

neural-sketch-containers.dtx

neural-sketch-containers.sty

Container logic for neural-sketch

Vincenzo Buono

March 26, 2025

```
implementation                                     ||| i*packagej
1 \ExplSyntaxOn
2
3
4 % ~~~~~
5 % Custom pgf layers
6 % ~~~~~ <<<
7 \pgfdeclarelayer{background}
8 \pgfdeclarelayer{interim}
9 \pgfdeclarelayer{foreground}
10 % layer registration ~~~~~ <<<
11 \pgfsetlayers{background,interim,main,foreground}
12
13
14 % ~~~~~
15 % Pass-through
16 % store unknown keys to pass-through
17 % ~~~~~ <<<
18 \prop_new:N \l_nsk_container_unknown_prop
19
20 % ~~~~~
21 % Position Parser
22 % ~~~~~ <<<
23 \dim_new:N \l__nsk_tmp_dim
24 \cs_new_protected_nopar:Npn \nsk__container_parse_pos_value:nn #1#2
25 {
26 % #1 = the user's "right=1cm of X" or "above=2mm of Y" ...
27 % #2 = the container's padding dimension (e.g. 3mm)
28
29 % Save the raw pos string to \l_nsk_container_pos_tl
30 \tl_set:Nx \l_nsk_container_pos_tl {#1}
31
32 % Split at '=' (only if the user wrote e.g. "right=1cm of X").
33 \seq_set_split:Nnn \l_tmpa_seq {=}{#1}
34
35 % If we have exactly 2 items, item 1 is "right" and item 2 is "1cm of X".
36 \int_compare:nNtT { \seq_count:N \l_tmpa_seq } = {2}
37 {
38 % The direction (left/right/above/below)
```

```

39 \tl_set:Nx \l_tmpa_tl { \seq_item:Nn \l_tmpa_seq {1} }
40 \tl_trim_spaces:N \l_tmpa_tl
41
42 % The distance+node portion
43 \tl_set:Nx \l_tmpb_tl { \seq_item:Nn \l_tmpa_seq {2} }
44 \tl_trim_spaces:N \l_tmpb_tl
45
46 \tl_show:N \l_tmpa_tl
47 \tl_show:N \l_tmpb_tl
48
49 % split at " of "
50 \seq_set_split:Nnx \l_tmpb_seq {of}{\l_tmpb_tl}
51
52
53 \int_compare:nNnT { \seq_count:N \l_tmpb_seq } = {2}
54 {
55 % item 1 = "1cm", item 2 = "X"
56 \tl_set:Nx \l_tmpc_tl { \seq_item:Nn \l_tmpb_seq {1} }
57 \tl_trim_spaces:N \l_tmpc_tl
58
59 \tl_if_empty:NTF \l_tmpc_tl
60 {
61 % No distance was provided; use default distance.
62 \tl_set:Nn \l_tmpc_tl {2cm}
63 }
64 {}
65
66 \tl_set:Nx \l_tmpd_tl { \seq_item:Nn \l_tmpb_seq {2} }
67 \tl_trim_spaces:N \l_tmpd_tl
68 \tl_show:N \l_tmpc_tl
69 \tl_show:N \l_tmpd_tl
70
71
72 % Convert "1cm" to a dimension variable
73 \dim_set:Nn \l__nsk_tmp_dim {\tl_use:N \l_tmpc_tl}
74 \dim_show:N \l__nsk_tmp_dim
75 % Now add the container's padding
76 \dim_add:Nn \l__nsk_tmp_dim { #2 }
77 \dim_add:Nn \l__nsk_tmp_dim { #2 }
78 \dim_add:Nn \l__nsk_tmp_dim { #2 }
79 \dim_show:N \l__nsk_tmp_dim
80
81 % ~~~~~~
82 % String build
83 % Rebuild "right=(dist+pad) of X"
84 % ~~~~~~ <<<
85 \tl_set:Nx \l_nsk_container_pos_tl
86 {
87 \tl_use:N \l_tmpa_tl
88 = \dim_use:N \l__nsk_tmp_dim
89 \c_space_tl of \c_space_tl
90 \tl_use:N \l_tmpd_tl
91 }
92 }

```

```

93 }
94 }
95
96
97 % ~~~~~
98 % Main nsk / container Primitive
99 % ~~~~~ <<<
100 \keys_define:nn {nsk / container}
101 {
102   % ref-id ~~~~~ <<<
103   % todo: passed through
104   id .tl_set:N = \l_nsk_block_id_tl,
105   id .initial:n = { },
106
107   % styles ~~~~~ <<<
108   border-type .code:n =
109   {
110     \tl_if_eq:nnTF {#1}{none}
111     {
112       \tl_set:Nn \l_nsk_container_border_type_tl {draw=none}
113     }
114     {
115       \tl_set:Nn \l_nsk_container_border_type_tl {#1}
116     }
117   },
118   border-type .initial:n = {solid},
119   border-type .default:n = {solid},
120
121   border-color .tl_set:N = \l_nsk_container_border_color_tl,
122   border-color .initial:V = \g__nsk_style_container_border_color_tl,
123   border-color .default:V = \g__nsk_style_container_border_color_tl,
124
125   fill .tl_set:N = \l_nsk_container_fill_tl,
126   fill .initial:V = \g__nsk_style_container_fill_tl,
127   fill .default:V = \g__nsk_style_container_fill_tl,
128
129   padding .dim_set:N = \l_nsk_container_padding_tl,
130   padding .initial:V = \g__nsk_style_container_padding_dim,
131   padding .default:V = \g__nsk_style_container_padding_dim,
132
133   shift-x .fp_set:N = \l_nsk_container_x_fp,
134   shift-x .initial:V = \g__nsk_style_container_x_fp,
135   shift-x .default:V = \g__nsk_style_container_x_fp,
136
137   shift-y .fp_set:N = \l_nsk_container_y_fp,
138   shift-y .initial:V = \g__nsk_style_container_y_fp,
139   shift-y .default:V = \g__nsk_style_container_y_fp,
140
141   % natural positioning using positioning library
142   pos .code:n =
143   {
144     \nsk__container_parse_pos_value:nn {#1} {\l_nsk_container_padding_tl}
145   },
146

```

```

147 pos .initial:n = { },
148 pos .default:n = { },
149
150 rotate .fp_set:N = \l_nsk_container_rotate_fp,
151 rotate .initial:V = \g__nsk_style_container_rotate_fp,
152 rotate .default:V = \g__nsk_style_container_rotate_fp,
153
154 scale .fp_set:N = \l_nsk_container_scale_fp,
155 scale .initial:V = \g__nsk_style_container_scale_fp,
156 scale .default:V = \g__nsk_style_container_scale_fp,
157
158 group-opts .code:n =
159 {
160   % #1 is exactly what is inside { ... }, including any commas.
161   % store it literally into \l_nsk_block_extra_style_tl
162   \tl_set:Nn \l_nsk_group_extra_style_tl {#1}
163 },
164 group-opts .initial:n = {},
165 group-opts .default:n = {},
166
167 % -- The special pass-through key for arbitrary options.
168 % forward unknown props to nsblocks
169 unknown .code:n =
170 {
171   % Store unknown key in the property list, using the content of \l_keys_key_tl as a string
172   \prop_put:Nxx \l_nsk_container_unknown_prop
173   { \tl_use:N \l_keys_key_tl }
174   { #1 }
175 },
176
177 }
178
179 % ~~~~~
180 % Public Interface
181 % - (user-facing)
182 % ~~~~~ <<<
183 \NewDocumentCommand \nskContainer { O{} +m }
184 {
185   \group_begin:
186
187   % (a) Clear the property list so it doesn't retain any old unknown keys
188   \prop_clear_new:N \l_nsk_container_unknown_prop
189
190   % (b) Parse recognized container keys + store unknown keys
191   \keys_set:nn {nsk / container}{#1}
192
193   % (c) Now call \nskGroup, passing the same #1 so that shift-x, rotate, etc.
194   %     get recognized by nsk / group (or remain unknown for that domain).
195
196   \tl_show:N \l_nsk_container_pos_tl
197   \tl_set:Nx \l_tmpx_tl
198   {
199     shift-x=\l_nsk_container_x_fp,
200     shift-y=\l_nsk_container_y_fp,

```

```

201 rotate=\l_nsk_container_rotate_fp,
202 scale=\l_nsk_container_scale_fp,
203 pos={\l_nsk_container_pos_tl},
204 }
205
206 \tl_show:N \l_tmpx_tl
207
208 % % pass-through ~~~~~ <<<
209 \tl_if_empty:VTF \l_nsk_block_extra_style_tl
210 { }
211 {
212 \tl_put_right:Nx \l_tmpx_tl {\l_nsk_block_extra_style_tl,}
213 }
214
215
216 \nsk__expand_group_aux:Vn \l_tmpx_tl {#2}
217
218 % (d) Build an \nskBlock that fits the bounding box, plus any unknown keys
219 % stored in \l_nsk_container_unknown_prop. We'll do a small helper:
220 \nsk__container_draw_block:
221
222 \group_end:
223
224 }
225
226 \cs_new_protected:Npn \nsk__expand_group_aux:nn #1 #2
227 {
228 \nskGroup[#1]
229 {
230 #2
231 }
232 }
233
234
235 % block expander ~~~~~ <<<
236 \cs_new_protected_nopar:Npn \nsk__expand_block_aux:n #1
237 {
238
239 \begin{pgfonlayer}{background}
240 \nskBlock [
241 % this are computed by bounding box
242 width=0pt, height=0pt,
243 shadow=false,
244 % text-north=\nskBlockID,
245 #1,
246 ]
247 \end{pgfonlayer}
248 }
249
250 % ~~~~~
251 % Variant Generation :V
252 % ~~~~~ <<<
253 \cs_generate_variant:Nn \nsk__expand_block_aux:n { V }
254 \cs_generate_variant:Nn \nsk__expand_group_aux:nn { Vn }

```

```

255 \cs_generate_variant:Nn \nskBlockFromList:V { V }
256
257 % ~~~~~
258 % Main draw
259 % ~~~~~ <<<
260 \cs_new_protected_nopar:Npn \nsk__container_draw_block:
261 {
262 % We'll build a token list of block options
263 \tl_clear_new:N \l_tmpa_tl
264
265 % put recognized container keys:
266 \tl_put_right:No \l_tmpa_tl { type=rectangle, }
267 \tl_put_right:No \l_tmpa_tl { fill=\l_nsk_container_fill_tl, }
268 \tl_put_right:No \l_tmpa_tl { border-color=\l_nsk_container_border_color_tl, }
269 % use tikz-ops keys
270 \tl_put_right:Nx \l_tmpa_tl
271 { tikz-opts={fit=(\l_nsk_group_bounds_name_tl), inner\space sep=\l_nsk_container_padding_tl,
272   \l_nsk_container_border_type_tl}, }
273
274
275 % forward unknown keys as-is
276 \prop_map_inline:Nn \l_nsk_container_unknown_prop
277 {
278   \tl_put_right:Nn \l_tmpa_tl
279   {
280     {##1} = {##2},
281   }
282 }
283
284 \nsk__expand_block_aux:V \l_tmpa_tl
285 }
286
287 \ExplSyntaxOff
\end{package}

```