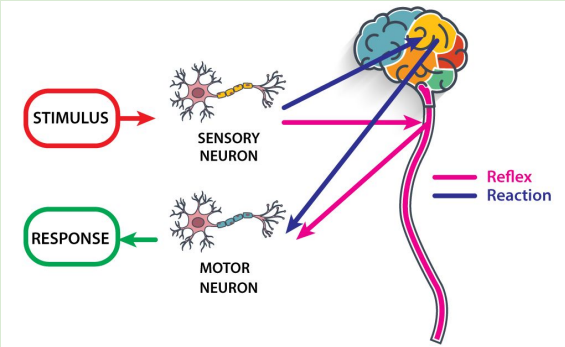
Mental Chronometry

Cross-curricular mission for CodeX

Mental Chronometry

The scientific study of processing speed, or reaction time, on a cognitive task

- Measures elapsed time between stimulus and response
- Used to study the time course of information processing in the nervous system
- A typical behavioral response is a button press

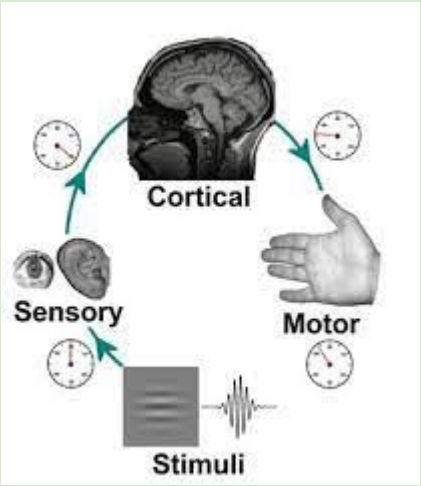


FIRIA LABS

Mental Chronometry

For this mission you will:

- experiment with different types of digital stimuli
- measure the reaction time for each one
- analyze the data
- draw conclusions



FIRIA LABS

Mission 10: Reaction Time

How fast is your reaction time?

- In this project you will make a device to measure your reaction time!
- Create a device that measures the time between:
  - Bright Pixel LEDs lighting up and
  - A CodeX button being pressed

FIRIA LABS

Experiment #1: Bright Light

After completing Mission 10: Reaction Time, you are ready for test data.

- Save the file as **MentalChronometry1**
- Run at least 10 tests
- Record your data in Chart #1
- Then calculate the average response time for bright light

Using sound with CodeX

Python has a soundlib module that can be imported, with built-in sound functions

Some of the functions are:

- Choose an instrument for a digital sound
- Set the pitch of the sound
- Start the sound
- Stop the sound after a delay

FIRIA LABS

Experiment #2: Audio Stimulus

Save your code with a new filename: **MentalChronometry2**

- Import the soundlib module
- Add audio variables above the while loop

Mental_chronometry

```
1   # Mental Chronometry
2   from codex import *
3   import time
4   import random
5   from soundlib import *
6
```

```
13  # Audio variables for digital sound
14  tone = soundmaker.get_tone('trumpet')
15  tone.set_pitch(800)
16
17  while True:
18      display.print("Press Button A")
19      wait_button()
```

FIRIA LABS

Experiment #2: Audio Stimulus

Replace the code for green pixels.

```
# light test case
pixels.set([GREEN, GREEN, GREEN, GREEN])
```

with the code to start the sound.

```
# Experiement -- sound
tone.play()
```

After the button is pressed, stop the sound.

```
end_time = time.ticks_ms()

# stop tone
time.sleep(0.5)
tone.stop()

reaction_time = time.ticks_diff(end_time, start_time)
```

FIRIA LABS

Experiment #2: Audio Stimulus

After modifying the code, you are ready for more test data.

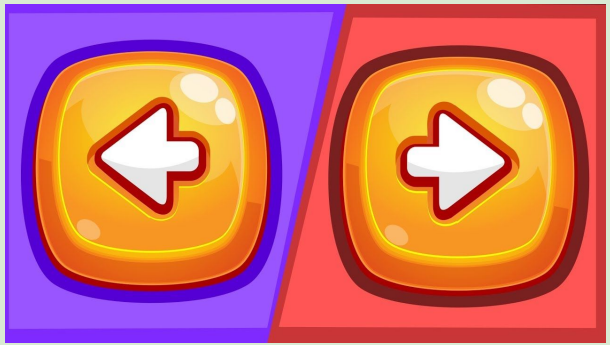
- Run at least 10 tests
- Record your data in Chart #2
- Then calculate the average response time for an audio stimulus

FIRIA LABS

Response with Selection

The first two experiments had a single response – one button press

- Modify the code so there are two possible responses
- Randomly light up either the two left pixels or the two right pixels
- Use two different buttons for the responses
  - BTN_L for the left pixels
  - BTN_B for the right pixels



FIRIA LABS

Experiment #3: 2-Button Responses

Save your code with a new filename: **MentalChronometry3**

- Delete the code for audio

```
# Audio variables for sound stimulus
tone = soundmaker.get_tone('trumpet')
tone.set_pitch(800)
```

```
# Experiement -- audio stimulus
tone.play()
```

```
# stop tone
time.sleep(0.5)
tone.stop()
```

FIRIA LABS

Experiment #3: 2-Button Responses

The computer will need to select a random side to light up.

- Add this code just before getting start_time
- Randomly select a button from the list
- Use an if statement to light up the left or right pixels

```
# Experiement -- 2 button responses
test_button = random.choice([BTN_L, BTN_B])
if test_button == BTN_L:
    pixels.set([GREEN, GREEN, BLACK, BLACK])
else:
    pixels.set([BLACK, BLACK, GREEN, GREEN])

start_time = time.ticks_ms()
```

FIRIA LABS

Experiment #3: 2-Button Responses

The next line of code calls the **wait()** function, which checks for BTN_A. Wait a minute! We now want to check for BTN_L or BTN_B.

- Add an parameter to the **wait()** function!
- The correct button is an argument, passed to the parameter and used in the function.

```
def wait_button(test_button):
    # Wait for test button.
    while True:
        if buttons.was_pressed(test_button):
            break
```

FIRIA LABS

Experiment #3: 2-Button Responses

Now that the **wait()** function has an argument, every wait() function call will need an argument.

- The first function call uses BTN_A as an argument to start the test
- The second function call uses either BTN_L or BTN_B, which is the value of test_button

```
while True:
    display.print("Press Button A")
    wait_button(BTN_A)
```

```
start_time = time.ticks_ms()

wait_button(test_button)

end_time = time.ticks_ms()
```

FIRIA LABS

Experiment #3: 2-Button Responses

Just one more thing ... cheating!

- Since two buttons are possible, either one could be pressed in advance
- Modify the “no cheating” code for BTN_L and BTN_B

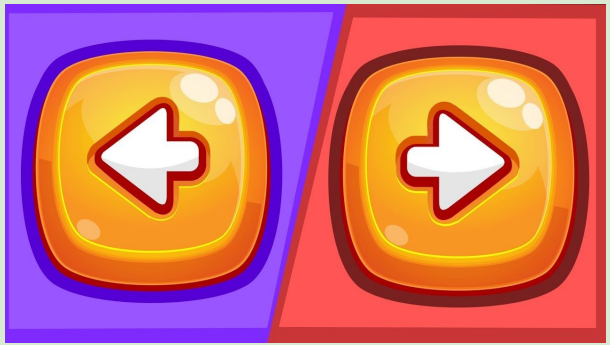
```
# Reset button state to prevent cheating
buttons.was_pressed(BTN_L)
buttons.was_pressed(BTN_B)
```

FIRIA LABS

Experiment #3: 2-Button Responses

After modifying the code, you are ready for more test data.

- Run at least 10 tests
- Record your data in Chart #3
- Then calculate the average response time for a 2-button stimulus

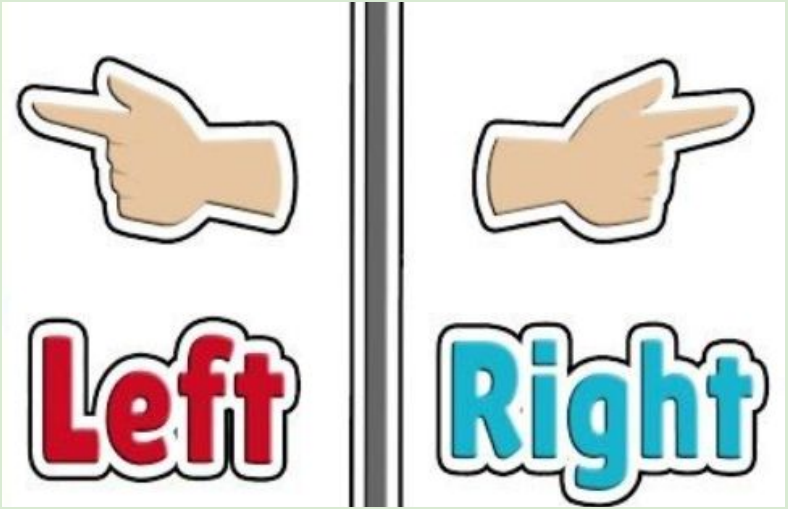


FIRIA LABS

Respond to Text

Change the stimulus by using text

- The experiment will still have two possible responses
- Randomly show “LEFT” or “RIGHT” on the display
- Use two different buttons for the responses
    - BTN_L for the left
    - BTN_B for the right



FIRIA LABS

Experiment #4: Response with Text

Save your code with a new filename: **MentalChronometry4**

- The only modification needed is inside the if statement
- Change the pixels.set() command to a print() statement

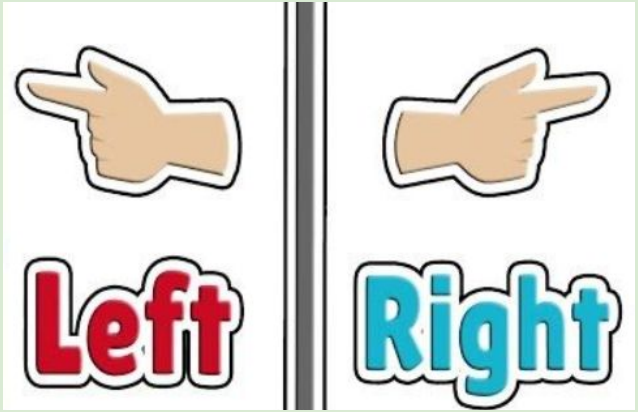
```
# Experiement -- 2 button responses with text
test_button = random.choice([BTN_L, BTN_B])
if test_button == BTN_L:
    display.print("LEFT", color=YELLOW, scale=7)
else:
    display.print("RIGHT", color=YELLOW, scale=7)
```

FIRIA LABS

Experiment #4: Response with Text

After modifying the code, you are ready for more test data.

- Run at least 10 tests
- Record your data in Chart #4
- Then calculate the average response time for a response with text

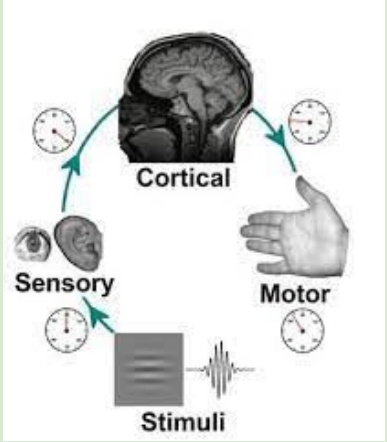


FIRIA LABS

Experiment Variations

Other experiments you can try:

- Experiment #1: Bright light
  - Try different colors and compare the reaction times
- Experiment #2: Audio stimulus
  - Try different tones (high, low) and compare times
  - Try two tones -- one for left and one for right
- Experiment #3: 2-Button Responses
  - Keep track of left and right reaction times separately to see if one is better than the other
  - Try combining a tone with the lights and compare to just sound or just lights
- Experiment #4: Response with Text
  - Try different colors and / or text scale
  - Use two images instead of text



FIRIA LABS

Data Analysis

- Compare your data with other participants
- Select one of the experiments
- Create a graph that shows the data for that experiment with at least four other participants

FIRIA LABS

Data Analysis

- Look at your data for all four experiments
- Create a chart with the data
- What other visual representations can you create? Come up with another representation of your own

FIRIA LABS

Conclusions

- Using your graph, chart and visual representation, what are your conclusions?