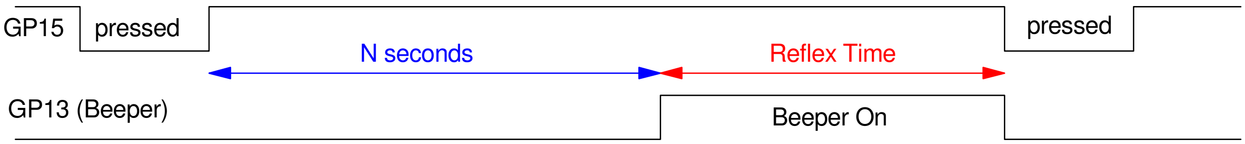
ECE 476/676 - Homework #6

Math, Random, and Matrix Routines - Due Monday, March 3rd

Reflex Game

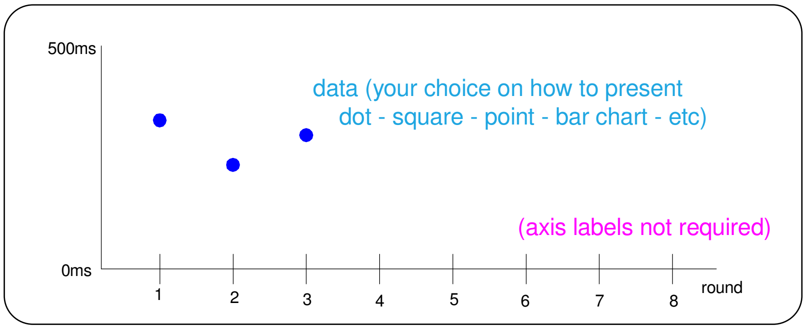
Write a Python program to measure your reflex time.

- Each round starts with pressing and releasing button GP15
- When pressed, a random time, N, is generated
  - N has a Gamma distrubution with p = 1/2 and r = 3
  - (sum of three exponential distibutions, each with a mean of 2 seconds)
- The Pico board then waits N seconds and then turns on the beeper
- Once the beeper turns on, you are to press GP15 again
- The time between the beeper turning on and your pressing GP15 is your reflex time.



1) Write a Python program to play this game one time

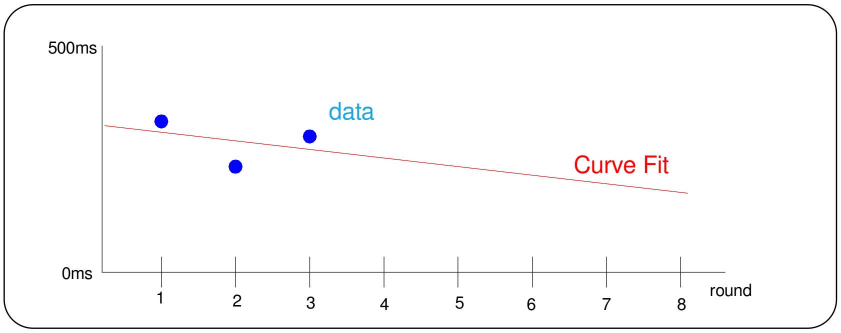
2) Add graphics at the end to display your reflex time after each round (from 1 to 8)



3) After round 2 and thereafter, add a routine to compute a least-squares curve fit for your reflex time after each round

reflex time = a * round + b

4) Display this curve fit after each round on the LCD display



5)Demo

Start with the gamma funciton

- Pass p and r
- Return x

```
def Gamma(p, r):
    N = 0
    for i in range(0,r):
        N += -log(random())/p
    return(N)

for i in range(0,5):
    t = Gamma(0.5, 3)
    print(t)

-----------

    4.7350
    5.7528
    3.8506
    3.8264
    1.9788
```

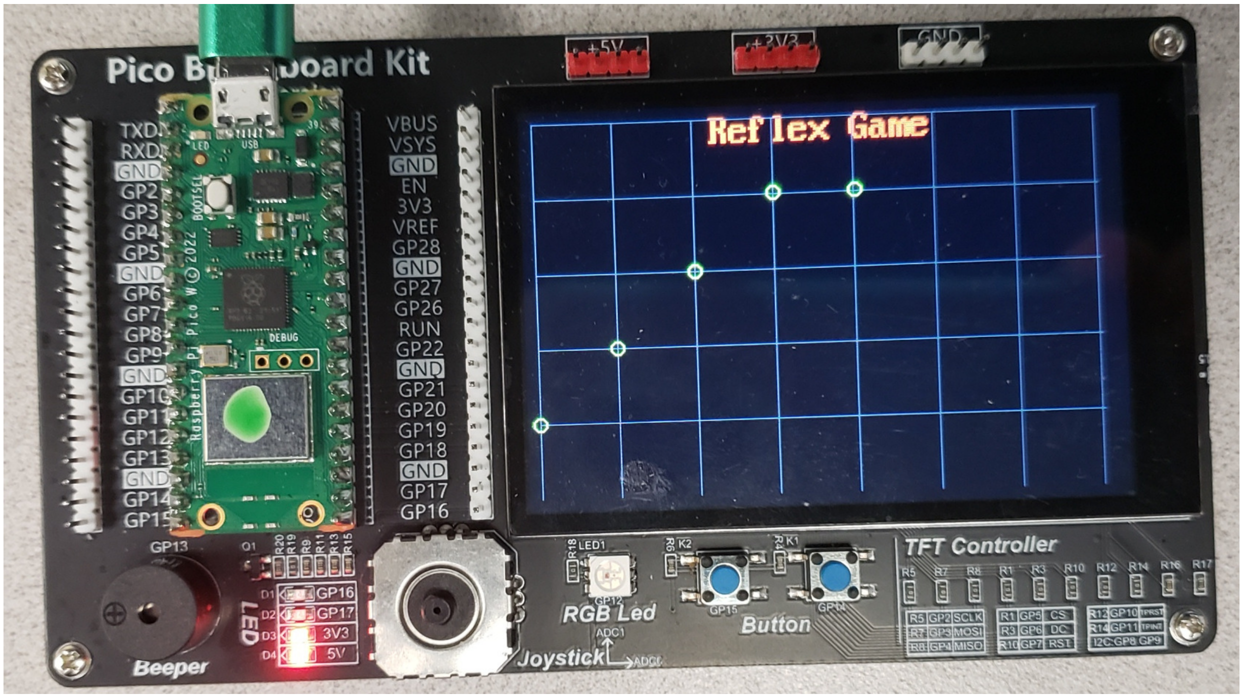
Looks reasonable....

Next, work on the display routine

- Pass two vectors
  - Rounds played
  - Reflex time
- Display the raw data

```
def Display(X, T):
    LCD.Clear(Black)
    LCD.Title('Reflex Game', Orange, White)

    kX = 450/7
    kY = 300/500

    for y in range(10,310,60):
        LCD.Line(10,y,460,y, Grey)
    for x in range(10, 460, 64):
        LCD.Line(x,10,x,310,Grey)

    n = len(X)

    for i in range(0,n):
        LCD.Circle(10 + kX*X[i], 310 - kY*T[i], 5, Yellow)


Display([0,1,2,3,4],[100,200,300,400,500])
```

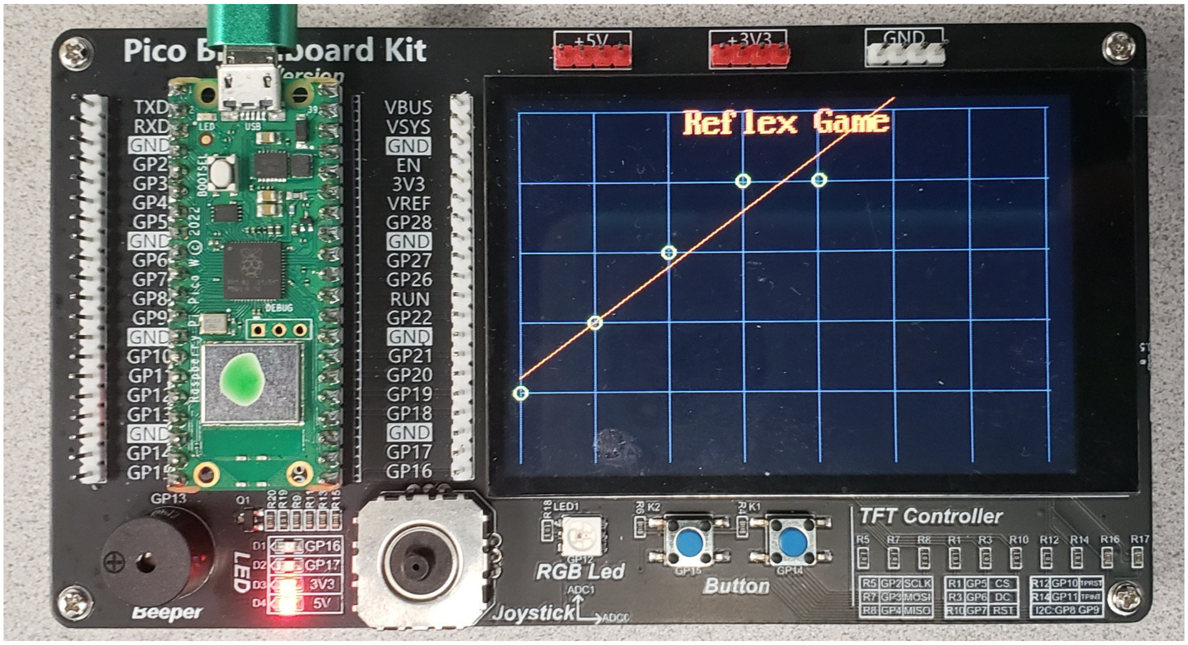
This displays five circles as desired



Display(X,Y) routine displays the data along with tic marks
Y axis = 0ms to 500ms (100ms per tic mark)
X axis = trial # (0..7)

Now add the least-squares function

```
def Display(X, T):
    LCD.Clear(Black)
    LCD.Title('Reflex Game', Orange, White)

    kX = 450/7
    kY = 300/500

    for y in range(10,310,60):
        LCD.Line(10,y,460,y, Grey)
    for x in range(10, 460, 64):
        LCD.Line(x,10,x,310,Grey)

    n = len(X)

    for i in range(0,n):
        LCD.Circle(10 + kX*X[i], 310 - kY*T[i], 5, Yellow)

    # Curve Fit
    if(n > 1):
        B = matrix.zeros(n, 2)
        Y = matrix.zeros(n, 1)
        for i in range(0,n):
            B[i][0] = X[i]
            B[i][1] = 1
            Y[i][0] = T[i]

        BT = matrix.transpose(B)
        BTB = matrix.mult(BT,B)
        BTBi = matrix.inv(BTB)
        BTY = matrix.mult(BT,Y)
        A = matrix.mult(BTBi,BTY)

        y0 = A[0][0]*0 + A[1][0]
        y7 = A[0][0]*7 + A[1][0]
        LCD.Line(10, 310-y0*kY, 460, 310-y7*kY, Orange)

Display([0,1,2,3,4],[100,200,300,300])
```



Display routine displays the data along with a least-squares curve fit

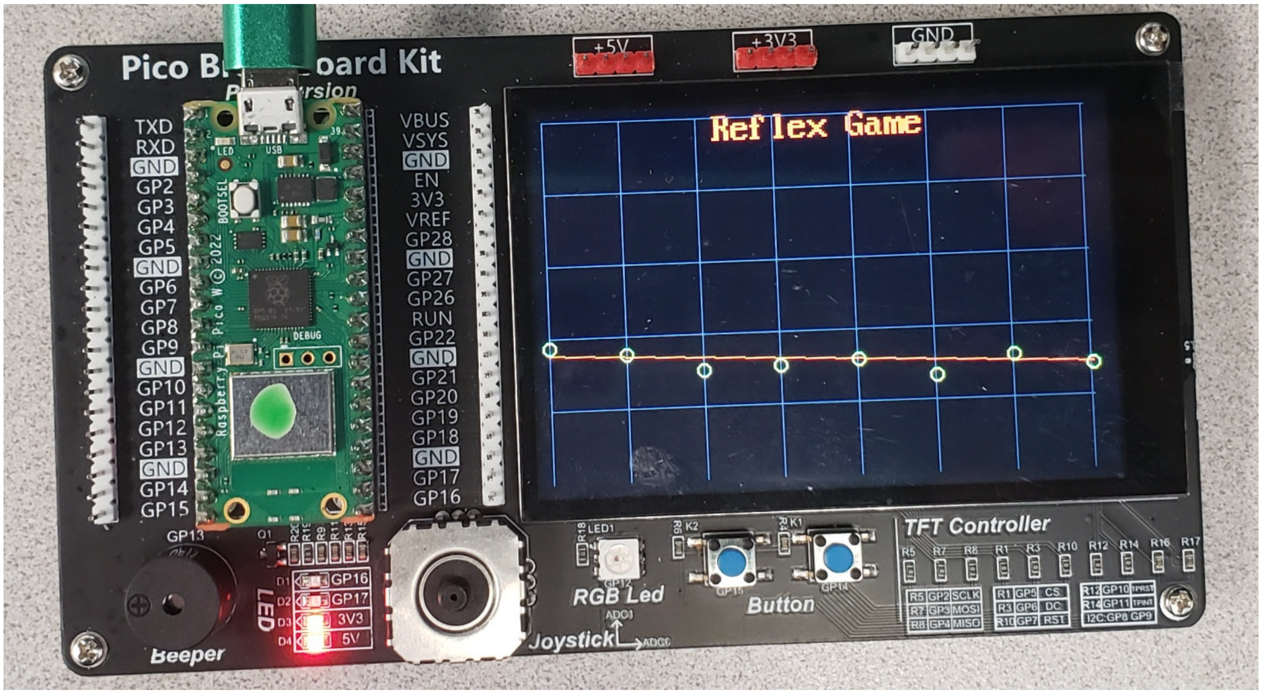
Now add the reflex game

```
from time import sleep, ticks_ms
from random import random, randrange
from math import log
import LCD
from machine import Pin
import matrix

def Gamma(p, r):
:
Black = LCD.RGB(0,0,0)
White = LCD.RGB(0,0,0)
Yellow = LCD.RGB(250,250,0)
Orange = LCD.RGB(250,150,0)
Grey = LCD.RGB(100,100,100)

def Display(X, T):
:


Button = Pin(15, Pin.IN, Pin.PULL_UP)
Beeper = Pin(13, Pin.OUT)
Beeper.value(0)


LCD.Init()

X = []
Y = []

Display(X, Y)

while(Button.value() == 1):
    pass
while(Button.value() == 0):
    pass

for Round in range(0,8):

    Time = Gamma(0.5, 3)
    sleep(Time)

    T0 = ticks_ms()
    Beeper.value(1)

    while(Button.value() == 1):
        pass

    T1 = ticks_ms()
    Beeper.value(0)

    X.append(Round)
    Y.append(T1-T0)
    print(Y)

    Display(X, Y)
```

The results after each round are displayed on the shell window:

```
[186]
[186, 175]
[186, 175, 150]
[186, 175, 150, 152]
[186, 175, 150, 152, 158]
[186, 175, 150, 152, 158, 133]
[186, 175, 150, 152, 158, 133, 158]
[186, 175, 150, 152, 158, 133, 158, 142]
>>>
```

The LCD display shows the data after each round along with the linear curve fit



Results after eight rounds
Y axis = 0ms to 500ms (100ms per tic mark)
X axis = trial # (0..7)