neural-sketch-groups.sty
Grouping support for neural-sketch

Vincenzo Buono

March 26, 2025

implementation   ¡¡¡ ¡*package¿

```
1 \ExplSyntaxOn
2
3 % ----------------------------------------------------------------------
4 % 1) Global Data Structures
5 % ----------------------------------------------------------------------
6 \prop_new:N \l_nsk_group_unknown_prop
7 \tl_set:Nn \l_nsk_group_refnode_id_tl {__nsk_phantom_refnode}
8
9
10 % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
11 % Positioning parsing and smart anchor detections
12 % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ <<<
13 \cs_new_protected_nopar:Npn \nsk__group_parse_pos:
14 {
15
16 % "below right=0cm and 0cm of basics"
17 \seq_set_split:Nnx \l_tmpa_seq {=}{\l_nsk_group_pos_tl}
18 \seq_show:N \l_tmpa_seq
19
20 % dira/b
21 \tl_clear_new:N \l_nsk_group_ps_dira_tl
22 \tl_clear_new:N \l_nsk_group_ps_dirb_tl
23
24 % If there's at least two items, we can inspect the first part
25 %     (the directions) to pick a multi-word anchor.
26 \int_compare:nNnT { \seq_count:N \l_tmpa_seq } > {1}
27 {
28 % \l_tmpa_tl := "below right" (for example)
29 \tl_set:Nx \l_tmpa_lx_tl { \seq_item:Nn \l_tmpa_seq {1} }
30 % 0cm and 0cm of basics.<..>
31 \tl_set:Nx \l_tmpa_rx_tl { \seq_item:Nn \l_tmpa_seq {2} }
32
33 % split lx side ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ <<<
34 % split below right
35 \seq_set_split:Nnx \l_tmpa_ps_seq {~} { \tl_use:N \l_tmpa_lx_tl }
36
37 \int_compare:nNnT { \seq_count:N \l_tmpa_ps_seq } = {1}
38 {
39 \tl_set:Nx \l_nsk_group_ps_dira_tl { \seq_item:Nn \l_tmpa_ps_seq {1} }
```

```
40 }
41
42 \int_compare:nNnT { \seq_count:N \l_tmpa_ps_seq } = {2}
43 {
44 \tl_set:Nx \l_nsk_group_ps_dira_tl { \seq_item:Nn \l_tmpa_ps_seq {1} }
45 \tl_set:Nx \l_nsk_group_ps_dirb_tl { \seq_item:Nn \l_tmpa_ps_seq {2} }
46 }
47 }
48
49 % prepend proper alignmet ~~~~~~~~~~~~~~~~~~~~~~ <<<
50 \str_if_in:NnTF \l_nsk_group_pos_tl {.}
51 {}
52 {
53 % not given --
54 \tl_clear_new:N \l_anchor_tl
55 \str_case:Vn \l_nsk_group_ps_dira_tl
56 {
57  {above}{\tl_set:Nn \l_anchor_tl {north}}
58   {below}{\tl_set:Nn \l_anchor_tl {south}}
59   {left}{\tl_set:Nn \l_anchor_tl {west}}
60   {right}{\tl_set:Nn \l_anchor_tl {east}}
61 }
62
63 \tl_put_right:Nx \l_nsk_group_pos_tl
64 {.\l_anchor_tl}
65 }
66 }
67
68 % -----------------------------------------------------------------------
69 % last-pos logic
70 % -----------------------------------------------------------------------
71 \cs_new_protected_nopar:Npn \nsk__process_last_pos_for_group:
72 {
73 % Only proceed if the user gave 'last-pos'
74 \tl_if_blank:VF \l_nsk_group_last_pos_tl
75 {
76  \seq_if_empty:NTF \g_nsk_block_id_history_seq
77  {
78   \msg_warning:nn {nsk}{No-previous-block-for-last-pos}
79  }
80  {
81   % exec last block id retrieval ~~~~~~~~~~~~~~~~ <<<
82   % block is already registered, so get prev
83   \nsk__internal_get_nth_block_id:n{2}
84   \tl_set:Nx \l_tmpa_tl { \g_nsk_block_last_requested_block_id }
85
86   \str_if_in:NnTF \l_nsk_group_last_pos_tl {=}
87   {
88    % case (a) ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ <<<
89    % There's at least one '='
90    % Next see if "of" is in there
91    % e.g. "below=0cm"
92    % => add " of <blockID>"
93    \tl_set:Nx \l_nsk_group_pos_tl
```

3

```
 94      {
 95        \exp_not:V \l_nsk_group_last_pos_tl\space of~\exp_not:V \l_tmpa_tl
 96      }
 97     }
 98     {
 99      % case (b) ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ <<<
100      % There's NO '=' in the user's string
101      % e.g. "below right"
102      % => produce "below right=of <blockID>"
103      \tl_set:Nx \l_nsk_group_pos_tl
104      {
105        \exp_not:V \l_nsk_group_last_pos_tl =of~\exp_not:V \l_tmpa_tl
106      }
107     }
108    }
109 }
110 }
111
112 \cs_new_protected_nopar:Npn \nsk__detect_anchor_pos:
113 {
114  \nsk__group_parse_pos:
115
116  % vertical  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ <<<
117  \tl_if_blank:VTF \l_nsk_group_pos_tl
118  {}
119  {
120   \tl_if_blank:VT \l_nsk_pos_align_tl
121   {
122    \tl_if_blank:VF \l_nsk_group_ps_dira_tl
123    {
124     \str_case:Vn \l_nsk_group_ps_dira_tl {
125      {above}{\tl_set:Nn \l_nsk_pos_align_tl {south~}}
126       {below}{\tl_set:Nn \l_nsk_pos_align_tl {north~}}
127       {right}{\tl_set:Nn \l_nsk_pos_align_tl {west~}}
128       {left}{\tl_set:Nn \l_nsk_pos_align_tl {east~}}
129    }
130    }
131
132    \tl_if_blank:VF \l_nsk_group_ps_dirb_tl
133    {
134     \str_case:Vn \l_nsk_group_ps_dirb_tl {
135      {right}{\tl_put_right:Nn \l_nsk_pos_align_tl {west}}
136       {left}{\tl_put_right:Nn \l_nsk_pos_align_tl {east}}
137    }
138    }
139   }
140  }
141 }
142
143
144 %----------------------------------------------------------
145 % Group
146 % logical grouping
147 %----------------------------------------------------------
```

```
148 \keys_define:nn {nsk / group}
149 {
150   % bounds id -------------------------------------------------
151   bounds-name .tl_set:N = \l_nsk_group_bounds_name_tl,
152   bounds-name .initial:n = {__nsk_container_bounds},
153   bounds-name .default:n = {__nsk_container_bounds},
154
155   % natural positioning using positioning library
156   pos .tl_set:N = \l_nsk_group_pos_tl,
157   pos .initial:n = { },
158   pos .default:n = { },
159
160   last-pos .tl_set:N = \l_nsk_group_last_pos_tl,
161   last-pos .initial:n = { },
162
163   % pos-alignment (edge flush) ~~~~~~~~~~~~~~~~~~ <<<
164   pos-align .tl_set:N = \l_nsk_pos_align_tl,
165   pos-align .default:n = { },
166   pos-align .initial:n = { },
167
168   shift-x .fp_set:N = \l_nsk_group_x_fp,
169   shift-x .initial:n = {0},
170   shift-x .default:n = {0},
171
172   shift-y .fp_set:N = \l_nsk_group_y_fp,
173   shift-y .initial:n = {0},
174   shift-y .default:n = {0},
175
176
177   rotate .fp_set:N = \l_nsk_group_rotate_fp,
178   rotate .initial:n = {0},
179   rotate .default:n = {0},
180
181   scale .fp_set:N = \l_nsk_group_scale_fp,
182   scale .initial:n = {1},
183   scale .default:n = {1},
184
185   % forward unknown props to scope
186   unknown .code:n =
187   {
188     % Store unknown key in the property list, using the content of \l_keys_key_tl as a string
189     \prop_put:Nxx \l_nsk_group_unknown_prop
190     { \tl_use:N \l_keys_key_tl }
191     { #1 }
192   },
193 }
194
195 %-----------------------------------------------------------------
196 % () Internal: build the final TikZ style as a comma-separated list
197 %    We'll store it in \l_tmpa_tl.
198 %-----------------------------------------------------------------
199
200 \cs_new_protected_nopar:Npn \nsk__build_group_style:
201 {
```

5

```
202  % styles
203  \tl_clear_new:N \l_tmpa_tl
204
205  \tl_put_right:Nx \l_tmpa_tl
206  {local~bounding~box=\l_nsk_group_bounds_name_tl, }
207
208  \tl_if_blank:VF \l_nsk_group_pos_tl
209  {
210   \tl_put_right:Nx \l_tmpa_tl
211   { shift={(\l_nsk_group_refnode_id_tl)}, }
212
213  }
214
215  \tl_if_blank:VF \l_nsk_pos_align_tl
216  {
217   \tl_put_right:Nx \l_tmpa_tl
218   {anchor={\l_nsk_pos_align_tl}, }
219
220   % use pic workaround / wait for buffer write
221   \tl_put_right:Nn \l_tmpa_tl
222   {scope~anchor, }
223  }
224
225
226  % rotate=...
227  \tl_put_right:Nx \l_tmpa_tl { rotate=\fp_use:N \l_nsk_group_rotate_fp, }
228
229  % scale=...
230  \tl_if_eq:nnF {\fp_to_decimal:N \l_nsk_group_scale_fp}{1.0}
231  { \tl_put_right:Nx \l_tmpa_tl { scale=\fp_use:N \l_nsk_group_scale_fp, } }
232
233  % forward unknown keys as-is
234  \prop_map_inline:Nn \l_nsk_group_unknown_prop
235  {
236   \tl_put_right:Nn \l_tmpa_tl
237   {
238    ##1 = ##2,
239   }
240  }
241
242  }
243
244  %-------------------------------------------------------------------
245  % 4) Internal: draw the scope with the built style
246  %    #1 -> the style token list (passed by value)
247  %    #2 -> user content
248  %-------------------------------------------------------------------
249  \cs_new_protected:Npn \nsk__draw_group_scope_aux:nnn #1 #2 #3
250  {
251  % phantom node ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ <<<
252  \tl_if_blank:VF \l_nsk_group_pos_tl
253  {
254   \node[
255    #1,
```

```
256    draw=none,
257    fill=none,
258    name=\l_nsk_group_refnode_id_tl,
259    anchor=center
260   ] {};
261 }
262
263
264 % actual draw ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ <<<
265 \begin{scope}[#2]
266  \pgftransformshift{
267   \pgfpoint{\fp_use:N \l_nsk_group_x_fp}{\fp_use:N \l_nsk_group_y_fp}
268  }
269  #3
270 \end{scope}
271 }
272
273 % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
274 % Variant Generation :V
275 % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ <<<
276 \cs_generate_variant:Nn \nsk__draw_group_scope_aux:nnn { VVn }
277
278 % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
279 % Public Interface
280 % - (user-facing)
281 % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ <<<
282 \NewDocumentCommand \nskGroup { O{} +m }
283 {
284
285  \keys_set:nn { nsk / group } {#1}
286
287
288  \nsk__process_last_pos_for_group:
289
290  % maybe detect anchor pos
291  \nsk__detect_anchor_pos:
292
293  % Build a comma-separated style in \l_tmpa_tl
294  \nsk__build_group_style:
295
296
297  % draw
298  \nsk__draw_group_scope_aux:VVn \l_nsk_group_pos_tl \l_tmpa_tl {#2}
299 }
300
301 \NewDocumentCommand \nskOnlayer { m +m }
302 {
303  \begin{pgfonlayer}{#1}
304   #2
305  \end{pgfonlayer}
306 }
307
308 \NewDocumentCommand \nskOnbg { +m }
309 {
```

```
310  \nskOnlayer{background}{#1}
311 }
312
313 \NewDocumentCommand \nskOnfg { +m }
314 {
315  \nskOnlayer{foreground}{#1}
316 }
317
318
319 \ExplSyntaxOff
```

¡/*package¿