

Order of proceedings:

1. Make a diagram with all coords systems used.
2. Make sure that program conforms with coord standards.
3. Re-insert geometric gliding, allow choice of gliding method.
4. Introduce hyperbolic gliding.
5. Make program self-documenting.

```

// HandoutBrowser.js; see
// http://katlas.math.toronto.edu/drorbn/?title=HandoutBrowser.j
// (c) Copyleft Dror Bar-Natan, 2007-2008. See
// http://www.math.toronto.edu/~drorbn/Copyleft/index.htm
// Double buffering by Scott Morrison

var text_level = 2;
var alt_text_level = 1;
var rate = 30; // Glide refresh rate.
var ticks_per_glide = rate;
var tl = 0; // ticks left for current glide.
var winW = 1024, winH = 740.
{
var viewX0, viewX1, viewY0, viewY1
var fromX0, fromX1, fromY0, fromY1
var mx, my, nx, ny; // Handout coordinates to window pixels
// linear conversion coeffs.
{
var toX0, toX1, toY0, toY1; // Target of current glide.
var menu, menu_div;
var slide_0, slide_div_0, slide_1, slide_div_1
var slide_div_2;
var slide_order = 0;
var visible_slide, hidden_slide;
var visible_slide_div, hidden_slide_div
var slideW, slideH;
var help_div;
var info_span;
var mouse_start, mouse_now, slide_start
var k, n;
var mode = "normal";
var rectangle_div;
var maxX, maxY, step;

var browser = "Mozilla";
if (parseInt(navigator.appVersion) > 3 && navigator.appName.indexOf("Microsoft") != -1) browser = "IE";

var slide_set;
var image_set = [];
if (slide_set != undefined) {
  image_set.length = slide_set.length
  for (i=0; i < slide_set.length; ++i)
    image_set[i] = new Image()
    image_set[i].src = slide_set[i]
}
}

function loaded() {
  slide_0 = document.getElementById("slide_0")
  slide_1 = document.getElementById("slide_1")
  slideW = slide_0.width; slideH = slide_0.height
  if (isNaN(maxX)) maxX = slideW; if (isNaN(maxY)) maxY = slideH
  if (isNaN(step)) step = 1.
  slide_div_0 = document.getElementById("slide_div_0")
  slide_div_1 = document.getElementById("slide_div_1")
  set_slide_order(0);
  rectangle_div = document.getElementById("rectangle_div")
  help_div = document.getElementById("help_div")
  help_div.innerHTML = "\
  <span id=\"beginner_help\" style=\"background-color:yellow\">
  Best viewed at full screen (<font color=red>F11</font>)<br>
  <a href=\"javascript: keyaction('t');\"><font color=red>t</font>ext level<a><br>
  <a href=\"javascript: keyaction('n');\"><font color=red>n</font>ext</a> / <a href=\"javascript:
  <a href=\"javascript: keyaction('g');\"><font color=red>g</font>lide on/off</a><br>
  <a href=\"javascript: keyaction('h');\"><font color=red>h</font>andout only</a><r></span>
  <span id=\"intermediate_help\" style=\"display: none; background-color:yellow;\">move view <a href=
  <a href=\"javascript: keyaction('c');\"><font color=red>c</font>apture</a> / <a href=\"javascr
  <a href=\"javascript: keyaction('r');\"><font color=red>r</font>ename view</a>
  </span>\"
  n=views.length; k=0;
  menu_div = document.getElementById("menu_div")
  info_span = document.getElementById("info_span")
  set_menu();
  viewX0=views[k][0]; viewY0=views[k][1]
  viewX1=views[k][2]; viewY1=views[k][3]

```

1. Divide into "modules".
2. Check if there's a self-documenting

*Mods needed for
Center-scale
coords.

move
to
html

```

set_view();
set_textmode();
setInterval("tick()", 1000/rate)
}

function set_slide_order(order) {
  if (order == 0) {
    slide_div_0.style.display = 'inline'
    slide_div_0.style.zIndex = 0
    slide_div_1.style.zIndex = -1
    slide_div_1.style.display = 'none'
    visible_slide = slide_0
    hidden_slide = slide_1
    visible_slide_div = slide_div_0
    hidden_slide_div = slide_div_1
  } else {
    slide_div_1.style.display = 'inline'
    slide_div_0.style.zIndex = -1
    slide_div_1.style.zIndex = 0;
    slide_div_0.style.display = 'none'
    visible_slide = slide_1
    hidden_slide = slide_0
    visible_slide_div = slide_div_1
    hidden_slide_div = slide_div_0
  }
  text_div.style.zIndex = 1
  slide_order = order;
}

function switch_slides() {
  set_slide_order(1 - slide_order)
}

function tick() {
  // var m,l;
  if (tl>0) {
    // m = ((tl-1)*(viewX0+viewX1) + (toX0+toX1))/(2.*tl)
    // l = Math.pow(Math.pow(viewX1-viewX0, tl-1) * (toX1-toX0), 1./tl)
    // viewX0 = m-l/2.; viewX1 = m+l/2.
    // m = ((tl-1)*(viewY0+viewY1) + (toY0+toY1))/(2.*tl)
    // l = Math.pow(Math.pow(viewY1-viewY0, tl-1) * (toY1-toY0), 1./tl)
    // viewY0 = m-l/2.; viewY1 = m+l/2.
    viewX0 += (toX0-viewX0)/tl; viewX1 += (toX1-viewX1)/tl
    viewY0 += (toY0-viewY0)/tl; viewY1 += (toY1-viewY1)/tl
    --tl; set_view();
  }
}

function resized() {set_view()}

function set_view() {
  var w,h;
  mode = "normal";
  rectangle_div.style.display = "none"
  if (browser == "IE") {
    winW = document.body.offsetWidth; winH = document.body.offsetHeight
  } else {
    winW = window.innerWidth; winH = window.innerHeight
  }
  var scaleX = maxX*winW/slideW/(viewX1-viewX0)
  var scaleY = maxY*winH/slideH/(viewY1-viewY0)
  if (scaleY < scaleX) { // View is vertically bound.
    // y->my*y+ny maps viewY0->0 and viewY1->winH:
    my = winH/(viewY1-viewY0); ny = -viewY0*winH/(viewY1-viewY0)
    // x->mx*x+nx preserves proportions and maps (viewX0+viewX1)/2->winW/2:
    mx = my/maxX*maxY*slideW/slideH; nx = winW/2-mx*(viewX0+viewX1)/2
    h = maxY*winH/(viewY1-viewY0)
    if (h != hidden_slide.height) {hidden_slide.height=h
    hidden_slide_div.style.top=-viewY0*slideH*scaleY/maxY
    w = h*(slideW/slideH);
    if (w != hidden_slide.width) {hidden_slide.width=w
    hidden_slide_div.style.left = winW/2-scaleY*slideW*(viewX0+viewX1)/2/maxX
  } else { // View is horizontally bound.

```

```

// x->mx*x+nx maps viewX0->0 and viewX1->winW:
mx = winW/(viewX1-viewX0); nx = -viewX0*winW/(viewX1-viewX0)
// y->yx*y+ny preserves proportions and maps (viewY0+viewY1)/2->winH/2:
my = mx/maxY*maxX*slideH/slideW; ny = winH/2-my*(viewY0+viewY1)/2
w = maxX*winW/(viewX1-viewX0)
if (w != hidden_slide.width) {hidden_slide.width=w
hidden_slide_div.style.left = -viewX0*slideW*scaleX/maxX
h = w*(slideH/slideW).
if (h != hidden_slide.height) {hidden_slide.height=h
hidden_slide_div.style.top = winH/2-scaleX*slideH*(viewY0+viewY1)/2/maxY
}
}
if (slide_set != undefined) {
var best_i=0; var best_q=0
var q, iw;
if (tl==0) {
for (i=0; i<slide_set.length; ++i)
iw = image_set[i].width
if (w<=iw) {q = w/iw} else {q = iw/w}
if (q>best_q) {best_i = i; best_q = q}
}
} else {
iw = image_set[0].width
if (w<=iw) {best_q = w/iw} else {best_q = iw/w}
best_i = 0;
}
hidden_slide.src = image_set[best_i].src
info_span.innerHTML = (
'<br>Showing ' + slide_set[best_i] +
' at ' + Math.round(w) + 'x' + Math.round(h) +
' in ' + winW + 'x' + winH +
' with q=' + Math.round(1000*best_q)/100(
);
}
switch_slides();
}

function set_visibility(id, level) {
var el = document.getElementById(id)
if (text_level >= level) el.style.display = 'inline'
else el.style.display = 'none'
}

function set_textmode() {
set_visibility("title_text", 1)
set_visibility("beginner_help", 2)
set_visibility("navigation_text", 2)
set_visibility("HandoutBrowser_banner", 2)
set_visibility("help_div", 2)
set_visibility("intermediate_help", 3)
set_visibility("menu_div", 3)
set_visibility("info_span", 3)
}

function set_menu() {
menu="var views = [<br>\n"
document.viewform.viewselect.options.length=n
for (i=0; i<n; ++i) {
document.viewform.viewselect.options[i] =
new Option((i+1)+"/"+n+" "+views[i][4])
menu += "<a href='\"+javascript: goto_view(\"+i+\");\n"
if (k==i) {menu += " style=\\\"background-color: pink;\\n"
else {menu += " style=\\\"background-color: yellow;\\n"
menu += ">[";
var l = views[i].length
for (j=0; j<l; ++j) {
var v=views[i][j]
if (isNaN(v)) menu += "\\n"
menu += v;
if (isNaN(v)) menu += "\\n"
if (j<l-1) menu += ", "
}
menu += "</a>";
if (i<n-1) menu += ", "
}

```

```
    menu += "<br>\n";
  }
  menu += "];";
  document.viewform.viewselect.selectedIndex = k;
  menu_div.innerHTML = menu;
}

function viewselected() {
  goto_view(document.viewform.viewselect.selectedIndex)
}

function goto_view(i) {
  k=i;
  toX0=views[k][0]; toY0=views[k][1]
  toX1=views[k][2]; toY1=views[k][3]
  set_menu();
  tl = ticks_per_glide;
}

function previous() {if (k>0) {--k} else {k=n-1}; goto_view(k);}
function next() {if (k<n-1) {++k} else {k=0}; goto_view(k);}

function rename() {
  var name = prompt("Rename current view", views[k][4])
  if (name != null) {
    views[k][4] = name;
    set_menu();
  }
}

function expunge() {
  if (n>0) {
    views.splice(k, 1); --n;
    if (k==n) --k;
    goto_view(k);
  }
}

function flip_with(j) {
  if (j>=0 && j<n) {
    var t=views[k];
    views[k] = views[j];
    views[j] = t;
    goto_view(j);
  }
}

function toggle_gliding() {
  if (ticks_per_glide == 1) ticks_per_glide=rate; else
    ticks_per_glide=1;
  if (tl>0) tl=1;
}

function set_capture() {
  if (mode != "capture") {
    mode = "capture";
    rectangle_div.style.left = 0; rectangle_div.style.top = 0
    rectangle_div.style.width = 250; rectangle_div.style.height = 60
    rectangle_div.style.display = "block"
  } else {
    mode = "normal";
    rectangle_div.style.display = "none"
  }
}

function shift_view(nx, ny) {
  dx=(viewX1-viewX0)/100;
  dy=(viewY1-viewY0)/100;
  viewX0 += nx*dx; viewY0 += ny*dy
  viewX1 += nx*dx; viewY1 += ny*dy
  set_view();
}
```

```
}  
  
function find_mouse(event) {  
  var x, y;  
  if (browser == "IE") {  
    x = window.event.clientX + document.documentElement.scrollLeft  
      + document.body.scrollLeft  
    y = window.event.clientY + document.documentElement.scrollTop  
      + document.body.scrollTop  
  } else {  
    x = event.clientX + window.scrollX  
    y = event.clientY + window.scrollY  
  }  
  return [x,y];  
}  
  
function drag_start(event) {  
  var x, y;  
  fromX0 = viewX0; fromX1 = viewX1  
  fromY0 = viewY0; fromY1 = viewY1  
  mouse_start = find_mouse(event)  
  if (mode == "capture") {  
    rectangle_div.style.left = mouse_start[0]  
    rectangle_div.style.top = mouse_start[1]  
  } else {  
    x = parseInt(visible_slide_div.style.left, 10)  
    if (isNaN(x)) x = 0;  
    y = parseInt(visible_slide_div.style.top, 10)  
    if (isNaN(y)) y = 0;  
    slide_start = [x,y];  
  }  
  if (browser == "IE") {  
    document.attachEvent("onmousemove", drag)  
    document.attachEvent("onmouseup", drag_end)  
    window.event.cancelBubble = true  
    window.event.returnValue = false  
  } else {  
    document.addEventListener("mousemove", drag, true)  
    document.addEventListener("mouseup", drag_end, true)  
    event.preventDefault();  
  }  
}  
  
function drag(event) {  
  mouse_now = find_mouse(event);  
  if (mode == "capture") {  
    rectangle_div.style.width = mouse_now[0] - mouse_start[0]  
    rectangle_div.style.height = mouse_now[1] - mouse_start[1]  
  } else {  
    viewX0 = fromX0 - (mouse_now[0] - mouse_start[0])/mx  
    viewX1 = fromX1 - (mouse_now[0] - mouse_start[0])/mx  
    viewY0 = fromY0 - (mouse_now[1] - mouse_start[1])/my  
    viewY1 = fromY1 - (mouse_now[1] - mouse_start[1])/my  
    visible_slide_div.style.left =  
      slide_start[0] + mouse_now[0] - mouse_start[0]  
    visible_slide_div.style.top =  
      slide_start[1] + mouse_now[1] - mouse_start[1]  
  }  
  if (browser == "IE") {  
    window.event.cancelBubble = true  
    window.event.returnValue = false  
  } else event.preventDefault();  
}  
  
function drag_end(event) {  
  if (mode == "capture") {  
    viewX0 = (mouse_start[0]-nx)/mx; viewX1 = (mouse_now[0]-nx)/mx  
    viewY0 = (mouse_start[1]-ny)/my; viewY1 = (mouse_now[1]-ny)/my  
    viewX0 = Math.round(viewX0/step)*step  
    viewY0 = Math.round(viewY0/step)*step  
    viewX1 = Math.round(viewX1/step)*step  
    viewY1 = Math.round(viewY1/step)*step  
    views.splice(k+1, 0, [viewX0, viewY0, viewX1, viewY1]); ++n
```

```
    rectangle_div.style.display = "none"
    goto_view(k+1);
}
if (browser == "IE") {
    document.detachEvent("onmousemove", drag)
    document.detachEvent("onmouseup", drag_end)
} else {
    document.removeEventListener("mousemove", drag, true)
    document.removeEventListener("mouseup", drag_end, true)
}
set_view();
}

function keypressed(e)
{
    var keynum;
    if(window.event) { keynum = e.keyCode } // II
    else if(e.which) { keynum = e.which } // Netscape/Firefox/Opera
    var key = String.fromCharCode(keynum)
    return keyaction(key);
}

function keyaction(key) {
    if (key == 'a') {flip_with(k+1);}
    if (key == 'B') {
        var t=alt_text_level; alt_text_level=text_level; text_level=t
        set_textmode();
    }
    if (key == 'c') {set_capture();}
    if (key == 'g') {toggle_gliding();}
    if (key == 'h') {document.location = visible_slide.src}
    if (key == 'n' || key == String.fromCharCode(34) || key == String.fromCharCode(39) {next()}
    if (key == 'p' || key == String.fromCharCode(33) || key == String.fromCharCode(37) {previous()}
    if (key == 'q') {flip_with(k-1);}
    if (key == 'r') {rename()}
    if (key == 't') {
        ++text_level; if (text_level >= 4) text_level=0
        set_textmode();
    }
    if (key == 'x') {expunge();}
    if (key == '0' || key == 'm') {
        toX0=0; toY0=0; toX1=maxX; toY1=maxY; tl=ticks_per_glide
    }
    if (key == '1' || key == 'j') {shift_view(-1,1);}
    if (key == '2' || key == 'k') {shift_view(0,1);}
    if (key == '3' || key == 'l') {shift_view(1,1);}
    if (key == '4' || key == 'u') {shift_view(-1,0);}
    if (key == '5' || key == 'i') {goto_view(k);}
    if (key == '6' || key == 'o') {shift_view(1,0);}
    if (key == '7') {shift_view(-1,-1);}
    if (key == '8') {shift_view(0,-1);}
    if (key == '9') {shift_view(1,-1);}
    if (key == '+' || key == '|') {
        var dx=(viewX1-viewX0)/200; viewX0+=dx; viewX1-=dx
        var dy=(viewY1-viewY0)/200; viewY0+=dy; viewY1-=dy
        set_view();
    }
    if (key == '-' || key == '[') {
        var dx=(viewX1-viewX0)/200; viewX0-=dx; viewX1+=dx
        var dy=(viewY1-viewY0)/200; viewY0-=dy; viewY1+=dy
        set_view();
    }
}
}
```