RatingScale

class psychopy.visual.**RatingScale**(*win*, *scale='<default>'*, *choices=None*, *low=1*, *high=7*, *precision=1*, *labels=()*, *tickMarks=None*, *tickHeight=1.0*, *marker='triangle'*, *markerStart=None*, *markerColor=None*, *markerExpansion=1*, *singleClick=False*, *disappear=False*, *textSize=1.0*, *textColor='LightGray'*, *textFont='Helvetica Bold'*, *showValue=True*, *showAccept=True*, *acceptKeys='return'*, *acceptPreText='key, click'*, *acceptText='accept?'*, *acceptSize=1.0*, *leftKeys='left'*, *rightKeys='right'*, *respKeys=()*, *lineColor='White'*, *skipKeys='tab'*, *mouseOnly=False*, *noMouse=False*, *size=1.0*, *stretch=1.0*, *pos=None*, *minTime=0.4*, *maxTime=0.0*, *flipVert=False*, *depth=0*, *name=None*, *autoLog=True*, ***kwargs*)

A class for obtaining ratings, e.g., on a 1-to-7 or categorical scale.

A RatingScale instance is a re-usable visual object having a `draw()` method, with customizable appearance and response options. `draw()` displays the rating scale, handles the subject's mouse or key responses, and updates the display. When the subject accepts a selection, `.noResponse` goes `False` (i.e., there is a response).

You can call the `getRating()` method anytime to get a rating, `getRT()` to get the decision time, or `getHistory()` to obtain the entire set of (rating, RT) pairs.

There are five main elements of a rating scale: the *`scale`* (text above the line intended to be a reminder of how to use the scale), the *`line`* (with tick marks), the *`marker`* (a moveable visual indicator on the line), the *`labels`* (text below the line that label specific points), and the *`accept`* button. The appearance and function of elements can be customized by the experimenter; it is not possible to orient a rating scale to be vertical. Multiple scales can be displayed at the same time, and continuous real-time ratings can be obtained from the history.

The Builder RatingScale component gives a restricted set of options, but also allows full control over a RatingScale via the 'customize_everything' field.

A RatingScale instance has no idea what else is on the screen. The experimenter has to draw the item to be rated, and handle *`escape`* to break or quit, if desired. The subject can use the mouse or keys to respond. Direction keys (left, right) will move the marker in the smallest available increment (e.g., 1/10th of a tick-mark if precision = 10).

Example 1:

A basic 7-point scale:

```
ratingScale = visual.RatingScale(win)
item = <statement, question, image, movie, ...>
while ratingScale.noResponse:
    item.draw()
    ratingScale.draw()
    win.flip()
rating = ratingScale.getRating()
decisionTime = ratingScale.getRT()
choiceHistory = ratingScale.getHistory()
```

Example 2:

For fMRI, sometimes only a keyboard can be used. If your response box sends keys 1-4, you could specify left, right, and accept keys, and not need a mouse:

```
ratingScale = visual.RatingScale(
    win, low=1, high=5, markerStart=4,
    leftKeys='1', rightKeys = '2', acceptKeys='4')
```

Example 3:

Categorical ratings can be obtained using choices:

```
ratingScale = visual.RatingScale(
    win, choices=['agree', 'disagree'],
    markerStart=0.5, singleClick=True)
```

For other examples see Coder Demos -> stimuli -> ratingScale.py.

Authors:

- 2010 Jeremy Gray: original code and on-going updates
- 2012 Henrik Singmann: tickMarks, labels, ticksAboveLine
- 2014 Jeremy Gray: multiple API changes (v1.80.00)

Parameters: win :

A `Window` object (required).

choices :

A list of items which the subject can choose among. `choices` takes precedence over `low`, `high`, `precision`, `scale`, `labels`, and `tickMarks`.

low :

Lowest numeric rating (integer), default = 1.

high :

Highest numeric rating (integer), default = 7.

precision :

Portions of a tick to accept as input [1, 10, 60, 100]; default = 1 (a whole tick). Pressing a key in *`leftKeys`* or *`rightKeys`* will move the marker by one portion of a tick. precision=60 is intended to support ratings of time-based quantities, with seconds being fractional minutes (or minutes being fractional hours). The display uses a colon (min:sec, or hours:min) to signal this to participants. The value returned by getRating() will be a proportion of a minute (e.g., 1:30 -> 1.5, or 59 seconds -> 59/60 = 0.98333). hours:min:sec is not supported.

scale :

Optional reminder message about how to respond or rate an item, displayed above the line; default = '<low>=not at all, <high>=extremely'. To suppress the scale, set `scale=None`.

labels :

Text to be placed at specific tick marks to indicate their value. Can be just the ends (if given 2 labels), ends + middle (if given 3 labels), or all points (if given the same number of labels as points).

tickMarks :

List of positions at which tick marks should be placed from low to high. The default is to space tick marks equally, one per integer value.

tickHeight :

The vertical height of tick marks: 1.0 is the default height (above line), -1.0 is below the line, and 0.0 suppresses the display of tickmarks. `tickHeight` is purely cosmetic, and can be fractional, e.g., 1.2.

marker :

The moveable visual indicator of the current selection. The predefined styles are 'triangle', 'circle', 'glow', 'slider', and 'hover'. A slider moves smoothly when there are enough screen positions to move through, e.g., low=0, high=100. Hovering requires a set of choices, and allows clicking directly on individual choices; dwell-time is not recorded. Can also be set to a custom marker stimulus: any object with a .draw() method and .pos will work, e.g., `visual.TextStim(win, text='[]', units='norm')`.

markerStart :

The location or value to be pre-selected upon initial display, either numeric or one of the choices. Can be fractional, e.g., midway between two options.

markerColor :

Color to use for a predefined marker style, e.g., 'DarkRed'.

markerExpansion :

Only affects the *glow* marker: How much to expand or contract when moving rightward; 0=none, negative shrinks.

singleClick :

Enable a mouse click to both select and accept the rating, default = `False`. A legal key press will also count as a singleClick. The 'accept' box is visible, but clicking it has no effect.

pos
: *tuple (x, y)*

Position of the rating scale on the screen. The midpoint of the line will be positioned at `(x, y)`; default = `(0.0, -0.4)` in norm units

size :

How much to expand or contract the overall rating scale display. Default size = 1.0. For larger than the default, set `size` > 1; for smaller, set < 1.

stretch:

Like `size`, but only affects the horizontal direction.

textSize :

The size of text elements, relative to the default size (i.e., a scaling factor, not points).

textColor :

Color to use for labels and scale text; default = 'LightGray'.

textFont :

Name of the font to use; default = 'Helvetica Bold'.

showValue :

Show the subject their current selection default = `True`. Ignored if singleClick is `True`.

showAccept :

Show the button to click to accept the current value by using the mouse; default = `True`.

acceptPreText :

The text to display before any value has been selected.

acceptText :

The text to display in the 'accept' button after a value has been selected.

acceptSize :

The width of the accept box relative to the default (e.g., 2 is twice as wide).

acceptKeys :

A list of keys that are used to accept the current response; default = 'return'.

leftKeys :

A list of keys that each mean "move leftwards"; default = 'left'.

rightKeys :

A list of keys that each mean "move rightwards"; default = 'right'.

respKeys :

A list of keys to use for selecting choices, in the desired order. The first item will be the left-most choice, the second item will be the next choice, and so on.

skipKeys :

List of keys the subject can use to skip a response, default = 'tab'. To require a response to every item, set `skipKeys=None`.

lineColor :

The RGB color to use for the scale line, default = 'White'.

mouseOnly :

Require the subject to use the mouse (any keyboard input is ignored), default = `False`. Can be used to avoid competing with other objects for keyboard input.

noMouse:

Require the subject to use keys to respond; disable and hide the mouse. *`markerStart`* will default to the left end.

minTime :

Seconds that must elapse before a response can be accepted, default = *`0.4`*.

maxTime :

Seconds after which a response cannot be accepted. If `maxTime` <= `minTime`, there's no time limit. Default = *`0.0`* (no time limit).

disappear :

Whether the rating scale should vanish after a value is accepted. Can be useful when showing multiple scales.

flipVert :

Whether to mirror-reverse the rating scale in the vertical direction.

`acceptResponse`(*triggeringAction*, *log=True*)

Commit and optionally log a response and the action.

draw(*log=True*)

Update the visual display, check for response (key, mouse, skip).

Sets response flags: *self.noResponse*, *self.timedOut*. *draw()* only draws the rating scale, not the item to be rated.

getHistory()

Return a list of the subject's history as (rating, time) tuples.

The history can be retrieved at any time, allowing for continuous ratings to be obtained in real-time. Both numerical and categorical choices are stored automatically in the history.

getRT()

Returns the seconds taken to make the rating (or to indicate skip).

Returns None if no rating available, or maxTime if the response timed out. Returns the time elapsed so far if no rating has been accepted yet (e.g., for continuous usage).

getRating()

Returns the final, accepted rating, or the current value.

The rating is None if the subject skipped this item, took longer than `maxTime`, or no rating is available yet. Returns the currently indicated rating even if it has not been accepted yet (and so might change until accept is pressed). The first rating in the list will have the value of markerStart (whether None, a numeric value, or a choice value).

reset(*log=True*)

Restores the rating-scale to its post-creation state.

The history is cleared, and the status is set to NOT_STARTED. Does not restore the scale text description (such reset is needed between items when rating multiple items)

setDescription(*scale=None*, *log=True*)

Method to set the brief description (scale).

Useful when using the same RatingScale object to rate several dimensions. *setDescription(None)* will reset the description to its initial state. Set to a space character (' ') to make the description invisible.

setFlipVert(*newVal=True*, *log=True*)

Sets current vertical mirroring to `newVal`.

setMarkerPos(*tick*)

Method to allow the experimenter to set the marker's position on the scale (in units of tick marks). This method can also set the index within a list of choices (which start at 0). No range checking is done.

Assuming you have defined rs = RatingScale(...), you can specify a tick position directly:

```
rs.setMarkerPos(2)
```

or do range checking, precision management, and auto-rescaling:

```
rs.setMarkerPos(rs._getMarkerFromTick(2))
```

To work from a screen coordinate, such as the X position of a mouse click:

```
rs.setMarkerPos(rs._getMarkerFromPos(mouseX))
```