

Listing 1: Transcript of Usability Study

```
1  R: (Researcher)
2  P: (Participant)
3
4  ===Flyouts On===
5
6      First system, Flyouts ON.
7
8  R:  Start the system by pressing Apple K.
9  P:      Okay, is there an Apple on here?
10 R:  Yeah, I've mapped Ctrl to Apple.
11 P:      Okay, great.
12 P:      I'll do what I have to do to get started, and...
13          *Presses Apple K*
14          *Chibipoint interface appears*
15 P:      Okay.
```

0.1 ./README

```
1  =Rights:
2  ChibiPoint is made by Alex Birch.
3  Source code license is GNU GPL v3.Full ↗
   ↗ text available in file`LICENSE`.
4
5
6  =Installation:
7  Using Google Chrome,
8  1. Go to chrome://extensions/
9  2. Ensure `Developer mode` at the top is ↗
   ↗ checked.
10 3. `Load unpacked extension`
11 4. Choose this folder, `chibipoint`.
12 5. ChibiPoint should appear in yourlist ↗
   ↗ of Extensions, and be `Enabled`.
13 6. If you wish, you can customise the↗
   ↗ invocation shortcut using the`↗
   ↗ Keyboard Shortcuts` link at the↗
   ↗ bottom of chrome://extensions/
14
15
16 =Invocation:
17 Browse the web as usual.
18 Press Cmd+K (Mac) / Ctrl+K (Windows)to ↗
   ↗ invoke ChibiPoint interface.
19
20 You should see a grid. If youdon't, that↗
```

```
21
22
23 =Usage:
24 Press the shortcuts you see on thegrid ↗
   ↗ to `drill down' in thatdirection.↗
25 Labels fly out from suggestedclickables.↗
   ↗ These `flyouts' can beinvoked by ↗
   ↗ pressing theircorresponding key.
26
27 By default, you drill with the numpad↗
   ↗ directions.
28 Flyout shortcuts are mapped to a 3x3grid↗
   ↗ on the left hand of a Qwerty↗
   ↗ keyboard.
29
30 Caps lock is not supported.
31
32
33 =Customizing keys:
34 By default we use numpad to `drilldown'.↗
   ↗ If you don't have anumpad, you ↗
   ↗ should probably changethe code a ↗
   ↗ little.
35
36 Lines 80-154 concern keybindings.Sorry ↗
   ↗ the world isn't nicer.
37 There are a few defaults you mightwant ↗
   ↗ to try out, so fiddle aroundwith ↗
```

```

38     ↪ numpad = true and Dvorak = false.
39 Just edit the code, and refresh any page ↪
    ↪ that was using ChibiPoint; it should ↪
    ↪ update immediately.

```

0.2 ./manifest.json

```

1  {
2    "name": "ChibiPoint",
3    "description": "Point at webpages using ↪
    ↪ the keyboard",
4    "version": "0.5",
5    "content_scripts": [
6      {
7        "matches": ["↪
    ↪ http://*/*" ↪
    ↪ , "https:↪
    ↪ //*/*"],
8    "css": ["styles.css"],
9    "js": ["src/lib/require.js", "src/↪
    ↪ lib/require-cs.js", "src/↪
    ↪ depend.js"],
10   "run_at": "document_start"
11   }
12 ],
13 "permissions": [
14   "tabs",
15   "activeTab"
16 ],

```

```

17 "background": {"scripts": ["src/↪
18   ↪ background.js"]},
19 "commands": {
20   "toggle-feature-foo": {
21     "suggested_key": {
22       "default": "Ctrl+K",
23       "mac": "Command+K"
24     },
25     "description": "Invoke ChibiPoint"
26   },
27 "manifest_version": 2,
28 "web_accessible_resources": [
29   "*.js"
30 ]
31 }

```

0.3 ./styles.less

```

1  // out: styles.css
2  // compress: false, strictMath: false
3
4
5  @gridcol: #ccc;
6  @gridthick: 1px;
7
8  @growcoeff: 1.05;
9
10
11 body {

```

```

12     padding: 0;
13 }
14
15 .chibiPoint_gridContainer {
16     padding: 0;
17     //top: 8.3%;
18     top: 0;
19     left: calc(((100%*(@growcoeff - 1))/2) / ↗
        ↗ @growcoeff);
20
21     position: fixed;
22     width: calc(100%/@growcoeff);
23     height: calc(100%/@growcoeff);
24
25     pointer-events: none;
26
27     z-index: 2147483646;
28 }
29
30 .chibiPoint_hiddenGridContainer {
31     display: none;
32 }
33
34 .chibiPoint_grid {
35
36     position: relative;
37     left: calc(-(100%*(@growcoeff - 1))/2);
38
39     //top: -10%;
40     top: 0;

```

```

41     //float: left;
42
43     width: calc(100%*@growcoeff);
44     height: calc(100%*@growcoeff);
45
46     //overflow: auto;
47     display: table;
48     table-layout: fixed;
49
50     pointer-events: inherit;
51 }
52
53 .chibiPoint_gridBorderOn {
54     border: @gridthick solid @gridcol;
55     border-width: @gridthick @gridthick ↗
        ↗ @gridthick @gridthick;
56     //outline: 1px solid 0;
57 }
58
59 .chibiPoint_gridBorderOff {
60 }
61
62 .chibiPoint_row {
63     //border: 1px solid #ccc;
64     //border-width: 0 1px 1px 0;
65
66     display: table-row;
67     pointer-events: inherit;
68
69 }

```

5

70			
71	@cellratio:calc(100%/3);	99	↪ @gridthick @gridthick;
72		100	//border-style:double;
73	.chibiPoint_cell {	101	outline: 1px solid 0;
74	padding:0;	102	outline-offset: 1px;*/
75		103	//this one!
76	width:@cellratio;	104	box-shadow: 0 0 0 1px #fff, 0 0 02px ↪
77	height:@cellratio;		↪ #888;
78	display: table-cell;	105	}
79		106	
80		107	.chibiPoint_cellBorderOff {
81	// text	108	}
82	line-height: @cellratio;	109	
83	text-align:center;	110	@backingprop: 0.8;
84	vertical-align:middle;	111	@radius: 15%;
85	color: #000;	112	@backinggrey: 230;
86	color:rgba(0, 0, 0, 0.5);	113	
87	font-family:Geneva, Arial,Helvetica, ↪	114	.chibiPoint_backing {
	↪ sans-serif;	115	position:absolute;
88		116	
89	pointer-events: inherit;	117	top:calc(100%*(1-@backingprop)/2);
90		118	left:calc(100%*(1-@backingprop)/2);
91	position:relative;	119	
92	overflow:visible;	120	width:calc(100%*@backingprop);
93	clear:both;	121	height:calc(100%*@backingprop);
94	}	122	background:rgba(@backinggrey, ↪
95			↪ @backinggrey, @backinggrey, 0.25) ↪
96	.chibiPoint_cellBorderOn {		↪ ;
97	/*border: @gridthick solid @gridcol;	123	
98	border-width: @gridthick @gridthick ↪	124	z-index:-1;

```
125  
126 -moz-border-radius: @radius @radius;  
127 -webkit-border-radius: @radius@radius;  
128 border-radius: @radius @radius;  
129  
130 //background-blend-mode: difference, ↵  
131 ↵ normal; // Chrome Canary  
132 // -webkit-background-blend-mode: ↵  
133 ↵ difference, normal; // WebKit ↵  
134 ↵ Nightly  
135 }  
136 @haircol: rgba(255, 0, 0, 1);  
137 @hairthick: 1px;  
138 @hairborderthick: 1px;  
139 .chibiPoint_fullContainer {  
140     top: 0;  
141     left: 0;  
142     width: 100%;  
143     height: 100%;  
144     position: fixed;  
145     padding: 0;  
146     pointer-events: none;  
147 }  
148 .chibiPoint_shownCrosshairs {  
149  
150  
151  
152  
153  
154  
155  
156  
157 .chibiPoint_hiddenCrosshairs {  
158     pointer-events: none;  
159  
160     display: none;  
161 }  
162  
163 .chibiPoint_hairBorder {  
164     //border-width: @hairborderthick ↵  
165     ↵ @hairborderthick @hairborderthick ↵  
166     ↵ @hairborderthick;  
167     //border-color: #00ffff;  
168  
169     //outline-color:;  
170     //outline: 1px solid 0;  
171 }  
172  
173 .chibiPoint_hairlinecommon {  
174     padding: 0;  
175  
176     position: relative;  
177  
178     pointer-events: inherit;  
179  
180     background: @haircol;  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```

```

180 .chibiPoint_hairBorder;
181 }
182
183 .chibiPoint_horizontalHair {
184     .chibiPoint_hairlinecommon;
185
186     left:inherit;
187
188     width:inherit;
189     height:@hairthick;
190 }
191
192 .chibiPoint_verticalHair {
193     .chibiPoint_hairlinecommon;
194
195     top:@hairthick;
196
197     width:@hairthick;
198     height:inherit;
199 }
200
201 .chibiPoint_targeted {
202     background: yellow!important;
203     //padding: 3px 5px;
204     //margin: -3px -5px;
205     //line-height: 1.7;
206
207     border-radius: 3px;
208     /*border-color: red!important;
209     border: 3px solid black;

```

```

210     box-sizing: border-box;*/
211     //border-radius: 3px;
212     box-shadow: 0 0 10px #ff00ff;
213
214     // turn off cros hairs:
215     //display: none;
216
217     background-blend-mode: multiply, ↗
218         ↗ normal; // Chrome Canary
219     -webkit-background-blend-mode: ↗
220         ↗ multiply, normal; // WebKit ↗
221         ↗ Nightly
222
223     //display:inline-block;
224 }
225
226 .chibiPoint_painted {
227     background: cyan!important;
228     //padding: 3px 5px;
229     //margin: -3px -5px;
230     //line-height: 1.7;
231
232     border-radius: 3px;
233     /*border-color: red!important;
234     border: 3px solid black;
235     box-sizing: border-box;*/
236     //border-radius: 3px;
237     box-shadow: 0 0 10px #ff0000;
238
239     background-blend-mode: multiply, ↗

```

	↪ normal; // Chrome Canary	258	}
237	-webkit-background-blend-mode:↪	259	.chibiPoint_paintedUnique6 {
	↪ multiply, normal; // WebKit ↪	260	background: rgba(240,172,24,↪
	↪ Nightly		↪ @flyoutAlpha)!important;
238	//display:inline-block;	261	}
239	}	262	.chibiPoint_paintedUnique7 {
240		263	background: rgba(8,160,94,@flyoutAlpha↪
241	.chibiPoint_paintedUnique0 {		↪)!important;
242	background: rgba(0,173,0,@flyoutAlpha)↪	264	}
	↪ !important;	265	.chibiPoint_paintedUnique8 {
243	}	266	background: rgba(135,54,94,↪
244	.chibiPoint_paintedUnique1 {		↪ @flyoutAlpha)!important;
245	background: rgba(13,255,150,↪	267	}
	↪ @flyoutAlpha)!important;	268	
∞ 246	}	269	.chibiPoint_cluck {
247	.chibiPoint_paintedUnique2 {	270	background: red!important;
248	background: rgba(249,25,137,↪	271	//padding: 3px 5px;
	↪ @flyoutAlpha)!important;	272	//margin: -3px -5px;
249	}	273	//line-height: 1.7;
250	.chibiPoint_paintedUnique3 {	274	
251	background: rgba(249,112,12,↪	275	/*border-radius: 3px;
	↪ @flyoutAlpha)!important;	276	border-color: red!important;
252	}	277	border: 3px solid black;*/
253	.chibiPoint_paintedUnique4 {	278	
254	background: rgba(186,167,19,↪	279	//display:inline-block;
	↪ @flyoutAlpha)!important;	280	}
255	}	281	
256	.chibiPoint_paintedUnique5 {	282	.chibiPoint_flyoutContainer {
257	background: rgba(172,0,237,↪	283	.chibiPoint_fullContainer;
	↪ @flyoutAlpha)!important;	284	

285	z-index:2147483647;	315	↪ sans-serif;
286	}	316	/*-webkit-text-stroke: 1px #000000;
287		317	-webkit-text-fill-color: #FFFFFF;*/
288	.chibiPoint_shownfC {	318	color: white;
289	}	319	text-shadow:
290		320	-1px -1px 0 #000,
291	.chibiPoint_hiddenfC {	321	1px -1px 0 #000,
292	display:none;	322	-1px 1px 0 #000,
293	}	323	1px 1px 0 #000;
294		324	}
295	.chibiPoint_aFlyout {	325	@flyoutAlpha: 1.0;
296	position: absolute;	326	
297	width:100%;	327	
298	height:100%;	328	.chibiPoint_flyoutUnique0 {
299	}	329	background: rgba(0,173,0,@flyoutAlpha)↵
300		330	↪ ;
301	.chibiPoint_aFlyoutHidden {	331	.chibiPoint_flyoutUnique1 {
302	display: none;	332	background: rgba(13,255,150,↵
303	}	333	↪ @flyoutAlpha);
304		334	}
305	.chibiPoint_flyout {	335	.chibiPoint_flyoutUnique2 {
306	//background: rgba(0,173,0, 0.30);	336	background: rgba(249,25,137,↵
307	//outline: 1px solid 0!important;	337	↪ @flyoutAlpha);
308	border: 1px solid black;	338	.
309	position: absolute;	339	.chibiPoint_flyoutUnique3 {
310	top:0;		background: rgba(249,112,12,↵
311	left: 0;		↪ @flyoutAlpha);
312	width:20px;		}
313	text-align: center;		
314	font-family:Geneva, Arial,Helvetica, ↵		

340	.chibiPoint_flyoutUnique4 {		↪ @flyoutLineAlpha);
341	background: rgba(186,167,19,↪	364	}
	↪ @flyoutAlpha);	365	.chibiPoint_lineUnique3 {
342	}	366	stroke: rgb(249,112,12,↪
343	.chibiPoint_flyoutUnique5 {		↪ @flyoutLineAlpha);
344	background: rgba(172,0,237,↪	367	}
	↪ @flyoutAlpha);	368	.chibiPoint_lineUnique4 {
345	}	369	stroke: rgb(186,167,19,↪
346	.chibiPoint_flyoutUnique6 {		↪ @flyoutLineAlpha);
347	background: rgba(240,172,24,↪	370	}
	↪ @flyoutAlpha);	371	.chibiPoint_lineUnique5 {
348	}	372	stroke: rgb(172,0,237,@flyoutLineAlpha↪
349	.chibiPoint_flyoutUnique7 {		↪);
350	background: rgba(8,160,94,@flyoutAlpha↪	373	}
	↪);	374	.chibiPoint_lineUnique6 {
351	}	375	stroke: rgb(240,172,24,↪
352	.chibiPoint_flyoutUnique8 {		↪ @flyoutLineAlpha);
353	background: rgba(135,54,94,↪	376	}
	↪ @flyoutAlpha);	377	.chibiPoint_lineUnique7 {
354	}	378	stroke: rgb(8,160,94,@flyoutLineAlpha)↪
355			↪ ;
356	.chibiPoint_lineUnique0 {	379	}
357	stroke: rgb(0,173,0,@flyoutLineAlpha);	380	.chibiPoint_lineUnique8 {
358	}	381	stroke: rgb(135,54,94,@flyoutLineAlpha↪
359	.chibiPoint_lineUnique1 {		↪);
360	stroke: rgb(13,255,150,↪	382	}
	↪ @flyoutLineAlpha);	383	
361	}	384	
362	.chibiPoint_lineUnique2 {	385	.chibiPoint_lineDiv {
363	stroke: rgb(249,25,137,↪	386	}

11

```

387
388 .chibiPoint_lineSvg {
389 }
390
391 @flyoutLineAlpha:1;
392 @flyoutLineWidth:2;
393 .chibiPoint_line {
394   //stroke:rgb(0,173,0);
395   stroke-width:@flyoutLineWidth;
396 }
397
398 .chibiPoint_evaluator {
399   position:fixed;
400
401   pointer-events: none;
402
403   font-family:Geneva, Arial,Helvetica, ↵
404     ↵ sans-serif;
405
406   color:red;
407
408   width:100%;
409   height:100%;
410
411   display:none;
412 }
413
414 .chibiPoint_timer {
415   .chibiPoint_evaluator;

```

```

416
417   left:10;
418   top:10;
419 }
420
421 .chibiPoint_keyCount {
422   .chibiPoint_evaluator;
423
424   left:10;
425   top:30;
426 }

```

0.4 ./src/depend.js

```

1  /// Cheekily change DOM prototype before ↵
2  ↵ page loads
3  // http://stackoverflow.com/questions ↵
4  ↵ /8915461/ ↵
5  ↵ how-to-find-out-what-event- ↵
6  ↵ listener-types-attached-to-specific- ↵
7  ↵ html-element-in-c
8  // from spektom, thanks to kdzwinel
9
10 var injectedJS = "\
11 (function(original) { \
12   Element.prototype.addEventListener= ↵
13     ↵ function(type, listener, ↵
14     ↵ useCapture) { \
15     var attr =this.getAttribute(' ↵
16     ↵ _handlerTypes'); \

```

```

9      var types = attr ?attr.split(',') : ↵
10      ↵ []; \
11      var found = false; \
12      for (var i = 0; i < types.length;++i↵
13      ↵ ) { \
14          if (types[i] == type) { \
15              found = true; \
16              break; \
17          } \
18          if (!found) { \
19              types.push(type); \
20          } \
21      this.setAttribute('_handlerTypes',↵
22      ↵ types.join(',')); \
23      return original.apply(this,arguments↵
24      ↵ ); \
25      } \
26      })(Element.prototype.addEventListener); ↵
27      ↵ \
28      \
29      (function(original) { \
30      Element.prototype.removeEventListener= ↵
31      ↵ function(type, listener,useCapture)↵
32      ↵ { \
33          var attr =this.getAttribute('↵
34          ↵ _handlerTypes'); \
35          var types = attr ?attr.split(',') : ↵
36          ↵ []; \
37          var removed = false; \
38          for (var i = 0; i < types.length;++i↵
39          ↵ ) { \
40              if (types[i] == type) { \
41                  types.splice(i, 1); \
42                  removed = true; \
43                  break; \
44              } \
45              if (removed) { \
46                  this.setAttribute('_handlerTypes',↵
47                  ↵ types.join(',')); \
48              } \
49              return original.apply(this,arguments↵
50              ↵ ); \
51          } \
52      })(Element.prototype.removeEventListener↵
53      ↵ ); \
54      ";
55      var script =document.createElement("↵
56      ↵ script");
57      script.type = "text/javascript";
58      script.appendChild(document.createTex↵
59      ↵ tNode(injectedJS));
60      document.documentElement.appendChild(↵
61      ↵ script);
62      //
63      url = "/src/";

```

```

53 if (chrome.extension !== undefined) {
54   url = chrome.extension.getURL("/src/");
55 }
56
57 require({ baseUrl:url },["lib/↵
    ↵ jquery-2.1.0.min","lib/within", "↵
    ↵ lib/Blob","lib/FileSaver", "↵
    ↵ evaluator", "test","trace", "lookup↵
    ↵ ", "testonly","setup", "Grid", "↵
    ↵ Crosshairs","Flyout"], function ($,↵
    ↵ within, blob,saver, evaluator, ↵
    ↵ test, trace,lookup, testonly, setup↵
    ↵ , gridclass,crosshairclass, flyout)↵
    ↵ {
58 // default namespace
59 window.birchlabs = {};
60
61 //This function is called when↵
    ↵ scripts/helper/util.js is ↵
    ↵ loaded.
62 //If util.js calls define(), then↵
    ↵ this function is not fired ↵
    ↵ until
63 //util's dependencies have loaded, ↵
    ↵ and the util argument will hold
64 //the module value for"helper/util".
65
66 if (typeof(chrome) == "object" &&↵
    ↵ chrome.extension) {
67 //chrome.extension.sendRequest({'↵

```

```

68     ↵ get_options': true},function(↵
    ↵ response) {
69     setup.main();
70 //});
71 } else {
72     setup.main();
73 }
});

```

0.5 ./src/test.js

```

1 define(["lib/jquery-2.1.0.min","lib/↵
    ↵ within", "lib/Blob","evaluator", "↵
    ↵ trace", "lookup","testonly", "Grid"↵
    ↵ , "Crosshairs","Flyout"], function ↵
    ↵ (jq, within,blob, evaler, tracer, ↵
    ↵ lookup,testonly, gridclass, ↵
    ↵ crosshairclass,flyout) {
2 var trace = tracer.trace;
3 var supertrace = tracer.supertrace;
4
5 window.birchlabs =window.birchlabs↵
    ↵ ||{};
6 var birchlabs = window.birchlabs;
7
8 window.evaluator =window.evaluator↵
    ↵ ||{};
9 var evaluator = window.evaluator;
10
11 //console.log(birchlabs);

```

```

12
13 function makeContainers() {
14     var Flyout = birchlabs.Flyout;
15     var Grid = birchlabs.Grid;
16     var Crosshairs = ↵
        ↵ birchlabs.Crosshairs;
17
18     if (birchlabs.flyoutsOn) {
19         Flyout.makecontainer(document.d ↵
            ↵ ocumentElement);
20         var flyouts = [];
21     }
22     Flyout.hide();
23
24     var grid = new Grid();
25     var container = makeGridContainer(↵
        ↵ document.documentElement);
26
27     var crosshairs = newCrosshairs(↵
        ↵ document.documentElement, grid);
28     crosshairs.hide();
29
30     birchlabs.grid = grid;
31     birchlabs.container = container;
32     birchlabs.crosshairs = crosshairs;
33     //birchlabs.theFlyout = flyout;
34     birchlabs.flyouts = flyouts;
35 }
36
37 function makeGridContainer(root) {

```

```

38     var parent = document.createElement(' ↵
        ↵ div');
39     parent.className = ' ↵
        ↵ chibiPoint_gridContainer ↵
        ↵ chibiPoint_hiddenGridContainer' ↵
        ↵ ;
40     $(root).append(parent);
41     return parent;
42 }
43
44 function init() {
45     birchlabs.evaluateTabMode = false;
46     birchlabs.disallowTabbing = false;
47     birchlabs.flyoutsOn = true;
48     if (birchlabs.evaluateTabMode) {
49         birchlabs.flyoutsOn = false;
50     }
51     makeContainers();
52     evaler.makeEvaluators();
53     var Flyout = birchlabs.Flyout;
54     var Grid = birchlabs.Grid;
55     var Crosshairs = ↵
        ↵ birchlabs.Crosshairs;
56
57     var flyouts = birchlabs.flyouts;
58     var grid = birchlabs.grid;
59     var crosshairs = ↵
        ↵ birchlabs.crosshairs;
60
61     birchlabs.targetedElement = null;

```

```

62
63 var keycodes = {
64     "backspace": 8,
65     "tab": 9,
66     "enter": 13,
67     "spacebar": 32,
68     "escape": 27,
69     "n": 78,
70     "p": 80,
71     "g": 71,
72     "f3": 114,
73     "f4": 115,
74     "dot": 46,
75     "zero": 48,
76     "activate": 167
77 };
78
79 // Do you have a numpad?
80 var numpad = true;
81
82 if (numpad) {
83     // Drilling: numpad
84     // 7 8 9
85     // 4 5 6
86     // 1 2 3
87     var numpadMappings = {"7": [55, 7], "↵": [56, 8], "9": [57, 9],
88                             "4": [52, 4], "5": [53, 5], "↵": [54, 6],

```

```

89     "1": [49, 1], "2": [50, 2], "↵": [51, 3]};
90
91     var zeroKey = keycodes.zero;
92     // Undo drilling:
93     // 0
94 }
95
96 // Are you one of.. those?
97 var dvorak = false;
98
99 if (dvorak) {
100     // Flyouts: right hand dvorak grid
101     // G C R
102     // H T N
103     // M W V
104     var flyoutShortcuts = [103, 99, 114,
105                             104, 116, 110,
106                             109, 119, 118];
107
108     if (!numpad) {
109         // Drilling: left hand dvorak↵
110         // ↵ grid
111         // ' , .
112         // A O E
113         // ; Q J
114         var numpadMappings = {"7": [39, 7], ↵: [44, 8], "9": [46, 9],
115                                 "4": [97, 4], "5": [111, 5], "6": [101, 6],

```

```

114 "1": [59, 1], "2": [113, 2], "3": [106, 3]]; 142
115     // Undo drilling: 143
116     // K
117     var zeroKey = 107; 144
118 } 145
119 } else { 146
120     // Flyouts: left hand qwerty grid 147
121     // Q W E
122     // A S D
123     // Z X C 148
124     var flyoutShortcuts = [113, 119, 101, 149
125                             97, 115, 100, 150
126                             122, 120, 99]; 151
127
128     // alphabet flyouts 152
129     // nah, these won't help anyone 153
130     /*var flyoutShortcuts = [97, 98, 99, 154
131                             100, 101, 102, 155
132                             103, 104, 105];*/ 156
133
134     if (!numpad) { 157
135         // Flyouts: right hand qwerty ↵ 158
136         ↵ grid 159
137         // U I O 160
138         // J K L
139         // N M , 161
140         var flyoutShortcuts = ↵ 162
141         ↵ [117, 105, 111, 163
142             106, 107, 108, 164
143             109, 44, 46];

```

```

    // Drilling: left hand qwerty ↵
    ↵ grid
    // Q W E
    // A S D
    // Z X C
    var numpadMappings = {"7" ↵
        ↵ : [113, 7], "8" : [119, 8], "9" ↵
        ↵ : [101, 9],
"4" : [97, 4], "5" : [115, 5], "6" : [100, 6],
"1" : [122, 1], "2" : [120, 2], "3" : [99, 3]};
    // Undo drilling:
    // V
    var zeroKey = 118;
}
}

birchlabs.numpadMappings = ↵
    ↵ numpadMappings;

if (birchlabs.flyoutsOn) {
    for (var i=0; ↵
        ↵ i<flyoutShortcuts.length; i ↵
        ↵ ++ ) {
        flyouts.push(new Flyout( ↵
            ↵ flyoutShortcuts[i]));
    }
}

//createGrid();

```



```

165     function setEvents(rootNode) {
166         var doc = ↵
            ↵ rootNode.contentDocument ↵
            ↵ || rootNode;
167         var body = rootNode.body;
168
169         if (!body || !↵
            ↵ body.addEventListener) {
170             return;
171         }
172
173         lookup.setAlternativeActive ↵
            ↵ Document(document);
174
175         if (document.getElementById("↵
            ↵ chibipointtestmode") != ↵
            ↵ null) {
176             birchlabs.testmode = true;
177         }
178
179         //trace(birchlabs.testmode);
180
181         // usually on keydown
182         //processSearch();
183         //draw();
184         //var container = ↵
            ↵ birchlabs.container;
185
186         if (birchlabs.testmode) {
187
188             toggleGrid();
189         }
190
191         var $window = $(window),
192             resize_ok = true,
193             timer;
194
195         timer = setInterval(function() {
196             resize_ok = true;
197         }, 50);
198
199         liveTargeter = function() {
200             if (resize_ok === true) {
201                 resize_ok = false;
202                 //trace($window.width());
203                 //trace($(".verticalHair") ↵
                    ↵ .first().offset().↵
                    ↵ left);
204                 //highlightTarget();
205                 crosshairs.highlight(getDocRoot());
206                 pointFlyouts();
207             }
208         };
209
210         $window.resize(liveTargeter);
211         $window.scroll(liveTargeter);
212
213         // some keys can only be caught ↵
            ↵ by keydown
214         doc.addEventListener('keydown', function(↵

```

```

215     ↪ ev) {
216         var code = ev.keyCode;
217         var ascii =String.fromCharCode
218             ↪ (code);
219
220         if (code == keycodes.enter) {
221             evaler.evaluatorIncrementKeys(code);
222             if(birchlabs.evaluateTabMode
223                 ↪ ) {
224                 evaler.evaluatorWriteState
225                     ↪ (
226                     ↪ document.activeElement
227                     ↪ );
228             } else {
229                 gridclick(grid);
230                 lookup.stopEvent(ev);
231             }
232
233             if (code == keycodes.tab) {
234                 if(birchlabs.evaluateTabMode
235                     ↪ ) {
236                     // negative, for shift-tab
237                     ↪ !
238                     var code2 = ev.shiftKey?
239                         ↪ -code : code;
240
241                 evaler.evaluatorIncrementKeys(code2);
242                 evaler.checkFocusAfterWait();
243
244             }
245
246         }
247     }
248
249     if (code == keycodes.tab) {
250         if(birchlabs.evaluateTabMode
251             ↪ ) {
252             // negative, for shift-tab
253             ↪ !
254             var code2 = ev.shiftKey?
255                 ↪ -code : code;
256
257             evaler.evaluatorIncrementKeys(code2);
258             evaler.checkFocusAfterWait();

```

```

236     } else {
237         if(
238             ↪ birchlabs.disallowTabbing
239             ↪ ) {
240             alert("Please do not
241                 ↪ press tab during
242                 ↪ this experiment!");
243             lookup.stopEvent(ev);
244             document.activeElement.blur();
245         }
246     }
247
248     /*if (code ==keycodes.spacebar
249         ↪ ) {
250         evaler.evaluatorIncrementKeys(code);
251     }*/
252     if(!birchlabs.evaluateTabMode)
253         ↪ {
254         if (code ==keycodes.escape)
255             ↪ {
256             if (!gridIsEmpty()) {
257                 evaler.evaluatorIncrementKeys(code);
258                 closeGrid();
259
260                 lookup.stopEvent(ev);
261             }
262         }
263     }
264 }));

```

```

259         // no key bindings in tab mode
260         if(!birchlabs.evaluateTabMode) {
261             doc.addEventListener('keypress',function↵
                ↵ (ev) {
262                 if(lookup.isInputElementActive↵
                    ↵ (doc)) {
263                     var tracer =↵
                        ↵ doc.getElementById("↵
                            ↵ trace");
264                     if (tracer != null) {
265                         doc.getElementById("trace").innerHTML= ↵
                            ↵ code;
266                     }
267                     return;
268                 }
269                 var code = ev.keyCode;
270                 var ascii =String.fromCharCode↵
                    ↵ (code);
271
272                 if (!lookup.is_shortcut(ev)&& ↵
                    ↵ ascii &&[keycodes.enter].↵
                    ↵ indexOf(code) == -1){
273                     var tracer =↵
                        ↵ doc.getElementById("↵
                            ↵ trace");
274                     if (tracer != null) {
275                         doc.getElementById("trace").innerHTML= ↵
                            ↵ code;
276                     }
277

```

```

        /*if (numpadtype) {
            var num = code-↵
                ↵ numpadstart;
            if (num<10 && num > 0){
                if (gridIsInUse()) {
                    drill(num, grid,↵
                        ↵ crosshairs);

                    lookup.stopEvent(ev);
                }
            } else if (code ==↵
                ↵ keycodes.zero) {
                backup(grid,crosshairs);

                lookup.stopEvent(ev);
            }
        }*/
        for (var j in numpadMappings)↵
            ↵ {
                if (code ==numpadMappings[↵
                    ↵ j][0]) {
                    if (gridIsInUse()) {
evaler.evaluatorIncrementKeys(code);
drill(numpadMappings[j][1], grid,↵
    ↵ crosshairs);

                    lookup.stopEvent(ev);
                    break;
                }
            }
        }

```

```

302         }
303         if (code == zeroKey) {
304     evaler.evaluatorIncrementKeys(code);
305             backup(grid, crosshairs);
306
307             lookup.stopEvent(ev);
308         }
309         if (code == keycodes.activate ↵
310             ↵ ) {
311     evaler.evaluatorIncrementKeys(code);
312             toggleGrid();
313
314             lookup.stopEvent(ev);
315         }
316         if (birchlabs.flyoutsOn) {
317             for (i=0; ↵
318                 ↵ i<flyoutShortcuts.length ↵
319                 ↵ i++) {
320                 if (code == ↵
321                     ↵ flyoutShortcuts[i]) ↵
322                     ↵ {
323     evaler.evaluatorIncrementKeys(code);
324             evaler.evaluatorWri ↵
325                 ↵ teState(flyouts[i] ↵
326                 ↵ ].getTarget());
327             flyouts[i].doClick();
328
329             closeGrid();
330
331             lookup.stopEvent(ev);

```

```

325         break;
326     }
327 }
328 }
329 }
330 }, false);
331 }
332 }
333
334 // guess this defers settingevents ↵
335 ↵ until root is loaded?
336 var rootNodes = [document].concat(↵
337     ↵ lookup.getRootNodes());
338 for (var i = 0; i < ↵
339     ↵ rootNodes.length; i++) {
340     var rootNode = rootNodes[i];
341     if (rootNode.contentDocument) {
342     rootNode.addEventListener('load', ↵
343         ↵ function(ev) {
344             setEvents(ev.target.contentDocument);
345             });
346     }
347     else if (!rootNode.contentDocument ↵
348         ↵ || ↵
349         ↵ rootNode.contentDocument.readyState ↵
350         ↵ == 'complete') {
351     setEvents(rootNode.contentDocument ? ↵
352         ↵ rootNode.contentDocument : rootNode ↵
353         ↵ );
354     }

```

```

346     }
347     //draw();
348 }
349
350 function getDocRoot() {
351     var rootNodes = [window].concat(lookup.getRootNodes());
352     for (var x = 0; x < rootNodes.length; x++) {
353         var doc2 = rootNodes[x].document || rootNodes[x].contentDocument;
354         if (!doc2 || !doc2.body)
355             continue;
356         return doc2;
357     }
358 }
359
360 function
361     ↪ getAllClickablesInBoundsAndSort(↪ root, rect, rects) {
362     var found = [];
363     var buckets = [];
364     for (var j=0; j<rects.length; j++) ↪ {
365         ↪ buckets.push([]);
366     }
367     var left = rect.left;
368     var top = rect.top;
369     var width = rect.width;

```

```

370     var height = rect.height;
371
372     var left2;
373     var top2;
374     var width2;
375     var height2;
376
377     var cache = false;
378
379     var r;
380     var i;
381     var res;
382     var res2;
383     //var q;
384     //var ew;
385     var itRect;
386     // some of the obscure ones are ↪
387     ↪ guesses
388     var careElements = {"A":true,
389                         "INPUT":true,
390                         "TEXTAREA":true,
391                         "SELECT":true,
392                         "BUTTON":true//,
393                         // "output":true,
394                         // "command":true,
395                         // "kbd":true
396                     };
397
398     $(root).find("[_handlerTypes],A, ↪
399     ↪ INPUT, SELECT, TEXTAREA, ↪

```

22

```

398         ↪ BUTTON").each(function() {
399         var found = false;
399         var handlers =this.getAttribute(↪ 417
        ↪ '_handlerTypes');
400         if (handlers != null ) {
401             var h = handlers.split(",");          418
402             for (i=0; i<h.length; i++) {          419
403                 if (h[i] == 'click' ||h[i] ↪
        ↪ == 'mousedown') {
404                     found = true;                420
405                     break;                        421
406                 }
407             }
408         }
409         if (!found) {
410         if(careElements[this.nodeName]) found= ↪
        ↪ true;                                          422
411         }                                          423
412         if (found) {
413             //if ($(this).css("visibility↪ 424
        ↪ ").indexOf("hidden") != ↪ 425
        ↪ -1 || !$(this).css("↪ 426
        ↪ display").indexOf("none") ↪
        ↪ !=-1) return;                                427
414         var visible =!$(this).is(':↪
        ↪ hidden') &&$(this).is(':↪ 428
        ↪ visible');                                    429
415         if (visible) {                            430
416             //if(this ==↪ 431
        ↪ document.documentElement↪432

```

```

        ↪ ) returnthis.ret.push(↪
        ↪ this);
        if(this ==↪
        ↪ document.documentElement↪
        ↪ ) return;

        itRect =↪
        ↪ this.getBoundingClientRect↪
        ↪ ();

        res = !( (itRect.top >top+↪
        ↪ height) || (itRect.top+↪
        ↪ itRect.height < top) ||↪
        ↪ (itRect.left> left+↪
        ↪ width ) ||(itRect.left+↪
        ↪ itRect.width < left));

        // it's certainly in thegrid ↪
        ↪ somewhere
        if(res) {

            // now find whichbuckets ↪
            ↪ to put it in
            for (i=0;i<rects.length;↪
            ↪ i++) {
                r = rects[i];
                left2 = r.left;
                top2 = r.top;
                width2 = r.width;
                height2 = r.height;

```

```

433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475

```

```

    res2 = !((itRect.top > top2+height2) ||
    (itRect.top + itRect.height < top2) ||
    (itRect.left > left2+width2) ||
    (itRect.left + itRect.width < left2));

    if (res2) {
        buckets[i].push(this);
    }
}

});
return buckets;
}

function highlightTarget() {
    var crosshairs = birchlabs.crosshairs;
    var flyouts = birchlabs.flyouts;
    var grid = birchlabs.grid;
    var doc2 = getDocRoot();

```

```

    var crosshaired = crosshairs.highlight(doc2).get(0);

    // some of the obscure ones are guesses
    var careElements = {"A":true,
                        "INPUT":true,
                        "TEXTAREA":true,
                        "SELECT":true,
                        "BUTTON":true//,
                        //"output":true,
                        //"command":true,
                        //"kbd":true
    };

    var rect = grid.getLastGrid().get(0).getBoundingClientRect();

    var rects = [];
    for (var i=0; i<3; i++) {
        for (var j=0; j<3; j++) {
            rects.push(grid.getLastRows().eq(i).children().eq(j).get(0).getBoundingClientRect());
        }
    }
    if (birchlabs.flyoutsOn) {

```

```

476     var buckets =getAllClickables↵
         ↵ InBoundsAndSort($(doc2.body↵ 494
         ↵ ), rect,rects);

477
478     // which element was selectedby ↵ 495
         ↵ each segment ↵ 496
479     var selected = [];
480     //console.log(crosshaired); ↵ 497
481
482     var bucketIndex = 0; ↵ 498
483     // one bucket for each segment ↵ 499
484     for (var i=0; i<3; i++) { ↵ 500
485         for (var j=0; j<3; j++) { ↵ 501
486             var fulfilled = false; ↵ 502
487             for (var b=0; b<buckets.length;↵ 503
                 ↵ b++) { ↵ 504
488                 // check everything in my↵ 505
                 ↵ allocated bucket first ↵ 506
489                 // but be prepared tocycle ↵ 507
                 ↵ round to another bucket↵ 508
                 ↵ ! ↵ 509
490             var myBucket = buckets[(b↵ 510
                 ↵ ucketIndex+b)%↵ 511
                 ↵ buckets.length]; ↵ 512
491             for (var n = 0; ↵ 513
                 ↵ n<myBucket.length; n++) ↵ 514
                 ↵ { ↵ 515
492                 var inSelectedAlready =↵ 516
                 ↵ false; ↵ 517
493                 if (myBucket[n] !=↵ 518

```

```

         ↵ crosshaired) {
         for (var s = 0;↵
             ↵ s<selected.length; ↵
             ↵ s++) {
                 // skip over
                 if (selected[s] ==↵
                     ↵ myBucket[n])
                     {
                         inSelectedAlready= ↵
                             ↵ true;
                         break;
                     }
                 }
                 if(!inSelectedAlready) {
                     // add to selected
                     selected[bucketIndex] = myBucket[n];
                     fulfilled = true;
                     break;
                 }
             }
             if (fulfilled) {
                 break;
             }
             bucketIndex++;
         }
         for (var i=0; i<flyouts.length;i↵

```



```

519         ↪ ++) {
520         var f = flyouts[i];
521         var targ = selected[i];
522         f.unsetTarget();
523         f.hide();
524         if (targ) {
525             f.setTarget(targ, ↪
526                 ↪ grid.getFirstGrid().get ↪
527                 ↪ (0));
528         }
529     }
530     // var delayPoint = setInterval(↪
531     ↪ function () {
532     //     clearInterval(delayPoint);
533     for (var i=0; i<flyouts.length; ↪
534         ↪ i++) {
535         var f = flyouts[i];
536         var targ = selected[i];
537         if (targ) {
538             f.show();
539         }
540     }
541     pointFlyouts();
542     // }, 50);
543     //trace(selected);
544 }
545
546 function unpointFlyouts() {
547     if (birchlabs.flyoutsOn) {

```

```

548         var flyouts = birchlabs.flyouts;
549         for (var i=0; i<flyouts.length; i ↪
550             ↪ ++) {
551             var f = flyouts[i];
552             f.unsetTarget();
553             f.hide();
554         }
555     }
556 }
557
558 function pointFlyouts() {
559     var grid = birchlabs.grid;
560     if (birchlabs.flyoutsOn) {
561         var flyouts = birchlabs.flyouts;
562         for (var i=0; i<flyouts.length; i ↪
563             ↪ ++) {
564             var f = flyouts[i];
565             f.point(grid.findPointerCoords());
566         }
567     }
568 }
569
570 function createGrid(root) {
571     var numpadMappings = ↪
572         ↪ birchlabs.numpadMappings;
573     var anticipatedHeight = $(root).↪

```

```

571     ↪ height();
572     var cellHeight = anticipatedHeight/3;
573     // do not display text smaller than ↪ 590
574     ↪ 12px ↪ 591
575     // which on a low-density monitor is ↪
576     ↪ 12pt
577
578     var parent = document.createElement ↪ 592
579     ↪ ('div'); ↪ 593
580     parent.className = 'chibiPoint_grid' ↪
581     ↪ chibiPoint_gridBorderOn'; ↪ 594
582
583     var drawBorders = false; ↪ 595
584     // if it's not too small, give it ↪ 596
585     ↪ borders ↪ 597
586     var rect = $(root).get(0).getBo ↪
587     ↪ undingClientRect();
588     if (rect.width>60 && rect.height ↪
589     ↪ >60) {
590         drawBorders = true; ↪ 598
591     }
592
593     for (var i = 0; i < 3; i++) {
594         var row = ↪ 599
595         ↪ document.createElement(' ↪
596         ↪ div');
597         row.className = 'chibiPoint_row' ↪ 600
598         ↪ '
599         for (var p = 0; p < 3; p++) {
600             var cell = ↪ 601

```

```

        ↪ document.createElement ↪
        ↪ ('div');
        if (drawBorders) {
            cell.className = " ↪
            ↪ chibiPoint_cell ↪
            ↪ chibiPoint_cellBorderOn ↪
            ↪ ";
        } else {
            cell.className = " ↪
            ↪ chibiPoint_cell";
        }

        if (cellHeight>12) {
            cell.innerHTML = Stri ↪
            ↪ ng.fromCharCode( ↪
            ↪ numpadMappings[(7-( ↪
            ↪ i*3-p)).toString() ↪
            ↪ ][0]).toUpperCase() ↪
            ↪ ;
            cell.style.fontSize ↪
            ↪ =Math.min(96, ↪
            ↪ cellHeight*0.5)+"px ↪
            ↪ "
            var trans = ↪
            ↪ document.createElement ↪
            ↪ ('div');
            trans.className = " ↪
            ↪ chibiPoint_backing" ↪
            ↪ ;
            cell.appendChild(trans);

```

```

602         }
603         row.appendChild(cell);
604     }
605     parent.appendChild(row);
606 }
607
608 //root.appendChild(parent);
609 $(root).append(parent);
610 }
611
612 function drill(num, grid, crosshairs) {
613     var Flyout = birchlabs.Flyout;
614
615     grid.getLastCells().text("");
616     grid.getLastGrid().addClass("chibiPoint_gridBorderOff");
617     grid.getLastCells().addClass("chibiPoint_cellBorderOff");
618     grid.getLastGrid().removeClass("chibiPoint_gridBorderOn");
619     grid.getLastCells().removeClass("chibiPoint_cellBorderOn");
620
621     var rowIndexed1 = 4-Math.ceil(num/3);
622     var columnIndexed1 = ((num-1)%3)+1;
623     //cell.innerHTML = 7-(i*3-p);
624
625     selectArray = [];
626     selectArray.push(grid.getLatestSelector());

```

```

627 selectArray.push(".chibiPoint_grid");
628 selectArray.push(".chibiPoint_row:↵
        ↵ nth-of-type("+rowIndexed1+")");
629 selectArray.push(".chibiPoint_cell:↵
        ↵ nth-of-type("+columnIndexed1+")");
630 grid.pushHistory(selectArray.join(""));
631
632 createGrid($(grid.getLatestSelector()
        ↵()).first());
633
634 // move crosshairs
635 crosshairs.updatePosition(grid);
636
637 highlightTarget();
638 Flyout.show();
639
640 }
641
642 function backup(grid, crosshairs) {
643     var numpadMappings = ↵
        ↵ birchlabs.numpadMappings;
644     var Flyout = birchlabs.Flyout;
645
646     if (gridIsInUse()) {
647
648         // remove a grid!
649         $(grid.getLatestSelector()).empty();
650         grid.getPreviousSelector();
651
652         //trace(grid.getLatestSelector());

```

```

653
654     var anticipatedHeight = $(
        ↪ grid.getLatestSelector()).
        ↪ height();
655     var cellHeight = anticipatedHeight
        ↪ /3;
656     grid.getLastRows().each(function(
        ↪ index) {
657         var i = index;
658         $(this).children().each(fun
        ↪ ction(index) {
659             var p = index;
660             if (cellHeight>12) {
661                 $(this).text(String.fro
        ↪ mCharCode(
        ↪ numpadMappings[(7-(i
        ↪ *3-p)).toString()
        ↪ ][0]).toUpperCase());
662
663                 var trans =
        ↪ document.createElement
        ↪ ('div');
664                 trans.className = "
        ↪ chibiPoint_backing";
665                 $(this).append(trans);
666             }
667         });
668     });
669
670     grid.getLastGrid().removeClass("

```

```

        ↪ chibiPoint_gridBorderOff");
        grid.getLastCells().removeClass("
        ↪ chibiPoint_cellBorderOff");
        grid.getLastGrid().addClass("ch
        ↪ ibiPoint_gridBorderOn");

        // if it's not too small, give it
        ↪ borders
        var rect = $(grid.getLatestSele
        ↪ ctor()).get(0).
        ↪ getBoundingClientRect();
        if (rect.width>60 && rect.height
        ↪ >60) {
            grid.getLastCells().addClass("
        ↪ chibiPoint_cellBorderOn");
        }

        unpointFlyouts();

        var exists = grid.getLastGrid().
        ↪ length>0;
        if (exists) {
            // move crosshairs
            crosshairs.updatePosition(grid);
            Flyout.show();
            highlightTarget();
        } else {
            closeGrid();
        }
    }
}

```

```

692 }
693
694 function clickOrFocus(element) {
695     evaler.clickOrFocus(element);
696     closeGrid();
697 }
698
699 function gridclick(grid) {
700     var coords = grid.findPointerCoords();
701     ↪ ;
702     var centerX = coords.centerX;
703     var centerY = coords.centerY;
704
705     var rootNodes = [window].concat(1 ↪
706     ↪ ookup.getRootNodes());
707     for (var i = 0; i < rootNodes.length; ↪
708     ↪ i++) {
709         var doc2 = rootNodes[i].document || ↪
710         ↪ rootNodes[i].contentDocument;
711         if (!doc2 || !doc2.body)
712             continue;
713
714         var element = $( ↪
715         ↪ doc2.elementFromPoint(centerX ↪
716         ↪ , centerY));
717
718         element.removeClass("chibiPoint ↪
719         ↪ _targeted");
720     }
721     element.addClass("chibiPoint_cluck");

```

```

715
716     var flash = setInterval(function() ↪
717     ↪ {
718         if (element.hasClass("chibiPo ↪
719         ↪ int_cluck")) {
720             element.removeClass("chibiP ↪
721             ↪ oint_cluck");
722         } else {
723             element.addClass("chibiPoint_cluck");
724         }
725     }, 100);
726
727     var delayUnclick = setInterval( ↪
728     ↪ function () {
729         clearInterval(flash);
730         clearInterval(delayUnclick);
731         element.removeClass("chibiP ↪
732         ↪ oint_cluck");
733         clickOrFocus(element.get(0));
734         pointFlyouts();
735     }, 500);
736     //element.get(0).click();
737
738 }
739
740 function shortcut() {
741     if (!birchlabs.evaluateTabMode) {
742         toggleGrid();

```

```

740     evaler.evaluatorIncrementKeys("↵
        ↵ Cmd_K");
741     }
742 }
743
744 function toggleGrid() {
745     if (!birchlabs.evaluateTabMode) {
746         var Flyout = birchlabs.Flyout;
747
748         var crosshairs = ↵
            ↵ birchlabs.crosshairs;
749         var container = ↵
            ↵ birchlabs.container;
750         var grid = birchlabs.grid;
751
752         document.activeElement.blur();
753         //document.documentElement.focus();
754
755         if (gridIsEmpty()) {
756             // need to make a grid
757             grid.initialize();
758             $(".chibiPoint_gridContainer").↵
                ↵ removeClass("↵
                ↵ chibiPoint_hiddenGridC↵
                ↵ ontainer");
759             createGrid(container);
760
761             //crosshairs = crosshairs || new ↵
                ↵ birchlabs.Crosshairs(↵
                ↵ document.documentElement, ↵

```

```

        ↵ grid);
        //birchlabs.crosshairs = ↵
        ↵ crosshairs;

        // in case crosshairs already↵
        ↵ instantiated
        crosshairs.updatePosition(grid);
        Flyout.show();
        highlightTarget();
        crosshairs.show();
    } else {
        // toggle grid off
        closeGrid();
    }
}

function closeGrid() {
    var Flyout = birchlabs.Flyout;

    var crosshairs = ↵
        ↵ birchlabs.crosshairs;
    var grid = birchlabs.grid;

    grid.initialize();
    $(grid.getLatestSelector()).empty();

    $(".chibiPoint_gridContainer").ad↵
        ↵ dClass("↵
        ↵ chibiPoint_hiddenGridContainer"↵

```

31

```

    ↪ );
786
787     unpointFlyouts();
788     Flyout.hide();
789     crosshairs.hide();
790 }
791
792 function gridIsEmpty() {
793     return $(".chibiPoint_gridContainer" ↪
    ↪ ).children().length == 0;
794 }
795
796 function gridIsInUse() {
797     //var grid = birchlabs.grid;
798     return $(".chibiPoint_grid").chil ↪
    ↪ dren().length > 0;
799 }
800
801     return { init: init,
802             createGrid: createGrid,
803             shortcut: shortcut};
804 });

```

0.6 ./src/Grid.js

```

1  define(["lib/jquery-2.1.0.min", "lib/ ↪
    ↪ within"], function(jq, within) {
2
3  window.birchlabs = window.birchlabs || {};
4  var birchlabs = window.birchlabs;

```

```

5
6  (function(){
7      var Grid = function() {
8          this.initialize();
9      }
10     var p = Grid.prototype;
11
12     p.initialize = function() {
13         this.selectHistory = [" ↪
    ↪ chibiPoint_gridContainer"];
14     };
15
16     p.getLatestSelector = function() {
17         return this.selectHistory[this.se ↪
    ↪ lectHistory.length-1];
18     }
19
20     p.getPreviousSelector = function() {
21         if (this.selectHistory.length>1) {
22             this.selectHistory.pop(this.sel ↪
    ↪ ectHistory.length-1);
23         }
24         return this.getLatestSelector();
25     }
26
27     p.pushHistory = function(input) {
28         this.selectHistory.push(input);
29     }
30
31     p.exists = function() {

```

```

32     return this.selectHistory.length>1;
33 }
34
35 p.getLastRows = function() {
36     return$(this.getLatestSelector()+".↵
    ↪ chibiPoint_row");
37 };
38
39 p.getLastCells = function() {
40     return$(this.getLatestSelector()+".↵
    ↪ chibiPoint_cell");
41 };
42
43 p.getLastGrid = function() {
44     return$(this.getLatestSelector()+".↵
    ↪ chibiPoint_grid").first();
45 };
46
47 p.getFirstGrid = function() {
48     return $(this.selectHistory[0]+".↵
    ↪ chibiPoint_grid").first();
49 };
50
51 p.findPointerCoords = function() {
52     var element = this.getLastGrid();
53     // maybe there are no grids made
54     if (element.get(0)) {
55         // click
56         //var offset =gridElement.offset();
57         var offset = element.get(0).getB↵

```

```

58         ↪ oundingClientRect();
59         //console.log($this.get(0));
60         //console.log($this.offset());
61         var width = element.outerWidth();
62         var height = element.outerHeight();
63
64         var centerX = offset.left +width / ↵
        ↪ 2;
65         var centerY = offset.top +height / ↵
        ↪ 2;
66
67         //console.log("x: "+centerX+" y:"+↵
        ↪ centerY);
68         return {centerX:centerX,↵
        ↪ centerY:centerY};
69     } else {
70         return null;
71     }
72 };
73
74 birchlabs.Grid = Grid;
75 })();
76
77 });

```

0.7 ./src/Crosshairs.js

```

1  define(["lib/jquery-2.1.0.min","lib/↵
    ↪ within"], function(jq, within) {

```



```

2
3 window.birchlabs = window.birchlabs||{};
4 var birchlabs = window.birchlabs;
5
6 (function() {
7     var Crosshairs = function(root,grid) {
8         this.initialize(root, grid);
9     };
10
11     var p = Crosshairs.prototype;
12
13     p.setX = function(x) {
14         this.x = x;
15
16         // convert to percent
17         var elem = this.grid.getFirstGrid();
18         var rect = elem.get(0).↵
19             ↵ getBoundingClientRect();
20         var pX = x*100/rect.width;
21         $(".chibiPoint_verticalHair").first↵
22             ↵ ().css({left:pX+"%"});
23     };
24
25     p.setY = function(y) {
26         this.y = y;
27
28         // convert to percent
29         var elem = this.grid.getFirstGrid();
30         var rect = elem.get(0).↵
31             ↵ getBoundingClientRect();

```

```

29     var pY = y*100/rect.height;
30
31     $(".chibiPoint_horizontalHair").f↵
32         ↵ first().css({top:pY+"%"});
33 };
34
35 p.show = function() {
36     this.hairs.className = "↵
37         ↵ chibiPoint_crosshairs↵
38         ↵ chibiPoint_shownCrosshairs";
39 };
40
41 p.hide = function() {
42     this.hairs.className = "↵
43         ↵ chibiPoint_crosshairs↵
44         ↵ chibiPoint_hiddenCrosshairs";
45     this.removeHighlights();
46 };
47
48 p.createCrosshairs = function(root) {
49     this.hairs = document.createElement('↵
50         ↵ div');
51     this.hairs.className = "↵
52         ↵ chibiPoint_crosshairs";
53
54     this.horizontalHair = ↵
55         ↵ document.createElement('div');
56     this.horizontalHair.className = "↵
57         ↵ chibiPoint_horizontalHair";
58
59     this.verticalHair = ↵

```

```

50     ↪ document.createElement('div');
    this.verticalHair.className = "↪
    ↪ chibiPoint_verticalHair";
51
52     $(this.hairs).append(this.horizon↪
    ↪ talHair);
53     $(this.hairs).append(this.vertica↪
    ↪ lHair);
54
55     $(root).append(this.hairs);
56 };
57
58 p.updatePosition = function() {
59     var coords = ↪
    ↪ this.grid.findPointerCoords();
60     if (coords) {
61         var centerX = coords.centerX;
62         var centerY = coords.centerY;
63
64         this.setX(centerX);
65         this.setY(centerY);
66     }
67 };
68
69 p.highlight = function(root) {
70     var coords = ↪
    ↪ this.grid.findPointerCoords();
71     if (coords) {
72         var centerX = coords.centerX;
73         var centerY = coords.centerY;

```

```

74
75     var element = $(↪
    ↪ root.elementFromPoint(centerX,↪
    ↪ centerY));
76
77     // unhighlight previous element
78     this.removeHighlights();
79     this.targetedElement = element;
80
81     element.addClass("chibiPoint_tar↪
    ↪ geted");
82
83     // doesn't seem to work :(
84     /*element.trigger('mouseover');
85     element.trigger('hover');
86     element.trigger('mouseenter');
87     element.mouseover();
88     element.hover();
89     element.mouseenter();*/
90     //element.get(0).mouseover();
91     //element.get(0).hover();
92
93     return element;
94 } else {
95     return null;
96 }
97 };
98
99 p.removeHighlights = function() {
100     if (this.targetedElement) {

```

```

101     this.targetedElement.removeClass(↵
        ↵ "chibiPoint_targeted");
102     this.targetedElement.removeClass(↵
        ↵ "chibiPoint_cluck");
103     }
104 };
105
106 p.initialize = function(root, grid) {
107     this.targetedElement = null;
108     this.createCrosshairs(root);
109     this.grid = grid;
110     this.updatePosition();
111     this.show();
112 };
113
114 birchlabs.Crosshairs = Crosshairs;
115 })();
116
117 });

```

0.8 ./src/Flyout.js

```

1  define(["lib/jquery-2.1.0.min", "lib/↵
    ↵ within", "evaluator"], function(jq, ↵
    ↵ within, evaler) {
2
3  window.birchlabs = window.birchlabs || {};
4  var birchlabs = window.birchlabs;
5
6  (function() {

```

```

7  var Flyout = function(root, label) {
8      this.initialize(root, label);
9  };
10
11  Flyout.latestId = 0;
12
13  Flyout.getNextId = function() {
14      return Flyout.latestId++;
15  };
16
17  Flyout.getContainer = function(root) {
18      return this.container || this.makeco↵
        ↵ ntainer(root);
19  };
20
21  Flyout.makecontainer = function(root) {
22      this.container = ↵
        ↵ document.createElement('div');
23      this.container.className = "↵
        ↵ flyoutContainer hiddenfC";
24
25      $(root).append(this.container);
26      return this.container;
27  };
28
29  var p = Flyout.prototype;
30
31  p.setX = function(x) {
32      var rect = Flyout.getContainer().↵
        ↵ getBoundingClientRect();

```

```

33     x = Math.min(Math.max(2, x), ↵
        ↵ rect.width-22);
34     this.x = x;
35
36     // convert to percent
37     var pX = x*100/rect.width;
38     $(this.flyout).css({left:pX+"%"});
39     //$(this.flyout).css({left:x});
40     //$(this.flyout).css({left:x});
41 };
42
43 p.setY = function(y) {
44     var rect = Flyout.getContainer().↵
        ↵ getBoundingClientRect();
45     y = Math.min(Math.max(2, y), ↵
        ↵ rect.height-22);
46     this.y = y;
47
48     // convert to percent
49     var pY = y*100/rect.height;
50     $(this.flyout).css({top:pY+"%"});
51     //$(this.flyout).css({left:x});
52 };
53
54 /*p.setY = function(y) {
55     this.y = y;
56
57     // convert to percent
58     var elem = this.grid.getFirstGrid();
59     var rect =elem.get(0).↵

```

60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79

```

        ↵ getBoundingClientRect();
        var pY = y*100/rect.height;

        $(".horizontalHair").first().css({↵
            ↵ top:pY+"%"});
        };*/
Flyout.show = function() {
    Flyout.getContainer().className = "↵
        ↵ chibiPoint_flyoutContainer ↵
        ↵ chibiPoint_shownfC";
    //this.hairs.className = "crosshairs ↵
        ↵ shownCrosshairs";
};
Flyout.hide = function() {
    Flyout.getContainer().className = "↵
        ↵ chibiPoint_flyoutContainer ↵
        ↵ chibiPoint_hiddenfC";
    //this.hairs.className = "crosshairs ↵
        ↵ hiddenCrosshairs";
};
p.hide = function() {
    $(this.aFlyout).addClass("chibiPoi↵
        ↵ nt_aFlyoutHidden");
    //$(this.aFlyout).removeClass("bir↵
        ↵ chAFlyoutShown");
};
p.show = function() {

```

```

80  $(this.aFlyout).removeClass("chibi ↵
    ↵ Point_aFlyoutHidden");
81  };
82
83  p.build = function(label) {
84      var container = Flyout.getContainer(↵
    ↵ this.root);
85
86      this.aFlyout = document.createElement ↵
    ↵ ('div');
87      this.aFlyout.className = "↵
    ↵ chibiPoint_aFlyout";
88
89      this.lineDiv = document.createElement ↵
    ↵ ('div');
90      this.lineDiv.className = "↵
    ↵ chibiPoint_lineDiv";
91      this.svg = document.createElement('↵
    ↵ svg');
92      this.svg.className = "↵
    ↵ chibiPoint_lineSvg"
93      this.svgLine = document.createElement ↵
    ↵ ('line');
94      this.svgLine.className = "↵
    ↵ chibiPoint_line↵
    ↵ chibiPoint_lineUnique"+↵
    ↵ this.unique;
95      this.svgLine.id = "↵
    ↵ chibiPoint_flyoutLine"+↵
    ↵ this.unique;

```

```

96  //svgLine.style="stroke:rgb(255,0,0);↵
    ↵ stroke-width:2";
97  $(this.svgLine).attr({x1:"0",
98                      y1:"0",
99                      x2:"200",
100                     y2:"200"});
101  //$(svgLine).attr({"data-bind":"↵
    ↵ attr: {x1:x1,y1:y1,x2:x2,y2:y2↵
    ↵ }"});
102
103  $(this.svg).append(this.svgLine);
104
105  $(this.lineDiv).append(this.svg);
106
107  this.flyout = document.createElement(↵
    ↵ 'div');
108  this.flyout.className = "↵
    ↵ chibiPoint_flyout↵
    ↵ chibiPoint_flyoutUnique"+↵
    ↵ this.unique;
109  this.flyout.innerHTML = String.fr↵
    ↵ omCharCode(label).toUpperCase()↵
    ↵ ;
110
111  $(this.aFlyout).append(this.flyout);
112  $(this.aFlyout).append(this.lineDiv);
113
114  $(container).append(this.aFlyout);
115
116  // refresh namespace

```

```

117     $(this.lineDiv).html($(this.lineDiv) ↵ 143
        ↵ .html());
118 };
119
120 /*p.updatePosition = function() { 144
121     var coords = ↵ 145
        ↵ this.grid.findPointerCoords(); 146
122     var centerX = coords.centerX; 147
123     var centerY = coords.centerY; 148
124
125     this.setX(centerX); 149
126     this.setY(centerY); 150
127 };*/ 151
128
129 // root required only if it's the first ↵ 152
        ↵ and you haven't done a ↵
        ↵ makeContainer yet
130 p.initialize = function(label, root) { 153
131     this.root = root; 154
132     this.unique = Flyout.getNextId(); 155
133     this.build(label); 156
134     this.setX(this.unique*20); 157
135     this.setY(this.unique*20);
136
137     //this.updatePosition(); 158
138     //this.show(); 159
139 }; 160
140
141 p.point = function(avoidCoords) { 161
142     if (this.target) { 162
        163

```

```

var rect = ↵
        ↵ this.target.getBoundingClientRect ↵
        ↵ ();
var left = rect.left;
var top = rect.top;
var width = rect.width;
var height = rect.height;

var me = $(Flyout.getContainer());

// adapted from within.js
var myRect = me.get(0). ↵
        ↵ getBoundingClientRect();
var res = !( (myRect.top > top+height) ↵
        ↵ || (myRect.top+myRect.height < ↵
        ↵ top) || (myRect.left > left+width ↵
        ↵ ) || (myRect.left+myRect.width < ↵
        ↵ left));

if(res) {
    this.show();
    // disperse away from grid, ↵
        ↵ assuming it exists
    var xdist = 0;
    var ydist = 0;
    var dist = 0;
    var angle = 0;
    if (avoidCoords) {
        xdist = avoidCoords.centerX- ↵
            ↵ rect.left;

```

```

164      ydist = avoidCoords.centerY-↵ 188
        ↵ rect.top; 189
165      dist =Math.sqrt(xdist*xdist+ydist↵ 190
        ↵ *ydist); 191
166      angle = Math.atan2(ydist, xdist); 192
167    } 193
168    var avoidDist = Math.min((1/(dist*↵ 194
        ↵ dist*dist))*1000000000, 60); 195
169  196
170    this.setX((rect.left-20)-Math.sin(↵ 197
        ↵ angle)*avoidDist); 198
171    this.setY((rect.top-20)+Math.cos(↵ 199
        ↵ angle)*avoidDist); 200
172    var theX = left; 201
173    var theY = top; 202
174  203
175    $("#"+this.svgLine.id).attr({x1:↵ 204
        ↵ this.x, 205
176    y1:this.y, 206
177    x2:theX, 207
178    y2:theY}); 208
179    this.paintTarget(); 209
180    } else { 210
181      this.hide(); 211
182      this.unpaintTarget(); 212
183    } 213
184  } else {
185    this.hide();
186  }
187  };

```

```

p.clickOrFocus = function(element) {
  evaler.clickOrFocus(element);
}

p.doClick = function() {
  if (this.target) {

    var element = $(this.target);
    $(this.target).addClass("chibiPoin↵
        ↵ t_cluck");

    var flash = setInterval(function() {
      if (element.hasClass("chibiPoint↵
        ↵ _cluck")) {
        element.removeClass("chibiPoin↵
        ↵ t_cluck");
      } else {
        element.addClass("chibiPoint_cluck");
      }
    }, 100);

    var toClick = this.clickOrFocus;
    var thatTarget = this.target;
    var delayUnclick =setInterval(↵
        ↵ function () {
      clearInterval(flash);
      clearInterval(delayUnclick);
      $(thatTarget).removeClass("chib↵

```

```

214         ↪ iPoint_cluck");
215         toClick(thatTarget);
216     }, 500);
217 };
218
219 p.paintTarget = function() {
220     if (this.target) {
221         $(this.target).addClass("chibiPoin↵
222             ↪ t_painted");
223         $(this.target).addClass("chibiPoin↵
224             ↪ t_paintedUnique"+this.unique);
225     }
226 };
227
228 p.unpaintTarget = function() {
229     if (this.target) {
230         $(this.target).removeClass("chibiP↵
231             ↪ oint_painted");
232         $(this.target).removeClass("chibiP↵
233             ↪ oint_paintedUnique"+this.unique)↵
234             ↪ ;
235     }
236 };
237
238 p.unsetTarget = function() {
239     this.unpaintTarget();
240     this.target = false;
241 };

```

```

238 p.getTarget = function() {
239     return this.target;
240 };
241
242 p.setTarget = function(element,↵
243     ↪ coordsyst) {
244     /*if (element == null) {
245         $("#"+this.svgLine.id).attr({x2:0,
246             ↪ y2:0});
247         return;
248     }*/
249     this.unsetTarget();
250     this.target = element;
251     //this.point();
252
253     /*var rect2 =↵
254         ↪ coordsyst.getBoundingClientRect()↵
255         ↪ ;
256     var pX = theX*100/rect2.width;
257     var pY = theY*100/rect2.height;*/
258
259     /*$(this.svgLine).attr({x2:theX,
260         ↪ y2:theY});
261     $(this.svgLine).attr({id:"hey"});
262     //$(this.lineDiv).html($(this.lin↵
263         ↪ eDiv).html());
264     $(this.lineDiv).html($(this.lineDiv).↵
265         ↪ html());*/
266 };

```



```
263 | birchlabs.Flyout = Flyout;
264 | }();
265 |
266 | }|;
```