<maxpage) {
            pages.flip.p3._x = pw;
            hflip = (hcover && gskip)? (page==0 || gtarget==maxpage): checkCover(page,1);
            if(!(preflip && hflip)) {
```

```
                if(gskip) setPages(page,gtarget,page+1,gtarget+1);
                else setPages(page,page+2,page+1,page+3);
            }
            hit = 1;
        }
        if(hflip && preflip) {
            hit = 0;
            preflip = false;
            return false;
        }
        if(hit) {
            anim._visible = false;
            flip = true;
            flipOff = false;
            ox = sx = hit*pw;
            pages.flip.setMask(pages.mask);
            aadd = hit*(pw/(gflip? 5:10 ));          //autoflip takes 10 frames to be done!!!
            aflip = true;
            pages.flip.fmask._x = pw;
            if(preflip) {
                oy = sy = (pages._ymouse<0)? -(ph/2): (ph/2);
            }
            r0 = Math.sqrt((sy+ph/2)*(sy+ph/2)+pw*pw);
            r1 = Math.sqrt((ph/2-sy)*(ph/2-sy)+pw*pw);
            pageN = eval("pages.flip.p2.page");
            pageO = eval("pages.flip.p3");
            oef();
            return true;
        }
    } else return false;
}

function getm() {   //get x,y reference points depending of turning style: manual/auto
    if(aflip && !preflip) {
        x = ax;
        y = ay;
    } else {
        x = pages._xmouse;
        y = pages._ymouse;
    }
}

function gotoPage(i,skip) {     //quickjump to the page number i
    i = getPN(i);               //i = target page
    gskip = (skip==undefined)? false: skip; //skip pages

    if(i<0) return false;
    var p = int(page/2);
    var d = int(i/2);
    if(p!=d && canflip && !gflip) {     //target!=current page
        if(p<d) {                       //go forward
            gdir = 1;
            gpage = d-p-1;
        } else {                        //go backward
            gdir = -1
            gpage = p-d-1;
        }
        gflip = true;
```

```
        if(gskip) gtarget = d*2, gpage = 0;
        autoflip();
        startsnd(0);    //sound
    } else gskip = false;
}
function getPN(i) { //get the right page number
    if(i==0) return 0;
    var find = false;
    for(j=1;j<=maxpage;j++) {
        if(i==pageNumber[j]) {
            i=j;
            find = true;
            break;
        }
    }
    if(find) return i;
    else return -1;
}
function removePage(i) {
    trace("remove page "+i);
    i = (Math.floor((i-1)/2)*2)+1;
    removedPages.push(pageNumber[i], pageNumber[i+1]);

    for(j=(i+2);j<=(maxpage+1);j++) {
        pageOrder[j-2]=pageOrder[j];
        pageCanTear[j-2]=pageCanTear[j];
        pageNumber[j-2]=pageNumber[j];
    }
    trace("removed pages "+i+","+(i+1));
    trace(removedPages.join(", "));
    maxpage -= 2;
}

//------------------------------------------------------------------------------

snd0 = new Sound();                 //adding sound objects
snd1 = new Sound();
snd2 = new Sound();
snd0.attachSound("pf1");
snd1.attachSound("pf2");
snd2.attachSound("pf3");


function startsnd(i) {              //Sound starter
    if(SoundOn) {
        if(i==0) {
            snd0.start(0,0);
            snd0.onSoundComplete = function () {
                startsnd(2);
                delete snd0.onSoundComplete;
            }
        } else {
            i--;
            this["snd"+i].start(0,0);
        }
    }
}
//------------------------------------------------------------------------
```

```
/*  ocean sounds

*/
function soundControll(command){
    if (command=="mute"){
        trace('mute sound');
        this.ocean0_mc.stop();
        this.ocean1_mc.stop();
    }
    if (command=="play"){
        //ocean0.start();
        trace('play sound');
        this.ocean0_mc.play();
        this.ocean1_mc.play();
    }
}



//-------------------------------------------------------------------------------------------------
PUT YOUR CODE HERE --------------


/*      you can use these functions:

        gotoPage( destinationPageNo, skip );    //quick jump to the page number: destinationPageNo;
 0-maxpages; skip: boolean; if true, pages will be skipped to the destination!

        canflip     //it's a variable. setting its value to false disables flipping

        other functions of page turning is automatic;

        WARNING!!!
        if you want to unload/reload tha pageflip, before unloading call function: removeML(); to r
mouse listener!
*/

function startAutoFlip () {
    intervalID = setInterval(nextPage,2000);    //2 seconds
}
function stopAutoFlip () {
    clearInterval(intervalID);
}
function prevPage() {
    gotoPage(page-2);
}
function nextPage() {
    gotoPage(page+2);
}
```