



# neural-sketch.sty (main code)

Vincenzo Buono

March 22, 2025

```
implementation                                     \ii i*package;
1
2 % external ~~~~~ <<<
3 \RequirePackage{tikz}
4 \RequirePackage{expl3}
5 \usetikzlibrary{
6   calc, fit, intersections, spath3,
7   shapes.geometric,
8   shapes.symbols,
9   shapes.misc,
10  arrows.meta,
11  positioning,
12  arrows.meta,
13  shadows,
14  shadings,
15  patterns,
16  decorations.markings,
17  backgrounds,
18  tikzmark,
19  math,
20 }
21
22
23 \ExplSyntaxOn
24
25 % ~~~~~
26 % Pkg opts
27 % ~~~~~ <<<
28 \seq_new:N \l__nsk_pkgopts_opts_seq
29 \keys_define:nn { nsk / pkgopts }
30 {
31   unknown .code:n = {
32     \seq_put_right:Nn \l__nsk_pkgopts_opts_seq {##1~={##2},}
33   }
34 }
35 \ProcessKeysOptions { nsk / pkgopts }
36
37 \bool_new:N \g__nsk_dark_mode_bool
38 \keys_set:nn { nsk / style } { \seq_use:Nn \l__nsk_pkgopts_opts_seq {, } }
39
40 \ExplSyntaxOff
41
```

```

42 % internal ~~~~~ <<<
43 \RequirePackage{neural-sketch-colors}
44 \RequirePackage{neural-sketch-styles}
45 \RequirePackage{neural-sketch-loader}
46
47 \ExplSyntaxOn
48 % \bool_if:NTF \g__nsk_dark_mode_bool
49 % {
50 %   \definecolor{ c__nsk_dark_bg_clr } {HTML}{1d2021}
51 %   \definecolor{ c__nsk_principal } {HTML}{ebdbb2}
52 %   \pagecolor{c__nsk_dark_bg_clr}
53 %   \colorlet{nskBg}{c__nsk_dark_bg_clr}
54 %   \colorlet{nskFg}{c__nsk_principal}
55 % }
56 % {
57 %   \definecolor{ c__nsk_principal } {HTML}{000000}
58 %   \colorlet{nskBg}{c__nsk_principal}
59 %   \colorlet{nskFg}{black}
60 % }
61
62 % -\ExplSyntaxOn
63 % -\bool_new:N \g__nsk_dark_mode_bool
64 % -\ifnsk@dark_mode
65 % -\bool_set_true:N \g__nsk_dark_mode_bool
66 % -\definecolor{ c__nsk_dark_bg_clr } {HTML}{1d2021}
67 % -\definecolor{ c__nsk_principal } {HTML}{ebdbb2}
68 % -\pagecolor{c__nsk_dark_bg_clr}
69 % -\colorlet{nskBg}{c__nsk_dark_bg_clr}
70 % -\colorlet{nskFg}{c__nsk_principal}
71 % -\color{c__nsk_principal}
72 % -\else
73 % -\bool_set_false:N \g__nsk_dark_mode_bool
74 % -\fi
75 % -\ExplSyntaxOff
76
77 % ~~~~~
78 % Global props -
79 % to track auto-increment counters
80 % ~~~~~ <<<
81 \prop_new:N \g_nsk_block_counters_prop
82 % ~~~~~
83 % Global seq -
84 % to track block-id reference counters
85 % ~~~~~ <<<
86 \seq_new:N \g_nsk_block_id_history_seq
87
88 % ~~~~~
89 % Main nsk / Block Primitive
90 % ~~~~~ <<<
91 \keys_define:nn {nsk / block}
92 {
93 % block type ~~~~~ <<<
94 type .tl_set:N = \l_nsk_block_type_tl,
95 type .initial:n = {rectangle},

```

```

96 type .default:n = {rectangle},
97
98 % ref-id ~~~~~ <<<
99 id .tl_set:N = \l_nsk_block_id_tl,
100 id .initial:n = { },
101
102 % coords ~~~~~ <<<
103 x .fp_set:N = \l_nsk_block_x_fp,
104 x .initial:n = {0},
105 x .default:n = {0},
106 y .fp_set:N = \l_nsk_block_y_fp,
107 y .initial:n = {0},
108 y .default:n = {0},
109
110 anchor .tl_set:N = \l_nsk_block_anchor_tl,
111 anchor .default:n = {},
112 anchor .initial:n = {},
113
114 % natural positioning using positioning library
115 pos .tl_set:N = \l_nsk_block_pos_tl,
116 pos .initial:n = { },
117
118 % last-nat pos ~~~~~ <<<
119 last-pos .tl_set:N = \l_nsk_block_last_pos_tl,
120 last-pos .initial:n = { },
121 last-pos .default:n = { },
122
123 % (works like last-pos but without error)
124 last-pos-s .tl_set:N = \l_nsk_block_last_pos_s_tl,
125 last-pos-s .initial:n = { },
126 last-pos-s .default:n = { },
127
128 % dimensions ~~~~~ <<<
129 width .dim_set:N = \l_nsk_block_width_dim,
130 width .initial:V = \g__nsk_style_block_width_dim,
131 width .default:V = \g__nsk_style_block_width_dim,
132
133 height .dim_set:N = \l_nsk_block_height_dim,
134 height .default:V = \g__nsk_style_block_height_dim,
135 height .initial:V = \g__nsk_style_block_height_dim,
136
137
138 % ~~~~~
139 % Inner (regular) border
140 % ~~~~~ <<<
141 border-type .code:n =
142 {
143   \tl_if_eq:nnTF {#1}{none}
144   {
145     \tl_set:Nn \l_nsk_block_border_type_tl {draw=none}
146   }
147   {
148     \tl_set:Nn \l_nsk_block_border_type_tl {#1}
149   }

```

```

150
151   },
152
153   border-type .initial:n = {solid},
154   border-type .default:n = {solid},
155
156   border-color .tl_set:N = \l_nsk_block_border_color_tl,
157   border-color .initial:V = \g__nsk_style_block_border_color_tl,
158   border-color .default:V = \g__nsk_style_block_border_color_tl,
159
160   border-radius .dim_set:N = \l_nsk_block_border_radius_dim,
161   border-radius .initial:V = \g__nsk_style_block_border_radius_dim,
162   border-radius .default:V = \g__nsk_style_block_border_radius_dim,
163
164   border-size .dim_set:N = \l_nsk_block_border_size_dim,
165   border-size .initial:V = \g__nsk_style_block_border_size_dim,
166   border-size .default:V = \g__nsk_style_block_border_size_dim,
167
168   % ~~~~~~
169   % Outer (halo) border
170   % ~~~~~~ <<<
171
172   outer-border-type .code:n =
173   {
174     \tl_if_eq:nnTF {#1}{none}
175     {
176       \tl_set:Nn \l_nsk_block_outer_border_type_tl {draw=none}
177     }
178     {
179       \tl_set:Nn \l_nsk_block_outer_border_type_tl {#1}
180     }
181   }
182   },
183
184   outer-border-type .initial:n = {none},
185   outer-border-type .default:n = {none},
186
187   outer-border-color .tl_set:N = \l_nsk_block_outer_border_color_tl,
188   outer-border-color .initial:n = { nskMainAccent },
189   outer-border-color .default:n = { nskMainAccent },
190
191   outer-border-radius .dim_set:N = \l_nsk_block_outer_border_radius_dim,
192   outer-border-radius .initial:V = \g__nsk_style_block_border_radius_dim,
193   outer-border-radius .default:V = \g__nsk_style_block_border_radius_dim,
194
195   outer-border-size .dim_set:N = \l_nsk_block_outer_border_size_dim,
196   outer-border-size .initial:n = {7mm},
197   outer-border-size .default:n = {7mm},
198
199   % styles ~~~~~~ <<<
200   fill .tl_set:N = \l_nsk_block_fill_tl,
201   fill .initial:V = \g__nsk_style_block_fill_tl,
202   fill .default:V = \g__nsk_style_block_fill_tl,
203

```

```

204 % shadow ~~~~~ <<<
205 shadow .bool_set:N = \l_nsk_block_shadow_bool,
206 shadow .initial:n = {true},
207 shadow .default:n = {true},
208
209 % importance ~~~~~ <<<
210 importance .fp_set:N = \l_nsk_block_importance_fp,
211 importance .initial:n = {1.0},
212 importance .default:n = {1.0},
213
214 % -- The special pass-through key for arbitrary TikZ options.
215 tikz-opts .code:n =
216 {
217   % #1 is exactly what is inside { ... }, including any commas.
218   % store it literally into \l_nsk_block_extra_style_tl
219   \tl_set:Nn \l_nsk_block_extra_style_tl {#1}
220 },
221 tikz-opts .initial:n = {},
222 tikz-opts .default:n = {},
223
224 % ~~~~~
225 % Text Anchoring
226 % ~~~~~ <<<
227 %-----
228 % text-north
229 %-----
230 text-north .tl_set:N = \l_nsk_text_north_tl,
231 text-north .initial:n = { },
232
233 text-north-loc .tl_set:N = \l_nsk_text_north_loc_tl,
234 text-north-loc .initial:V = \g__nsk_style_text_north_loc_tl,
235
236 text-north-style .tl_set:N = \l_nsk_text_north_style_tl,
237 text-north-style .initial:n = { },
238
239 %-----
240 % text-south
241 %-----
242 text-south .tl_set:N = \l_nsk_text_south_tl,
243 text-south .initial:n = { },
244
245 text-south-loc .tl_set:N = \l_nsk_text_south_loc_tl,
246 text-south-loc .initial:V = \g__nsk_style_text_south_loc_tl,
247
248 text-south-style .tl_set:N = \l_nsk_text_south_style_tl,
249 text-south-style .initial:n = { },
250
251 %-----
252 % text-east
253 %-----
254 text-east .tl_set:N = \l_nsk_text_east_tl,
255 text-east .initial:n = { },
256
257 text-east-loc .tl_set:N = \l_nsk_text_east_loc_tl,

```

```

258 text-east-loc .initial:V = \g__nsk_style_text_east_loc_tl,
259
260 text-east-style .tl_set:N = \l_nsk_text_east_style_tl,
261 text-east-style .initial:n = { },
262
263 %-----
264 % text-west
265 %-----
266 text-west .tl_set:N = \l_nsk_text_west_tl,
267 text-west .initial:n = { },
268
269 text-west-loc .tl_set:N = \l_nsk_text_west_loc_tl,
270 text-west-loc .initial:V = \g__nsk_style_text_west_loc_tl,
271
272 text-west-style .tl_set:N = \l_nsk_text_west_style_tl,
273 text-west-style .initial:n = { },
274
275 %-----
276 % text-center
277 %-----
278 text-center .tl_set:N = \l_nsk_text_center_tl,
279 text-center .initial:n = { },
280
281 text-center-loc .tl_set:N = \l_nsk_text_center_loc_tl,
282 text-center-loc .initial:V = \g__nsk_style_text_center_loc_tl,
283
284 text-center-style .tl_set:N = \l_nsk_text_center_style_tl,
285 text-center-style .initial:n = { },
286
287 % ~~~~~~
288 % Embedding
289 % ~~~~~~ <<<
290 embed .tl_set:N = \l_nsk_embed_tl,
291 embed .default:n = {},
292 embed .initial:n = {},
293
294 embed-gfx .tl_set:N = \l_nsk_embed_gfx_tl,
295 embed-gfx .default:n = {},
296 embed-gfx .initial:n = {},
297
298 embed-padding .dim_set:N = \l_nsk_block_embed_padding_dim,
299 embed-padding .default:n = \g__nsk_style_block_embed_padding_dim,
300 embed-padding .initial:n = \g__nsk_style_block_embed_padding_dim,
301
302 embed-border-radius .dim_set:N = \l_nsk_block_embed_border_radius_dim,
303 embed-border-radius .initial:V = \g__nsk_style_block_embed_border_radius_dim,
304 embed-border-radius .default:V = \g__nsk_style_block_embed_border_radius_dim,
305
306 % ~~~~~~
307 % Patterns
308 % ~~~~~~ <<<
309 pattern .tl_set:N = \l_nsk_block_pattern_tl,
310 pattern .initial:n = { },
311 pattern .default:n = { },

```

```

312
313 pattern-color .tl_set:N = \l_nsk_block_pattern_color_tl,
314 pattern-color .initial:n = { },
315 pattern-color .default:n = { },
316 }
317
318 % ~~~~~
319 % Auto-generate an ID if user does not supply one
320 % ~~~~~ <<<
321 \cs_new_protected_nopar:Npn \nsk__maybe_set_id:
322 {
323   \tl_if_blank:VTF \l_nsk_block_id_tl
324   { \nsk__assign_autogenerated_id: }
325   { }
326 }
327
328 \cs_new_protected_nopar:Npn \nsk__assign_autogenerated_id:
329 {
330   % If shape key not in the counters prop yet, init to 0
331   \prop_if_in:NVF \g_nsk_block_counters_prop \l_nsk_block_type_tl
332   {
333     % () Use \prop_gput to store globally:
334     \prop_gput:NVn \g_nsk_block_counters_prop \l_nsk_block_type_tl { 0 }
335   }
336
337   % Retrieve current count
338   \prop_get:NVN \g_nsk_block_counters_prop \l_nsk_block_type_tl \l_tmpa_tl
339
340   % Convert to int and increment
341   \int_set:Nn \l_tmpa_int { \l_tmpa_tl + 1 }
342
343   % Update the counters property globally:
344   \prop_gput:NVx \g_nsk_block_counters_prop \l_nsk_block_type_tl { \int_use:N \l_tmpa_int }
345
346   % Construct "diamond1", "diamond2", "rectangle3", etc.
347   \tl_set:No \l_nsk_block_id_tl
348   {
349     \tl_use:N \l_nsk_block_type_tl
350     \int_use:N \l_tmpa_int
351   }
352 }
353
354 % ~~~~~
355 % Border size importance
356 % ~~~~~ <<<
357 \cs_new_protected_nopar:Npn \nsk__maybe_scale_border_by_importance:
358 {
359   % \fp_if_gt:NNT \l_nsk_block_importance_fp \l_nsk_block_importance_fp
360   \fp_compare:nT { \l_nsk_block_importance_fp > 1.0 }
361   {
362     % Convert the current border-size dimension into a float:
363     \fp_set:Nn \l_tmpa_fp { \l_nsk_block_border_size_dim }
364
365     % Multiply by the importance factor:

```



```

366 \fp_set:Nn \l_tmpa_fp { \l_tmpa_fp * \l_nsk_block_importance_fp }
367
368 % Update the actual dimension variable:
369 \dim_set:Nn \l_nsk_block_border_size_dim { \fp_to_dim:N \l_tmpa_fp }
370 }
371 }
372
373 % ~~~~~~
374 % Variant Generation :V
375 % ~~~~~~ <<<
376 \cs_generate_variant:Nn \nsk__render_one_text_aux:nnnn { nVVV }
377 \cs_generate_variant:Nn \nsk__draw_block_aux:n { V }
378
379
380 % ~~~~~~
381 % Text Rendering
382 % ~~~~~~ <<<
383 \cs_new:Npn \nsk_compute_text_node_anchor:nn #1 #2
384 {
385   \str_case:nn {#1}
386   {
387     {north}{\str_case:nnF {#2}{ {west}{south~west} {east}{south~east} }}{south}}
388     {south}{\str_case:nnF {#2}{ {west}{north~west} {east}{north~east} }}{north}}
389     {east}{\str_case:nnF {#2}{ {north}{north~west} {south}{south~west} }}{west}}
390     {west}{\str_case:nnF {#2}{ {north}{north~east} {south}{south~east} }}{east}}
391     {center}{center}
392   }
393 }
394
395 \cs_new_protected_nopar:Npn \nsk__render_one_text_aux:nnnn #1 #2 #3 #4
396 {
397   % Compute the node anchor based on the block's side (#1)
398   % and the extra alignment provided (#3)
399   \tl_set:Nx \l_tmpa_tl { \nsk_compute_text_node_anchor:nn {#1} {#3} }
400   \tl_show:N \l_tmpa_tl
401   % Draw the node at the block's anchor coordinate (e.g. (blockID.north))
402   % but with the computed node anchor.
403   \tl_if_blank:nTF {#3}
404   {
405     \draw[shift=(\l_nsk_block_id_tl.#1)]
406     node[anchor=\l_tmpa_tl, #4] {#2};
407   }{
408     % If extra alignment (#3) is "north" or "south", swap the order.
409     \str_if_eq:nnTF {#3}{north}
410     {
411       \draw[shift=(\l_nsk_block_id_tl.#3~#1)]
412       node[anchor=\l_tmpa_tl, #4] {#2};
413     }
414     {
415       \str_if_eq:nnTF {#3}{south}
416       {
417         \draw[shift=(\l_nsk_block_id_tl.#3~#1)]
418         node[anchor=\l_tmpa_tl, #4] {#2};
419       }

```

```

420 {
421   \draw[shift=(\l_nsk_block_id_tl.#1~#3)]
422     node[anchor=\l_tmpa_tl, #4] {#2};
423 }
424 }
425 }
426 }
427
428 \cs_new_protected_nopar:Npn \nsk__render_one_text:nnnn #1 #2 #3 #4
429 {
430   % #1 = the actual anchor name in TikZ (e.g., "north", "south", "center", etc.)
431   % #2 = text content
432   % #3 = location/placement (e.g., "above left", "right", etc.)
433   % #4 = style (e.g., "color=red, font=\bfseries")
434   % If the user did not provide text (#2), do nothing
435   \tl_if_blank:VTF {#2}
436   {
437     % do nothing
438   }
439   {
440     % Otherwise, call the :VVVV variant
441     \nsk__render_one_text_aux:nVVV {#1} {#2} {#3} {#4}
442   }
443 }
444
445 \cs_new_protected_nopar:Npn \nsk__render_all_text:
446 {
447   \nsk__render_one_text:nnnn {north}
448   { \l_nsk_text_north_tl }
449   { \l_nsk_text_north_loc_tl }
450   { \l_nsk_text_north_style_tl }
451
452   \nsk__render_one_text:nnnn {south}
453   { \l_nsk_text_south_tl }
454   { \l_nsk_text_south_loc_tl }
455   { \l_nsk_text_south_style_tl }
456
457   \nsk__render_one_text:nnnn {east}
458   { \l_nsk_text_east_tl }
459   { \l_nsk_text_east_loc_tl }
460   { \l_nsk_text_east_style_tl }
461
462   \nsk__render_one_text:nnnn {west}
463   { \l_nsk_text_west_tl }
464   { \l_nsk_text_west_loc_tl }
465   { \l_nsk_text_west_style_tl }
466
467   \nsk__render_one_text:nnnn {center}
468   { \l_nsk_text_center_tl }
469   { \l_nsk_text_center_loc_tl }
470   { \l_nsk_text_center_style_tl }
471 }
472
473 % Internal debug/log block ~~~~~<<<

```

```

474 \cs_new_protected_nopar:Npn \nsk__block_debug:
475 {
476   \iow_term:x {type      = \tl_use:N \l_nsk_block_type_tl}
477   \iow_term:x {x         = \fp_use:N \l_nsk_block_x_fp}
478   \iow_term:x {y         = \fp_use:N \l_nsk_block_y_fp}
479   \iow_term:x {width     = \dim_use:N \l_nsk_block_width_dim}
480   \iow_term:x {height    = \dim_use:N \l_nsk_block_height_dim}
481 }
482
483
484 % ~~~~~
485 % Last relative pos
486 % Automagically relative positioning to last previous block
487 % ~~~~~ <<<
488 \cs_new_protected_nopar:Npn \nsk__process_last_pos:
489 {
490   % Only proceed if the user gave 'last-pos'
491   \tl_if_blank:VF \l_nsk_block_last_pos_tl
492   {
493     \seq_if_empty:NTF \g_nsk_block_id_history_seq
494     {
495       \msg_warning:nn {nsk}{No-previous-block-for-last-pos}
496     }
497     {
498       % exec last block id retrieval ~~~~~ <<<
499       % block is already registered, so get prev
500       \nsk__internal_get_nth_block_id:n{2}
501       \tl_set:Nx \l_tmpa_tl { \g_nsk_block_last_requested_block_id }
502
503       \str_if_in:NnTF \l_nsk_block_last_pos_tl {=}
504       {
505         % case (a) ~~~~~ <<<
506         % There's at least one '='
507         % Next see if "of" is in there
508         % e.g. "below=0cm"
509         % => add " of <blockID>"
510         \tl_set:Nx \l_nsk_block_pos_tl
511         {
512           \exp_not:V \l_nsk_block_last_pos_tl\space of~\exp_not:V \l_tmpa_tl
513         }
514       }
515       {
516         % case (b) ~~~~~ <<<
517         % There's NO '=' in the user's string
518         % e.g. "below right"
519         % => produce "below right=of <blockID>"
520         \tl_set:Nx \l_nsk_block_pos_tl
521         {
522           \exp_not:V \l_nsk_block_last_pos_tl =of~\exp_not:V \l_tmpa_tl
523         }
524       }
525     }
526   }
527 }

```

```

528
529 \cs_new_protected_nopar:Npn \nsk__process_last_pos_s:
530 {
531   % Only proceed if the user gave a nonblank last-pos-x value:
532   \tl_if_blank:VF \l_nsk_block_last_pos_s_tl
533   {
534     % Check that there is more than one block in the history.
535     \int_compare:nNnTF {\seq_count:N \g_nsk_block_id_history_seq} > {1}
536     {
537       % There is a previous block. Retrieve it without calling the error-triggering routine.
538       \int_set:Nn \l_tmpa_int { \seq_count:N \g_nsk_block_id_history_seq }
539       \tl_set:Nn \l_tmpa_tl
540       {
541         \seq_item:Nn \g_nsk_block_id_history_seq
542         { \int_eval:n { \l_tmpa_int - 1 } }
543       }
544       % Process the key's value as in the original routine:
545       \str_if_in:NnTF \l_nsk_block_last_pos_s_tl {=}
546       {
547         \tl_set:Nx \l_nsk_block_pos_tl
548         {
549           \exp_not:V \l_nsk_block_last_pos_s_tl\space of~\exp_not:V \l_tmpa_tl
550         }
551       }
552       {
553         \tl_set:Nx \l_nsk_block_pos_tl
554         {
555           \exp_not:V \l_nsk_block_last_pos_s_tl =of~\exp_not:V \l_tmpa_tl
556         }
557       }
558     }
559     {
560       % There is no previous block: do nothing silently.
561     }
562   }
563 }
564 % ~~~~~~
565 % Block drawing [Aux]
566 % ~~~~~~ <<<
567 \cs_new_protected_nopar:Npn \nsk__draw_block_aux:n #1
568 {
569
570   \node [
571     #1,
572   ] {};
573
574   % embedding ~~~~~~ <<<
575   \tl_if_blank:VF {\l_nsk_embed_tl}
576   {
577     \dim_set:Nn \l_tmpa_dim { \l_nsk_block_width_dim - 2\l_nsk_block_embed_padding_dim}
578     \dim_set:Nn \l_tmpb_dim { \l_nsk_block_height_dim - 2\l_nsk_block_embed_padding_dim }
579
580     \begin{scope}
581       \pgftransformshift{\pgfpointanchor{\tl_use:N \l_nsk_block_id_tl}{center}}

```

```

582 \pgfset{minimum\space width=\l_tmpa_dim, minimum\space height=\l_tmpb_dim}
583 \pgfsetcornersarced{\pgfpoint{\l_nsk_block_embed_border_radius_dim}{\l_nsk_block_embed_bor
584 \pgfnode{\l_nsk_block_type_tl}{center}{}{nodename}{\pgfusepath{stroke,clip}}
585
586 % raw embedding ~~~~~ <<<
587 \tl_use:N \l_nsk_embed_tl
588 \end{scope}
589
590 }
591
592 \tl_if_blank:VF {\l_nsk_embed_gfx_tl}
593 {
594 \dim_set:Nn \l_tmpa_dim { \l_nsk_block_width_dim - 2\l_nsk_block_embed_padding_dim}
595 \dim_set:Nn \l_tmpb_dim { \l_nsk_block_height_dim - 2\l_nsk_block_embed_padding_dim }
596
597 \begin{scope}
598 \pgftransformshift{\pgfpointanchor{\tl_use:N \l_nsk_block_id_tl}{center}}
599 \pgfset{minimum\space width=\l_tmpa_dim, minimum\space height=\l_tmpb_dim}
600 \pgfsetcornersarced{\pgfpoint{\l_nsk_block_embed_border_radius_dim}{\l_nsk_block_embed_bor
601 \pgfnode{\l_nsk_block_type_tl}{center}{}{nodename}{\pgfusepath{stroke,clip}}
602
603 % gfx node ~~~~~ <<<
604 \node[]
605 {\includegraphics[width=\l_nsk_block_width_dim, height=\l_nsk_block_height_dim]{\tl_use:N \l_nsk_embed_gfx_tl}}
606 \end{scope}
607
608 }
609
610
611 }
612
613 % ~~~~~
614 % Pattern Aliasing resolution
615 % ~~~~~ <<<
616 \cs_new_protected_nopar:Npn \nsk__resolve_pattern_alias:
617 {
618 % If \l_nsk_block_pattern_tl is empty, do nothing
619 \tl_if_blank:VTF \l_nsk_block_pattern_tl
620 { }
621 {
622 % Check if the user's pattern key is in the alias property
623 \prop_if_in:NVTF \g_nsk_pattern_aliases_prop \l_nsk_block_pattern_tl
624 {
625 % It's an alias => retrieve the real pattern
626 \prop_get:NVN \g_nsk_pattern_aliases_prop \l_nsk_block_pattern_tl \l_tmpa_tl
627 \tl_set:NV \l_nsk_block_pattern_tl \l_tmpa_tl
628 }
629 {
630 % Not an alias => user gave a real pattern or a custom name => do nothing
631 }
632 }
633 }
634
635

```

```

636 \cs_new_protected_nopar:Npn \nsk__build_block_style:
637 {
638   % Build the style token list incrementally:
639   \tl_clear_new:N \l_tmps_tl
640
641   \tl_set:Nn \l_tmp_pre_acts {}
642   \tl_set:Nn \l_tmp_post_acts {}
643
644   % extras ~~~~~ <<<
645   \nsk__resolve_pattern_alias:
646
647   % Possibly scale border-size according to importance:
648   \nsk__maybe_scale_border_by_importance:
649
650   % id-ref -----
651   \tl_put_right:No \l_tmps_tl { name=\l_nsk_block_id_tl,}
652
653   % {x,y} pos -----
654   \tl_if_blank:VTF \l_nsk_block_pos_tl
655   {
656     % -- pos is blank => place the node using x,y
657     \tl_put_right:Nx \l_tmps_tl
658     { at={(\fp_use:N \l_nsk_block_x_fp,\fp_use:N \l_nsk_block_y_fp)}, }
659   }
660   {
661     % -- pos is given => place the node by raw TikZ positioning
662     % i.e., [right=2cm of otherNode]
663     \tl_put_right:No \l_tmps_tl { \l_nsk_block_pos_tl, }
664   }
665
666   \tl_if_blank:VF \l_nsk_block_anchor_tl
667   {
668     \tl_put_right:No \l_tmps_tl {anchor=\l_nsk_block_anchor_tl,}
669   }
670
671   % shape-type ~~~~~ <<<
672   \tl_put_right:No \l_tmps_tl {shape=\l_nsk_block_type_tl,}
673   % dims ~~~~~ <<<
674   \tl_put_right:No \l_tmps_tl {minimum\space width=\l_nsk_block_width_dim,}
675   \tl_put_right:No \l_tmps_tl {minimum\space height=\l_nsk_block_height_dim,}
676   % borders ~~~~~ <<<
677   \tl_put_right:No \l_tmps_tl {draw=\l_nsk_block_border_color_tl,}
678   \tl_put_right:No \l_tmps_tl {\l_nsk_block_border_type_tl,}
679   \tl_put_right:No \l_tmps_tl {rounded\space corners=\l_nsk_block_border_radius_dim,}
680   \tl_put_right:No \l_tmps_tl {line\space width=\l_nsk_block_border_size_dim,}
681   % outer border ~~~~~ <<<
682   \tl_put_right:No \l_tmp_pre_acts {draw=\l_nsk_block_outer_border_color_tl,}
683   \tl_put_right:No \l_tmp_pre_acts {\l_nsk_block_outer_border_type_tl,}
684   \tl_put_right:No \l_tmp_pre_acts {rounded\space corners=\l_nsk_block_outer_border_radius_dim,}
685   \tl_put_right:No \l_tmp_pre_acts {line\space width=\l_nsk_block_outer_border_size_dim,}
686
687   % fill ~~~~~ <<<
688   \tl_put_right:No \l_tmps_tl {fill=\l_nsk_block_fill_tl,}
689

```

```

690 % shadow ~~~~~ <<<
691 \bool_if:NT \l_nsk_block_shadow_bool
692 {
693   \tl_put_right:No \l_tmps_tl {nskShadow, }
694 }
695 % patterns ~~~~~ <<<
696 \tl_if_blank:VTF \l_nsk_block_pattern_tl
697 { }
698 {
699   % something like:
700   %   postaction={pattern=..., pattern color=...}
701   \tl_clear_new:N \l_tmpb_tl
702   \tl_put_right:Nx \l_tmpb_tl { pattern=\l_nsk_block_pattern_tl, }
703   \tl_if_blank:VTF \l_nsk_block_pattern_color_tl
704   { }
705   {
706     \tl_put_right:Nx \l_tmpb_tl { pattern\space color=\l_nsk_block_pattern_color_tl, }
707   }
708
709   % append to post-actions
710   \tl_put_right:Nx \l_tmp_post_acts {\l_tmpb_tl}
711 }
712
713
714 % pass-through ~~~~~ <<<
715 \tl_if_empty:VTF \l_nsk_block_extra_style_tl
716 { }
717 {
718   \tl_put_right:NV \l_tmps_tl {\l_nsk_block_extra_style_tl,}
719 }
720
721
722 % add any pre-actions ~~~~~ <<<
723 \tl_put_right:Nx \l_tmps_tl { preaction={\tl_use:N \l_tmp_pre_acts}, }
724 % add any post-actions ~~~~~ <<<
725 \tl_put_right:Nx \l_tmps_tl { postaction={\tl_use:N \l_tmp_post_acts}, }
726
727 % Now call function which takes
728 % params by value: --
729 \nsk__draw_block_aux:V
730 \l_tmps_tl
731 }
732
733
734 % ~~~~~
735 % Ref-Counter history
736 % ~~~~~ <<<
737 \cs_new_protected_nopar:Npn \nsk__store_block_in_history:
738 {
739   % Push the current block's ID into the global sequence
740   \seq_gput_right:Nx \g_nsk_block_id_history_seq \l_nsk_block_id_tl
741 }
742
743 \cs_new_protected_nopar:Npn \nsk__internal_get_nth_block_id:n #1

```

```

744 {
745 % #1 => integer offset n (nth-last)
746 % We will reference the global \g_nsk_block_id_history_seq inside.
747
748
749 \seq_if_empty:NTF \g_nsk_block_id_history_seq
750 {
751   \tl_set:Nn \l_tmpa_tl {ERROR-NoBlocks}
752   \GenericError{}{No block in the history!}{}{%
753     Attempted~to~reference~\string\nskBlockIDLast[#1],~but~no~blocks~exist.%
754   }
755 }
756 {
757   \int_compare:nNnTF {#1} > { \seq_count:N \g_nsk_block_id_history_seq }
758   {
759     \tl_set:Nn \l_tmpa_tl {ERROR-BadIndex}
760     \GenericError{}{Block~index~out~of~bounds~}{}{%
761       \string\nskBlockIDLast[#1] exceeds the number of existing blocks.%
762     }
763   }
764   {
765     % We want the item at \((total - n + 1)\).
766     \tl_set:Nn \l_tmpa_tl
767     {
768       \seq_item:Nn
769       \g_nsk_block_id_history_seq
770       { \int_eval:n{ \seq_count:N \g_nsk_block_id_history_seq - #1 + 1 } }
771     }
772   }
773 }
774
775
776 % Return it by expanding the token list \l_tmpa_tl
777 % \tl_use:N \l_tmpa_tl
778 \tl_gset:Nx \g_nsk_block_last_requested_block_id {\l_tmpa_tl}
779
780 % \seq_show:N \g_nsk_block_id_history_seq
781 % \tl_show:N \g_nsk_block_last_requested_block_id
782 }
783
784
785 % ~~~~~~
786 % Main block draw
787 % ~~~~~~ <<<
788 \cs_new_protected_nopar:Npn \nsk__draw_block:
789 {
790 % 1. draw the main block shape
791 \nsk__build_block_style:
792
793 % 2. draw the text anchorings
794 \nsk__render_all_text:
795 }
796
797 % ~~~~~~

```



```

798 % Main nsk / figure Primitive
799 % ~~~~~ <<<
800
801 \keys_define:nn { nsk / figure }
802 {
803   center .bool_set:N = \l_nsk_figure_center_bool,
804   center .initial:n = { false },
805   center .default:n = { true },
806
807   % pass-through ~~~~~ <<<
808   unknown .code:n =
809   {
810     \prop_put:NVn \l_nsk_figure_unknown_prop { \l_keys_key_tl } { #1 }
811   },
812 }
813
814 % ~~~~~
815 % Public Interface
816 % - (user-facing)
817 % ~~~~~ <<<
818 \NewDocumentCommand \nskBlock { 0{ } }
819 {
820   \group_begin:
821
822   % # 1 -> k=v list
823   \keys_set:nn {nsk / block } {#1}
824
825   % possibly autogenerate the id
826   \nsk__maybe_set_id:
827
828   % store id in history
829   \nsk__store_block_in_history:
830
831
832   % process last-pos -> set cpos if needed
833   % if last-pos-s is given, use it (without error); otherwise use last-pos
834   \tl_if_blank:VTF \l_nsk_block_last_pos_s_tl
835   { \nsk__process_last_pos: }
836   { \nsk__process_last_pos_s: }
837
838   \nsk__block_debug:
839
840   % actually draw the block
841   \nsk__draw_block:
842
843   \group_end:
844 }
845
846 \NewDocumentEnvironment{nskFigure}{0{}}
847 {
848   \prop_gclear:N \g_nsk_block_counters_prop
849   % (b) Clear the unknown-props container
850   \prop_clear:N \l_nsk_figure_unknown_prop
851

```

```

852 % (c) Parse the user-provided options
853 \keys_set:nn { nsk / figure } { #1 }
854
855 % (d) Convert the unknown key=value pairs into a comma list
856 \tl_clear_new:N \l_nsk_figure_opts_tl
857 \prop_map_inline:Nn \l_nsk_figure_unknown_prop
858 {
859   \tl_put_right:Nx \l_nsk_figure_opts_tl { ##1 = { ##2 }, }
860 }
861
862 % (e) If center=true, wrap the tikzpicture in a center environment
863 \bool_if:NTF \l_nsk_figure_center_bool
864 {
865   \begin{center}
866   \begin{tikzpicture}[\tl_use:N \l_nsk_figure_opts_tl]
867   }
868 {
869   \begin{tikzpicture}[\tl_use:N \l_nsk_figure_opts_tl]
870   }
871 }{
872   \end{tikzpicture}
873   \bool_if:NT \l_nsk_figure_center_bool
874   {
875     \end{center}
876   }
877 }
878
879 % ~~~~~
880 % Miscs Utils
881 % ~~~~~ <<<
882 \NewDocumentCommand \nskBlockID {}
883 {
884   \tl_use:N \l_nsk_block_id_tl
885 }
886
887 % reference counting ~~~~~ <<<
888 % \nskBlockIDLast[1] -> "the nth last block ID"
889 \NewDocumentCommand \nskBlockIDLast { 0{1} }
890 {
891   \nsk__internal_get_nth_block_id:n {#1}
892   \tl_use:N \g_nsk_block_last_requested_block_id
893 }
894
895 \NewDocumentCommand \nskLastBlockID {} {
896   % "Alias" for the single last block (i.e., \nskBlockIDLast[])
897   % block is already registered, so get prev ->2
898   \nskBlockIDLast[2]
899 }
900
901
902 \ExplSyntaxOff
903
904 i/package;

```