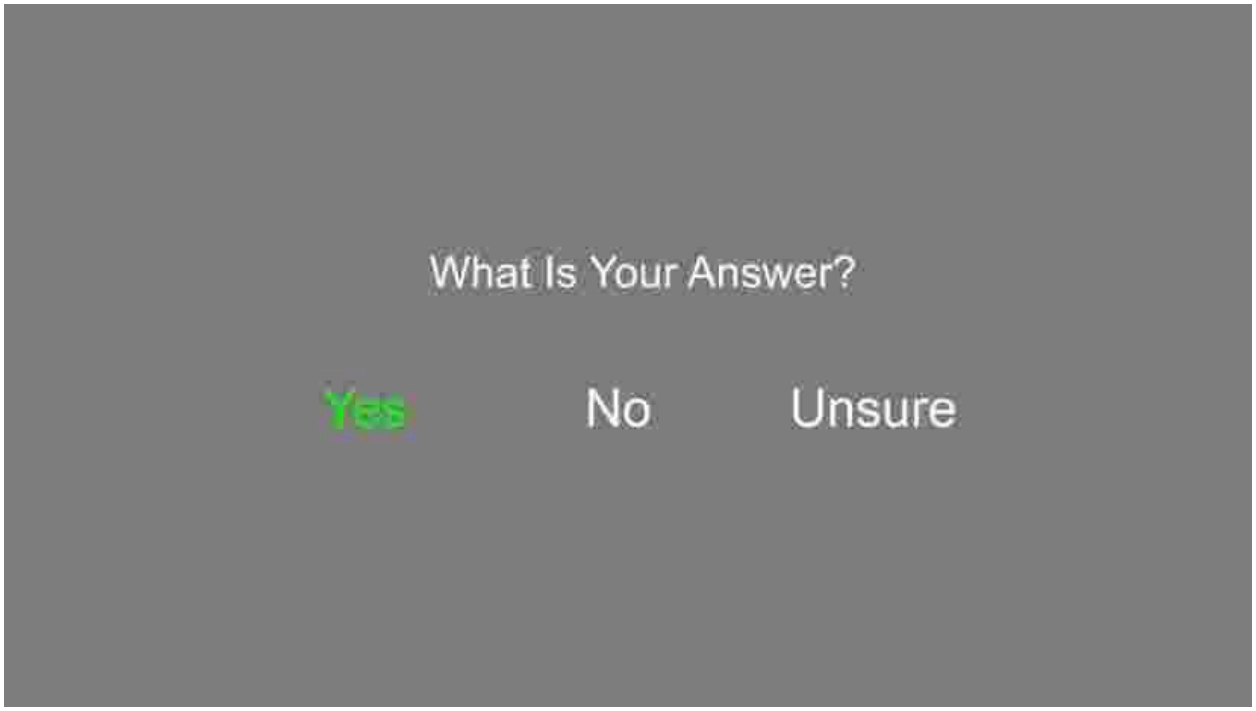
Adding a Rating Scale GUI

There are examples in the ExampleScripts folder under "RatingScale_GUI" to see how to have the rating available. You can use different yield statements to trigger the rating GUI to show up.



Add this line to show a rating:

```
yield sightlab.showRatings("What Is Your Answer?")
```

Can set Choice Text and adjust options, such as pausing the timer

```
yield sightlab.showRatings('What Is Your Answer?', ratingScale = ["Yes", "No",
"Unsure"], pauseTimer = True)
import sightlab_utils.sightlab as sl
from sightlab_utils.settings import *
sightlab = sl.SightLab()
scaleList = ["Yes", "No", "Unsure"]

def sightLabExperiment():
 while True:
```

```
  yield viztask.waitKeyDown(' ')
  yield sightlab.startTrial()
  yield viztask.waitKeyDown(' ')
  yield sightlab.showRatings('What Is Your Answer?', ratingScale = scaleList,
pauseTimer = True)
  yield sightlab.endTrial()
viztask.schedule(sightlab.runExperiment)
viztask.schedule(sightLabExperiment)
```

For getting access to the rating value you can use this code:

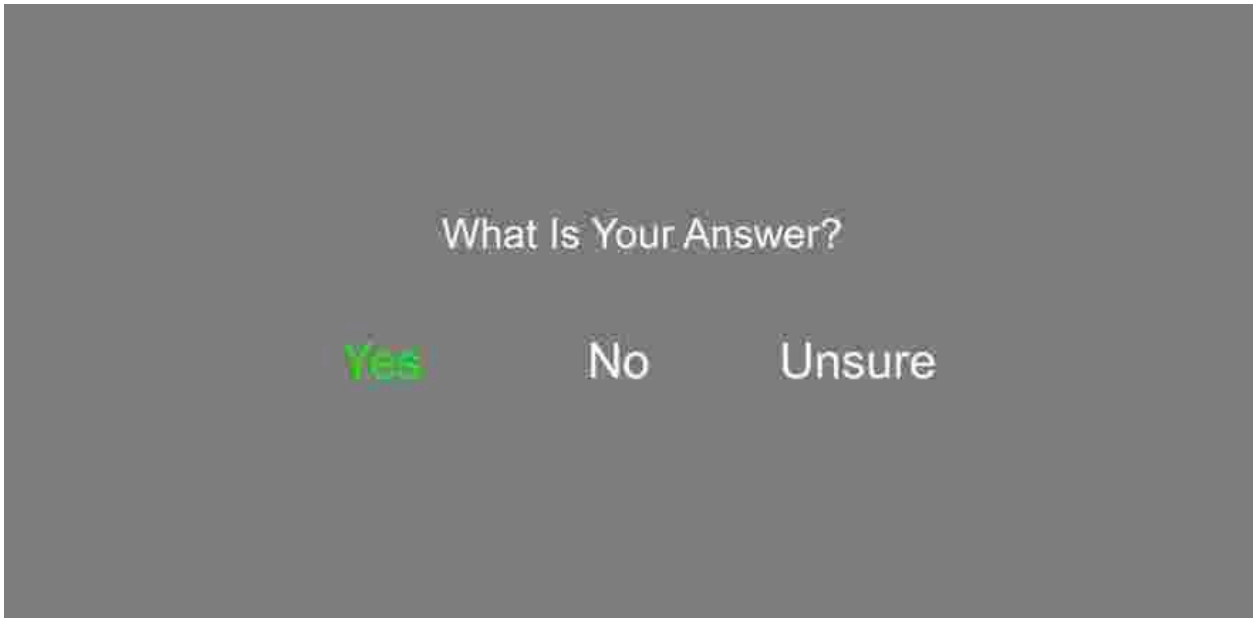
```
chosen_rating = sightlab.ratingChoice
```

And for something to change based on the rating value:

```
if chosen_rating >= 8:
    print("Rating too high.")
```

There are many options available to trigger an event. Here are just a few examples:

- Standard viztask commands, such as when a media file stops playing, an animation finishes or a button is pressed.
- Using proximity sensors with vizproximity.waitEnter (if you have a proximity sensor already setup, for more on proximity sensors see here)
- Using a grab event to be triggered when an object is grabbed.
- A gaze event (i.e. when you view an object), see the example RatingScale_after_view for this scenario in the ExampleScripts folder
- And much more When you run the experiment now the rating will show up after the specified trigger. Use the right hand trigger to proceed and the left and right trackpad or thumbstick to choose the answers. For the desktop use the left mouse button and arrow keys.



The chosen values will also be saved in the trial_data.csv file

5.171	1000	1000	1000	0	1.82	0	-7.92	10.35	0 FIXATION	0	-	-
5.187	1000	1000	1000	0	1.82	0	-7.92	10.35	0 FIXATION	0	-	-
5.204	1000	1000	1000	0	1.82	0	-7.92	10.35	0 FIXATION	0	-	-
5.221	1000	1000	1000	0	1.82	0	-7.92	10.35	0 SACCADE	0	-	-
5.237	1000	1000	1000	0	1.82	0	-7.92	10.35	0 SACCADE	0	0	-
5.255	1000	1000	1000	0	1.82	0	-7.92	10.35	0 SACCADE	0	0	-
5.271	1000	1000	1000	0	1.82	0	-7.92	10.35	0 SACCADE	0	0	-
5.288	-0.139	1.636	1	-0.003	1.815	0.025	-7.92	10.35	0 SACCADE	2.128	125.173	subject chose: Unsure
5.481	-0.139	1.636	1	0	1.82	0	-7.92	10.35	0 FIXATION	0	('avg: ', 8.18	-
5.486	-0.139	1.636	1	0	1.82	0	-7.92	10.35	0 FIXATION	0	-	-
5.491	-0.139	1.636	1	0	1.82	0	-7.92	10.35	0 FIXATION	0	-	-
5.504	-0.139	1.636	1	0	1.82	0	-7.92	10.35	0 FIXATION	0	-	-
5.52	-0.139	1.636	1	0	1.82	0	-7.92	10.35	0 FIXATION	0	-	-
5.537	-0.139	1.636	1	0	1.82	0	-7.92	10.35	0 FIXATION	0	-	-
5.554	-0.139	1.636	1	0	1.82	0	-7.92	10.35	0 FIXATION	0	-	-
5.571	-0.139	1.636	1	0	1.82	0	-7.92	10.35	0 FIXATION	0	-	-

To add to the experiment summary, use this code:

Attributes:

```
chosen_rating = sightlab.ratingChoice

experimentData = {
    'Rating': chosen_rating
}
for columnName, data in experimentData.items():
    sightlab.setExperimentSummaryData(columnName, data)
```

Can set ratingScale to be any string

```
scaleList = ["Yes", "No", "Unsure"]
showRatings(
    message,
    ratingScale=["0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10"],
    color=viz.WHITE,
    characterLimit=8,
    pauseTimer=False,
)
yield sightlab.showRatings(
    "What Is Your Answer?", ratingScale=scaleList, pauseTimer=True
)
```

There is also the option for a startRating or endRating (note: If using this method with the GUI it is best to set the Start and End conditions to 'None' and use code to set the start and end conditions).

```
yield sightlab.endTrial(
    endRatings={"message": "Which One Do You Prefer?", "ratingScale": ["Left",
"Right"]}
)
```

To add longer questions, you can use a combination of instructions and ratings, as in this example. This also shows using vizinput to collect demographics and save them to the data file.

```
import sightlab_utils.sightlab as sl
from sightlab_utils.settings import *

sightlab = sl.SightLab(gui = False, pid = False)

env = vizfx.addChild("sightlab_resources/environments/DeckersOffice.OSGB")
sightlab.setEnvironment(env)
sightlab.setStartText(' ')

#run sightlab experiment
def sightLabExperiment():
    yield viztask.waitEvent(EXPERIMENT_START)
    for i in range(0, sightlab.getTrialCount()):
```

```
        yield sightlab.startTrial()

        import vizinput
        # Create an empty dictionary to store responses
        responses = {}

        # Collect demographic data
        responses['Age'] = vizinput.input("What is your age?")

        responses['Gender'] = vizinput.choose("What is your gender?", ["Male",
"Female", "Other", "Prefer not to answer"])

        responses['Ethnicity'] = vizinput.choose("What is your ethnicity?", [
            "White/Caucasian (non-Hispanic)",
            "African American",
            "Hispanic/Latino",
            "Asian/Pacific Islander",
            "Middle Eastern/West Asian",
            "Other"
        ])

        # Save responses to experiment summary
        for key, value in responses.items():
            sightlab.setExperimentSummaryData(key, value)

        vizinput.message("Please enter your name in the space provided.")

        responses['Name'] = vizinput.input("Enter your name:")
        sightlab.setExperimentSummaryData("Name", responses['Name'])

        yield vizinput.message("By clicking continue, you indicate you consent
to participate in the study. Once you click forward, the simulation will
begin.")

        yield viztask.waitTime(2)

        # Survey Questions
        yield sightlab.showInstructions("Reflecting on your experience so far,
\n\n please indicate your agreement with the following statements:",
pauseTimer=True)
        yield viztask.waitEvent('triggerPress')
        yield sightlab.hideInstructions(endFade=False)
```

```
        yield sightlab.setDataLoggingState(1)

        yield sightlab.showRatings("I found the experience engaging.",
ratingScale=["1", "2", "3", "4", "5"], endFade=False)
        responses['Experience_Engagement'] = sightlab.ratingChoice

        yield sightlab.showRatings("The experience felt realistic.",
ratingScale=["1", "2", "3", "4", "5"], endFade=False)
        responses['Experience_Realism'] = sightlab.ratingChoice

        yield sightlab.showRatings("The purpose of the experience was clear.",
ratingScale=["1", "2", "3", "4", "5"], endFade=False)
        responses['Experience_Clarity'] = sightlab.ratingChoice

        yield sightlab.showRatings("I enjoyed participating in the
experience.", ratingScale=["1", "2", "3", "4", "5"], endFade=False)
        responses['Experience_Enjoyment'] = sightlab.ratingChoice


        yield sightlab.showInstructions("Please rate how well each of the
following \n\n statements describes your thoughts, feelings, and behaviors:",
pauseTimer=True)
        yield viztask.waitEvent('triggerPress')
        yield sightlab.hideInstructions(endFade=False)
        yield sightlab.setDataLoggingState(1)

        yield sightlab.showRatings("I enjoy social interactions.",
ratingScale=["1", "2", "3", "4", "5"], endFade=False)
        responses['Personality_Social'] = sightlab.ratingChoice

        yield sightlab.showRatings("I am naturally curious about new
experiences.", ratingScale=["1", "2", "3", "4", "5"], endFade=False)
        responses['Personality_Curious'] = sightlab.ratingChoice

        yield sightlab.showRatings("I adapt easily to new situations.",
ratingScale=["1", "2", "3", "4", "5"], endFade=False)
        responses['Personality_Adaptability'] = sightlab.ratingChoice

        yield sightlab.showRatings("I can focus on a task for extended
periods.", ratingScale=["1", "2", "3", "4", "5"])
        responses['Personality_Focus'] = sightlab.ratingChoice
```

```
        # Save responses to experiment summary
        for key, value in responses.items():
            sightlab.setExperimentSummaryData(key, value)

        yield sightlab.endTrial(endExperimentText = 'Thank You for
Participating')

viztask.schedule(sightlab.runExperiment)

viztask.schedule(sightLabExperiment)
```