neural-sketch-render.sty

Conditional multi-type code render for neural-sketch

Vincenzo Buono

March 26, 2025

1 Introduction

This file provides conditional code rendering in a switch-like interface implementation ⟨⟨⟨ ¡*package¿

```
1 \ExplSyntaxOn
2
3
4 % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
5 % Internal variables -
6 % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ <<<
7 \tl_clear_new:N \l__nsk_switch_type_tl
8 \tl_clear_new:N \l__nsk_switch_operand_tl
9 \bool_new:N \l__nsk_switch_matched_bool
10
11 % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
12 % Main nsk / render Primitive
13 % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ <<<
14 \keys_define:nn { nsk / render }
15 {
16  type .choice:,
17  type / string .code:n = { \tl_set:Nn \l__nsk_switch_type_tl { string } },
18  type / int    .code:n = { \tl_set:Nn \l__nsk_switch_type_tl { int } },
19  type / float  .code:n = { \tl_set:Nn \l__nsk_switch_type_tl { float } },
20  type / bool   .code:n = { \tl_set:Nn \l__nsk_switch_type_tl { bool } },
21  type .initial:n = { string },
22 }
23
24
25 % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
26 % Public Interface
27 % - (user-facing)
28 % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ <<<
29 \NewDocumentEnvironment{nskSwitch}{ O{} m }
30 {
31  \group_begin:
32  \keys_set:nn { nsk / render } {#1}
33  \tl_set:Nx \l__nsk_switch_operand_tl {#2}
```

```
34  \bool_set_false:N \l__nsk_switch_matched_bool
35 }
36 { \group_end: }
37
38 % ˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜ <<<
39 % If no previous case has matched, compares the switch operand with the
40 % provided case value (according to the chosen type) and, if they match,
41 % sets the flag and executes the code in a group.
42 \NewDocumentCommand{\nskCase}{ d[] +m }
43 {
44 \bool_if:NTF \l__nsk_switch_matched_bool
45 { } % already matched: do nothing
46 {
47  \tl_show:N \l__nsk_switch_operand_tl
48  \str_if_eq:VnTF \l__nsk_switch_type_tl { string }
49  {
50   \str_if_eq:VnTF \l__nsk_switch_operand_tl  {#1}
51   { \bool_set_true:N \l__nsk_switch_matched_bool \group_begin: #2 \group_end: }
52   { }
53  }
54  {
55   \str_if_eq:VnTF \l__nsk_switch_type_tl { int }
56   { \int_compare:nTF
57    { \int_eval:n { \l__nsk_switch_operand_tl } = \int_eval:n {#1} }
58    { \bool_set_true:N \l__nsk_switch_matched_bool \group_begin: #2 \group_end: }
59    { } }
60   {
61    \str_if_eq:VnTF \l__nsk_switch_type_tl { float }
62    { \fp_compare:nTF
63     { \fp_eval:n { \l__nsk_switch_operand_tl } == \fp_eval:n {#1} }
64     { \bool_set_true:N \l__nsk_switch_matched_bool \group_begin: #2 \group_end: }
65     { } }
66    {
67     \str_if_eq:VnTF \l__nsk_switch_type_tl { bool }
68     { \str_if_eq:nnTF { \l__nsk_switch_operand_tl } {#1}
69      { \bool_set_true:N \l__nsk_switch_matched_bool \group_begin: #2 \group_end: }
70      { } }
71     { % fallback: treat as string
72      \str_if_eq:nnTF { \l__nsk_switch_operand_tl } {#1}
73      {
74       \bool_set_true:N \l__nsk_switch_matched_bool \group_begin: #2 \group_end:
75      }
76      { } } } }
77   }
78  }
79 }
80
81 % ˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜˜ <<<
82 % If no case has matched, execute the default code.
83 \NewDocumentCommand{\nskDefault}{ +m }
84 {
85 \bool_if:NF \l__nsk_switch_matched_bool
86 {
87  \group_begin: #1 \group_end:
```

```
88   }
89 }
90
91 \ExplSyntaxOff
92
```
⟨/package⟩