Package ‘shinyKnobs’

February 13, 2020

Title A Collection of Knob Inputs for ‘shiny’
Version 0.1.3
Description A collection of highly configurable, touch-enabled knob input controls for 'shiny'. These components can be styled to fit in perfectly in any app, and allow users to set precise values through many input modalities. Users can touch-and-drag, click-and-drag, scroll their mouse wheel, double click, or use keyboard input.
License MIT + file LICENSE
Encoding UTF-8
LazyData true
RoxygenNote 7.0.1
Imports shiny, htmltools
URL https://github.com/cotepat/shinyKnobs
BugReports https://github.com/cotepat/shinyKnobs/issues
NeedsCompilation no
Author Patrice Cote [aut, cre],
Chris Johnson [aut, cph] (precision-inputs)
Maintainer Patrice Cote <pat@patricecote.com>
Repository CRAN
Date/Publication 2020-02-13 09:30:03 UTC

R topics documented:
touchGripKnobInput .......................................................... 2
touchKnobInput ............................................................. 4

Index 7
touchGripKnobInput

**Description**

Highly configurable, touch-enabled grip dial knob input for Shiny

**Usage**

```r
touchGripKnobInput(
  inputId,
  label = NULL,
  labelPosition = "top",
  width = "auto",
  step = "any",
  min = 0,
  max = 1,
  initial = 0.5,
  color = "orange",
  guideTicks = 9,
  gripBumps = 5,
  gripExtrusion = 0.5,
  minRotation = 20,
  maxRotation = 340,
  dragResistance = 100,
  wheelResistance = 100,
  globalRatePolicy = NULL,
  globalRatePolicyDelay = 500
)
```

**Arguments**

- **inputId**  
The input slot that will be used to access the value.

- **label**  
Optional label for knob

- **labelPosition**  
Position of label (‘top’ or ‘bottom’)

- **width**  
Width of the knob as a percentage of the container element

- **step**  
The step amount for value changes, usually used with min and max parameters. Can be ‘any’ (no step)

- **min**  
The minimum input value.

- **max**  
The maximum input value.

- **initial**  
Initial value of the knob. Knob resets to this value when double-clicked.

- **color**  
The color to use for the focus indicator and indicator dot. You can set any *hex* color (ex : '#0F1BC3') or named color, choices are 'purple', 'blue', 'green', 'yellow', 'red', 'orange' or 'transparent' to disable.
touchGripKnobInput

- **guideTicks**: Number of tick marks on the outer guide ring (default = 9)
- **gripBumps**: Number of grip bumps that appear when interacting with the dial (default = 5)
- **gripExtrusion**: The degree to which the grips ‘cut’ into the dial when the user interacts with it. (range 0 to 1, default = 0.5)
- **minRotation**: The angle (in degrees) of rotation corresponding to the min value, relative to pointing straight down (0 degree) (default = pointing to the first guide tick mark)
- **maxRotation**: The angle (in degrees) of rotation corresponding to the max value, relative to pointing straight down (0 degree) (default = pointing to the last guide tick mark)
- **dragResistance**: The amount of resistance to value change on mouse/touch drag events. Higher value means more precision, and the user will have to drag farther to change the input’s value. (0 to 100)
- **wheelResistance**: The amount of resistance to value change on mouse wheel scroll. Higher value means more precision, and the mouse wheel will be less effective at changing the input’s value. (0 to 100)
- **globalRatePolicy**: Rate policy determining the behavior of output value events. NULL will output values in real-time, 'debounce' will output values once the knob stops moving, 'throttle' will output values while the knob is moving but only at a certain frequency (controlled with ratePolicyDelay). This setting will affect every touchKnobInput in the project.
- **globalRatePolicyDelay**: Delay to use when globalRatePolicy is set to 'throttle' or 'debounce'. This setting will affect every touchKnobInput in the project.

**Value**

Numeric value server-side.

**Examples**

```r
if (interactive()) {

library("shiny")
library("shinyKnobs")

ui <- fluidPage(
  touchGripKnobInput(
    inputId = "myKnob",
    width = "25%",
    label = "A label...",
    color = "#428BCA"
  ),
 verbatimTextOutput(outputId = "res")
)

server <- function(input, output, session) {

```

touchKnobInput

Highly configurable, touch-enabled knob input for Shiny

Description
Highly configurable, touch-enabled knob input for Shiny

Usage
touchKnobInput(
  inputId, 
  label = NULL, 
  labelPosition = "top", 
  width = "auto", 
  step = "any", 
  min = 0, 
  max = 1, 
  initial = 0.5, 
  color = "orange", 
  indicatorDot = TRUE, 
  indicatorRingType = "positive", 
  dragResistance = 100, 
  wheelResistance = 100, 
  globalRatePolicy = NULL, 
  globalRatePolicyDelay = 500
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inputId</td>
<td>The input slot that will be used to access the value.</td>
</tr>
<tr>
<td>label</td>
<td>Optional label for knob</td>
</tr>
<tr>
<td>labelPosition</td>
<td>Position of label (‘top’ or ‘bottom’)</td>
</tr>
<tr>
<td>width</td>
<td>Width of the knob as a percentage of the container element</td>
</tr>
<tr>
<td>step</td>
<td>The step amount for value changes, usually used with min and max parameters. Can be ‘any’ (no step)</td>
</tr>
<tr>
<td>min</td>
<td>The minimum input value.</td>
</tr>
<tr>
<td>max</td>
<td>The maximum input value.</td>
</tr>
<tr>
<td>initial</td>
<td>Initial value of the knob. Knob resets to this value when double-clicked.</td>
</tr>
</tbody>
</table>
touchKnobInput

color
The color to use for the indicator ring fill, focus indicator, and indicator dot (if present). You can set any *hex* color (ex: '#0F1BC3') or named color, choices are 'purple', 'blue', 'green', 'yellow', 'red', 'orange' or 'transparent' to disable

indicatorDot
Whether the knob should display an indicator dot for making it easier to read the current value. (TRUE or FALSE)

indicatorRingType
The fill style for the indicator ring. 'positive' - color fills in from the left as value increases. 'negative' - color fills in from the right as value decreases. 'split' - color fills left/right from middle as value increases/decreases relative to the middle value (half-way between min and max) Type of knob can be 'positive' (clockwise), 'negative' (counter-clockwise) or 'split' (plus/minus vs center position)

dragResistance
The amount of resistance to value change on mouse/touch drag events. Higher value means more precision, and the user will have to drag farther to change the input's value. (0 to 100) wheelResistance
The amount of resistance to value change on mouse wheel scroll. Higher value means more precision, and the mouse wheel will be less effective at changing the input's value. (0 to 100)

globalRatePolicy
Rate policy determining the behavior of output value events. NULL will output values in real-time, 'debounce' will output values once the knob stops moving, 'throttle' will output values while the knob is moving but only at a certain frequency (controlled with ratePolicyDelay). This setting will affect every touchKnobInput in the project.

globalRatePolicyDelay
Delay to use when globalRatePolicy is set to 'throttle' or 'debounce'. This setting will affect every touchKnobInput in the project.

Value
Numeric value server-side.

Examples

if (interactive()) {

library("shiny")
library("shinyKnobs")

ui <- fluidPage(
  touchKnobInput(
    inputId = "myKnob",
    label = "A label...",
    initial = 0,
    width = "25%",
    min = -50,
    max = 50,
    color = "#428BCA"
  )
)


```r
},
    verbatimTextOutput(outputId = "res")
)

server <- function(input, output, session) {

    output$res <- renderPrint(input$myKnob)

}

shinyApp(ui = ui, server = server)

}
Index

touchGripKnobInput, 2

touchKnobInput, 4