Reaction timer with PicoBoard and Scratch

Description:

The objective of this program is to measure your reaction time with a visual and aural cue. The Scratch Program will wait a random amount of seconds and then start a clock and play a sound. You then press the space bar or the button on the PicoBoard to signal that you heard the sound/saw the clock move. Scratch then measures your elapsed time and provides feedback. Scratch also keeps a list of all your times and provides your average time. In addition, Scratch will graph your times and indicate your average on this graph.

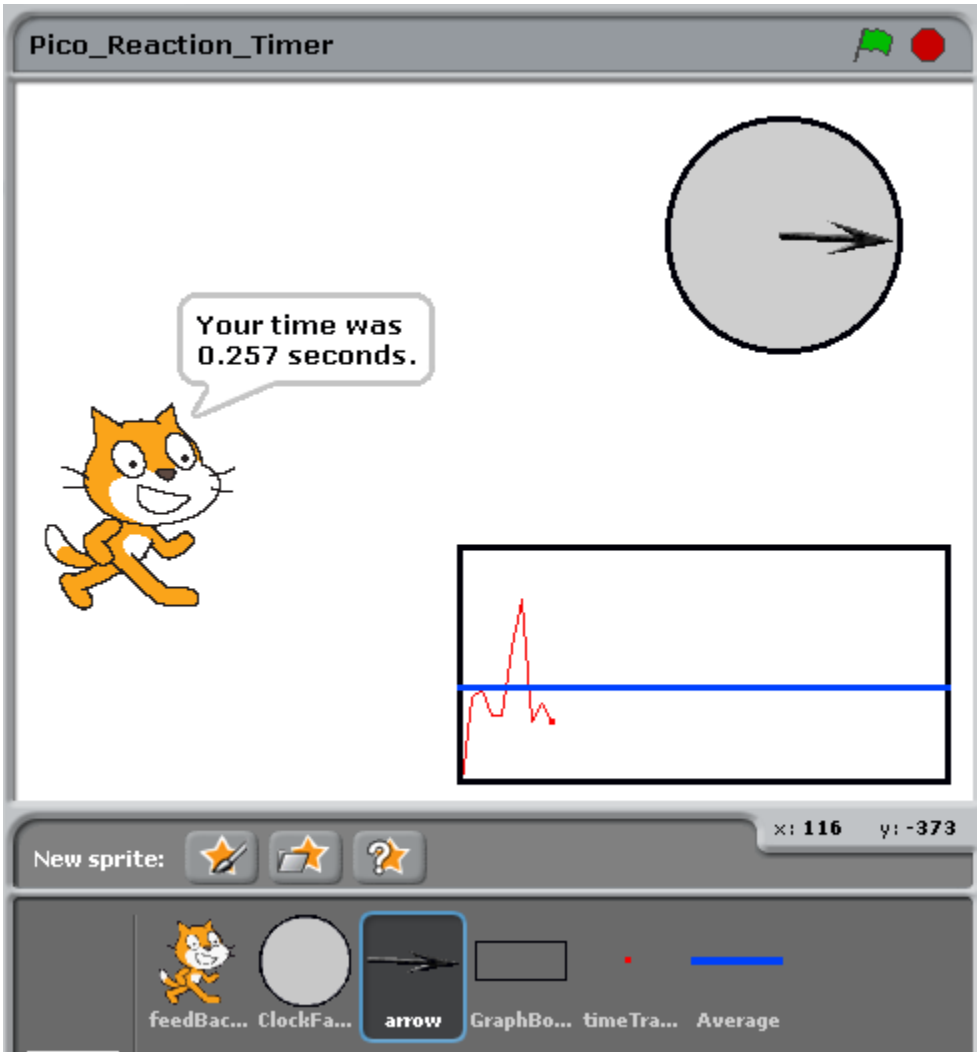
This project will require the programmer to master the following skills:

1. Reading Sensor Input from the PicoBoard (Button for user select and Slider for Sound Value)
2. Use the Scratch Timer and Variables to measure elapsed Time.
3. Use the Scratch Random Tile
4. Use a List in Scratch to store the user's results.
5. Use a 'For Loop' type structure in Scratch to calculate the average values in the List.
6. Provide graphical feedback with Pen in creating Line Graphs.

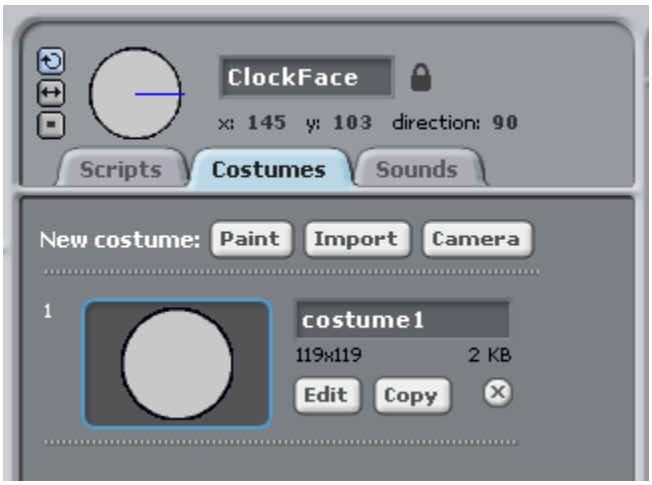
Process:

Part 1: Setup the Sprites and Variables

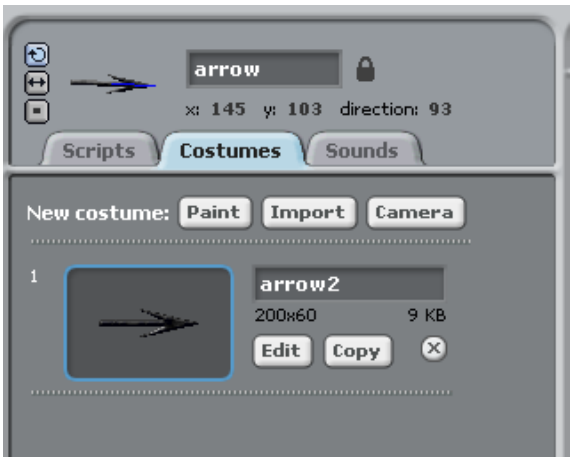
1. Start Scratch and save the project as 'lastname_reaction.'
2. Create the Following Sprites:
    a. feedBackCat: Tells the User what their time was.
    b. ClockFace: Grey Circle for the background of the clock
    c. arrow: Spins on the ClockFace and indicates the passing of time.
    d. GraphBox: Surrounds the timeTracer and Average Line Graphs
    e. timeTracer: Will trace out the user reaction times to create a line graph
    f. Average: A Blue Line that will indicate the current user time average.

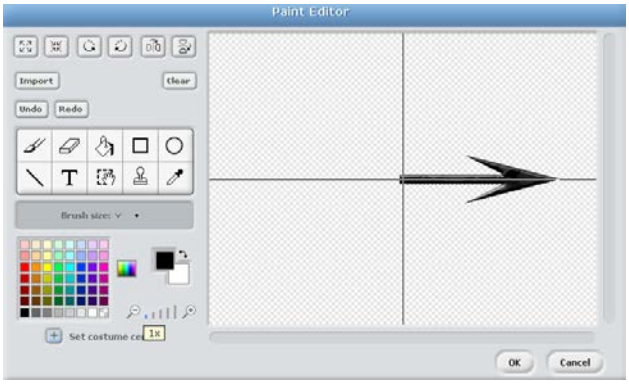


3. The Details for the Images of the Sprites are as follows:
    a. feedBackCat: Standard Scratch Kitty
    b. ClockFace:

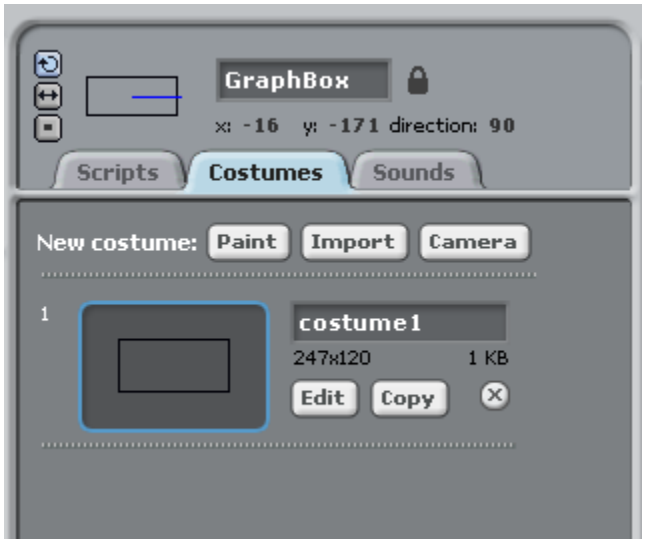


    c. Arrow (Make sure the costume center is at the end of the arrow)

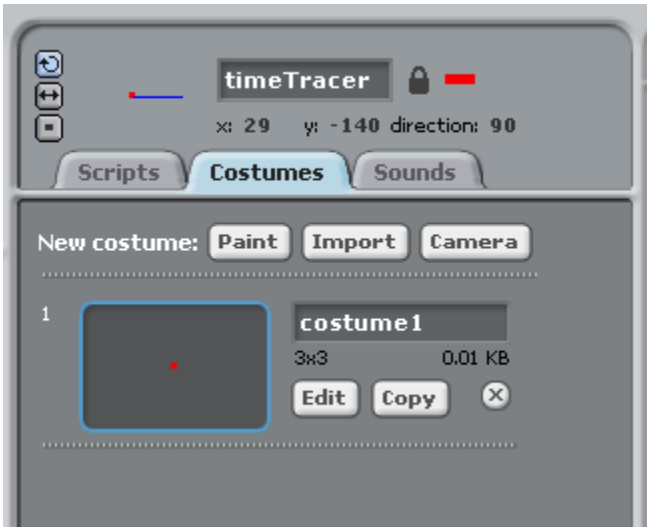




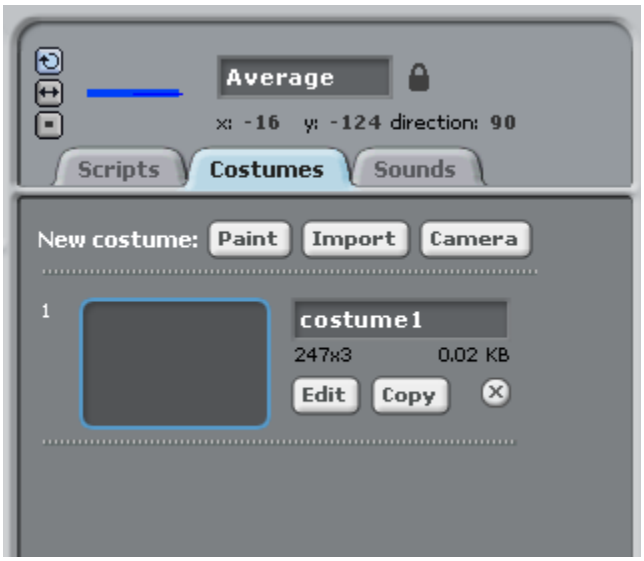
d. GraphBox (Note the Dimensions of 247x120) This does not have to be exact, but we need to know the dimensions later to control the timeTracer and Average Sprites.



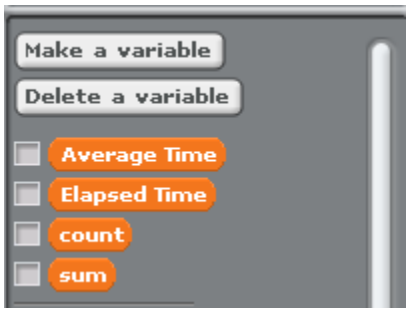
e. timeTracer (Just a Red Dot)



f. Average Line (Should be the same width as the GraphBox)

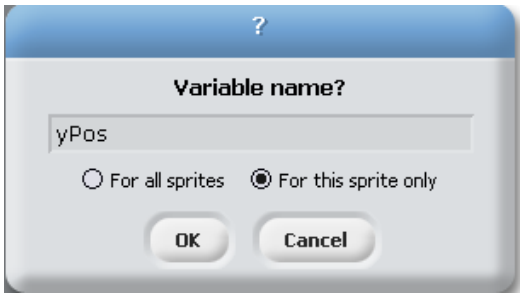


4. Create the following Variables (For All Sprites)



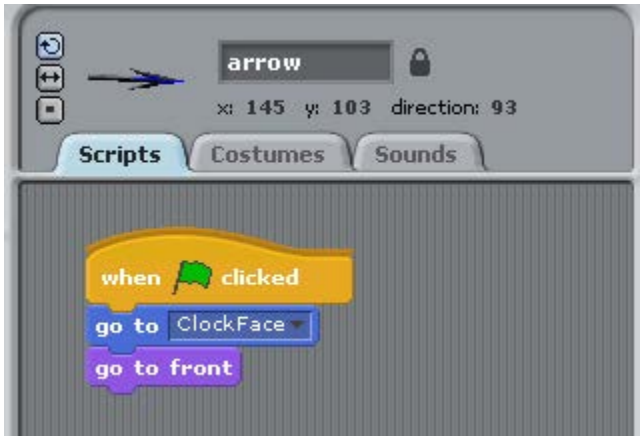
5. Create the following List for all Sprites: (List is named 'Times')

6. On the timeTracer Sprite, create the following ‘private’ variables (“For This Sprite Only”)

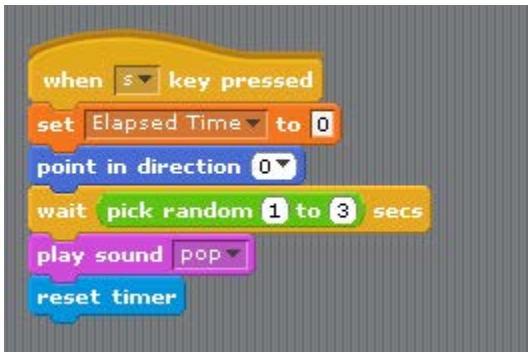


Part 2: Write the Scripts for the arrow Sprite.

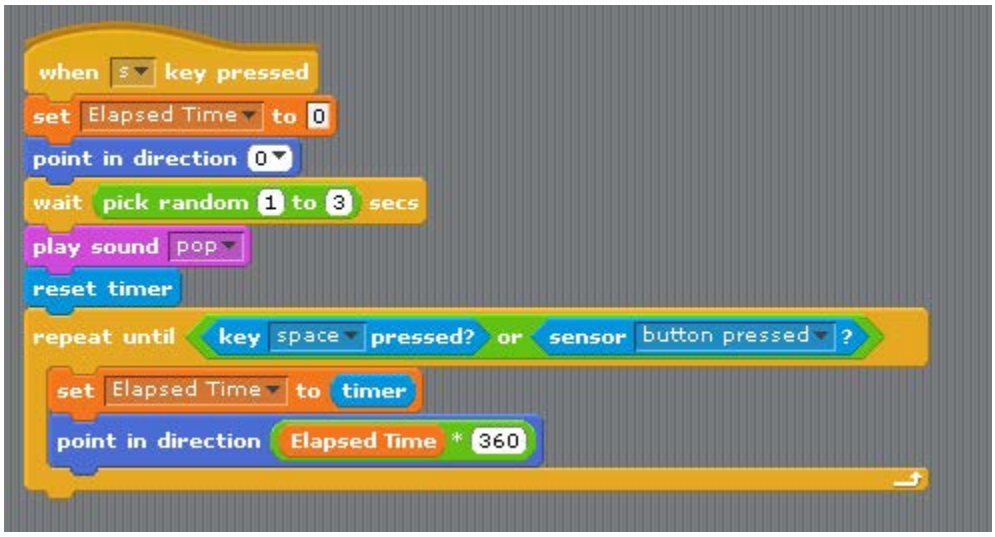
1. To have the arrow Sprite always move on the Clock Face, assemble the following scripts on the arrow Sprite: (The blue 'go to ClockFace' script sets the position of the arrow. The 'go to front' ensures that the arrow will draw above the ClockFace.)



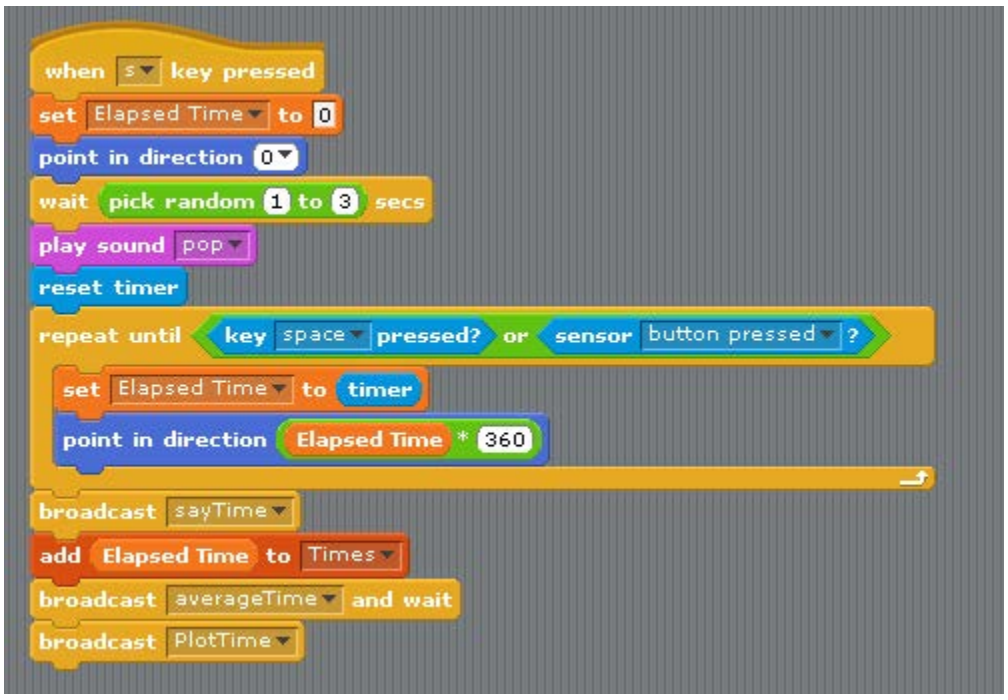
2. The next section of script acts as the core game logic for the reaction timer. This first piece will initialize the timing process. Using the 'when s key pressed' block, assemble the following scripts:
    a. Set 'Elapsed Time' to 0: (Resets this variable)
    b. Point the Arrow to 0 degrees (Represents '0' time)
    c. Wait 'pick random 1 to 3' seconds (Waits a random time to make a sound)
    d. Play the pop sound
    e. Reset the Scratch Timer (The Scratch Timer object always runs while a Scratch Program is open. This Timer can be reset to measure intervals of time within the Scratch Program.)



3. The next section uses a 'repeat until key spaced pressed or button pressed' to register the user input. During the repeat loop the Elapsed Time Variable is set to the Timer. The Arrow is also pointed to the Elasped time *360 to simulate the moving clock hand.

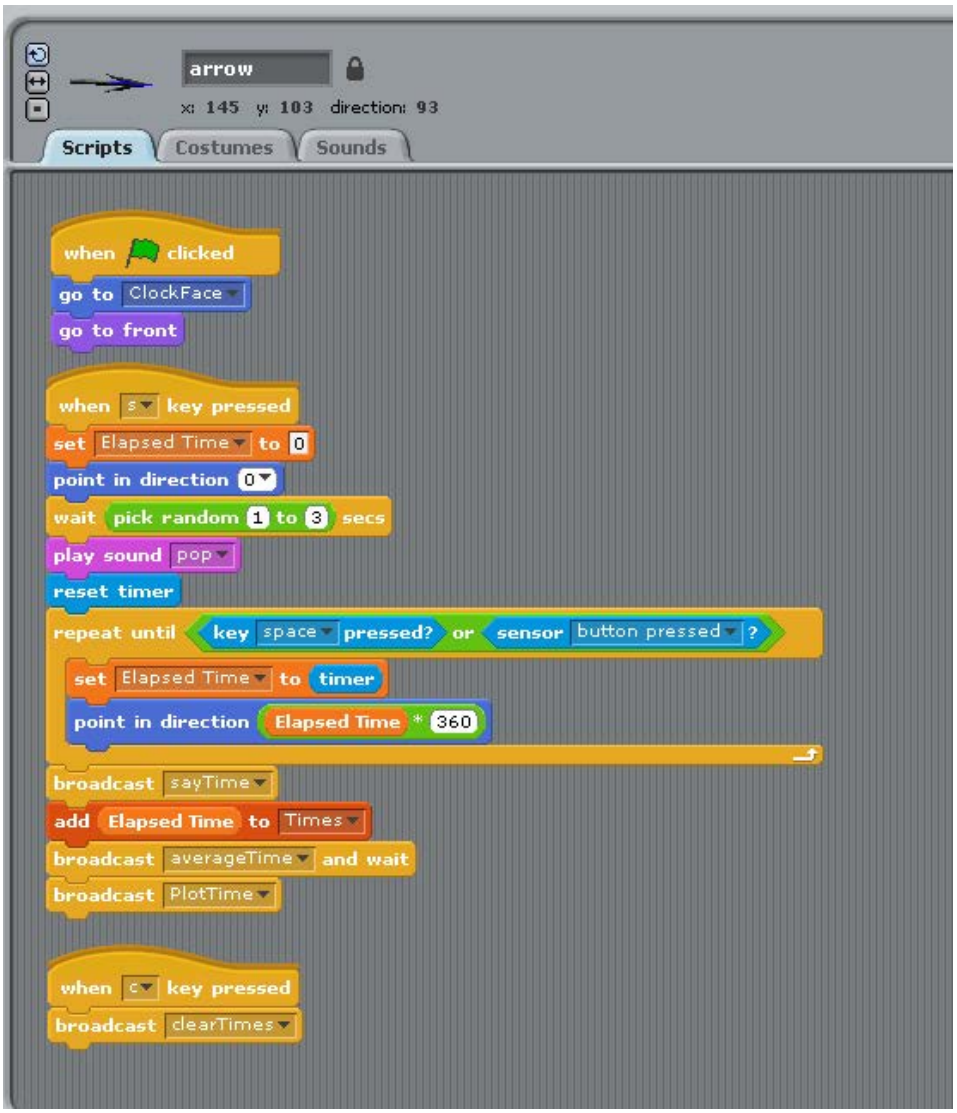


4. The final section of this script Broadcasts (like calling functions in other objects in the program) messages to say the resulting time, append the Times List with the Elapsed Time result, average the Times, and Plot the Times. (Note that these scripts do not 'perform' these operations. They only pass the messages to other sprites. We will complete these scripts later in the lesson.)



5. Finally, add a 'when c key pressed' script to the arrow Sprite to call a Broadcast to clearTimes. Again, this only calls the message. Later we will define what clearTimes does.

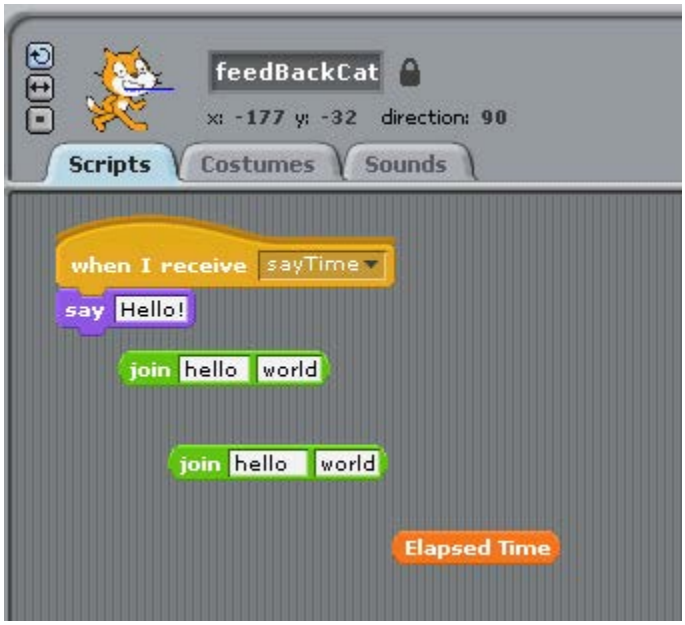
6. The entire Script for the arrow should look like this:



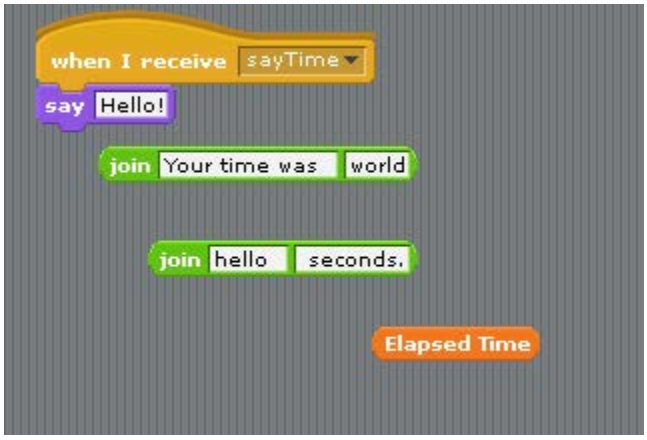
7. At this point, if you pressed the 's' key, the program would make a sound and the arrow would spin. When you press the space key or the button on the PicoBoard, the arrow will stop and you would see the variable indicating the Elapsed Time. We now need to add feedback for the user.

Part 3: Create the Scripts for the Cat

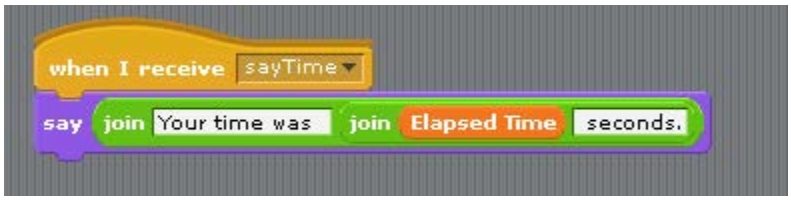
1. On the feedBackCat Sprite, place the following blocks. We will assemble these blocks to have the feedBackCat say the Elapsed Time value: (The join blocks will allow us to combine 'strings' of characters together.)



2. Change the values to the following in the join blocks:
   a. Change 'hello' to 'Your time was "
   b. Change 'world' to ' seconds.'

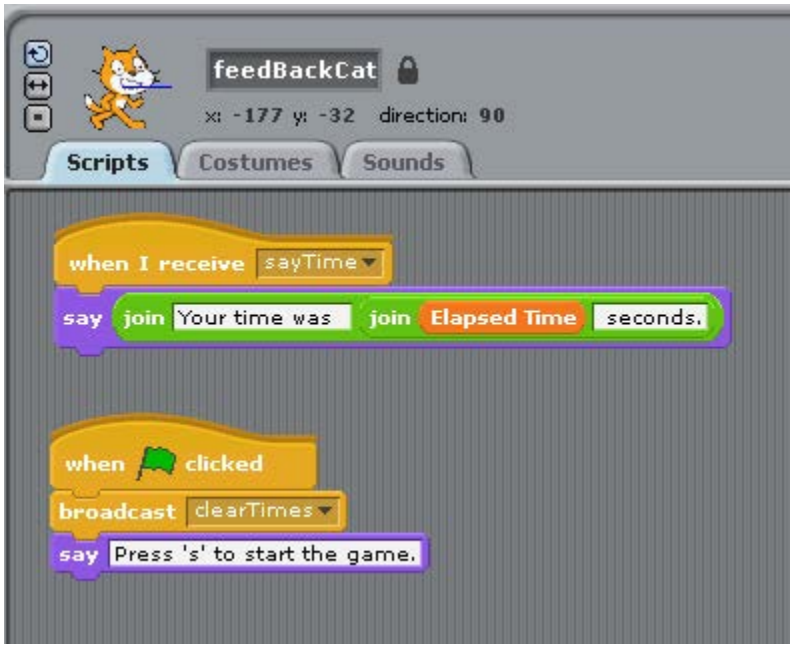


3. Combine the Blocks as follows:



4. Add this When Green Flag Clicked script to call the clearTimes message and Print the phrase 'Press 's' to start the game' to the screen.

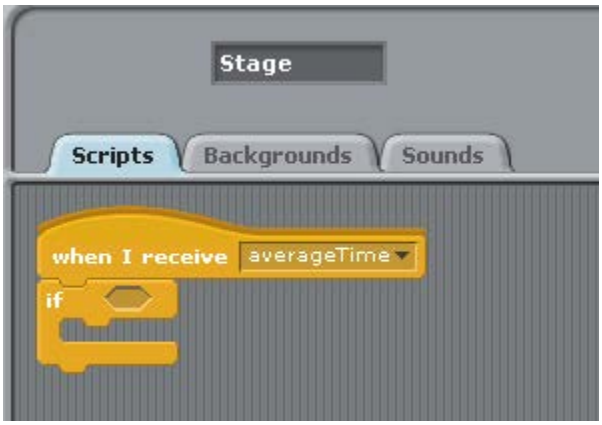
5. The Scripts for feedBackCat should look like this:



6. If you clicked the Green Flag and pressed 's', the game should start with a pop and the start of the clock. When you press the SpaceBar or the PicoBoard button, the cat should say your time.

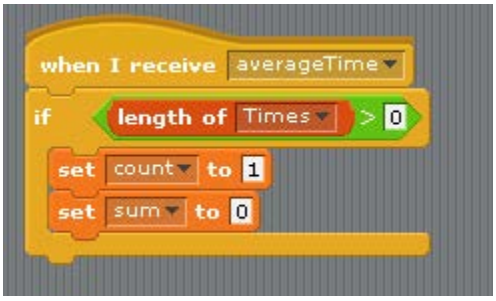
Part 4: Writing the Scripts for the Stage: Calculating the Average Reaction Times

1. Go to the Stage and put a 'when I receive 'averageTime'' and if statement in the Stage Scripts.

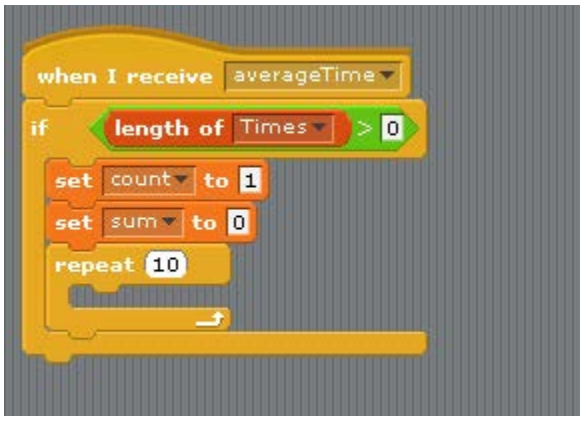


2. We will now add a 'length of Times' > 0 condition to the if statement. This will keep us from dividing by zero when calculating an average.

3. Inside the if statement, add the following blocks:
   a. set 'count' to 1 (Sets the count for the 'index variable in our loop')
   b. set 'sum' to 0 (This will be the sum of all the values in the Times List)

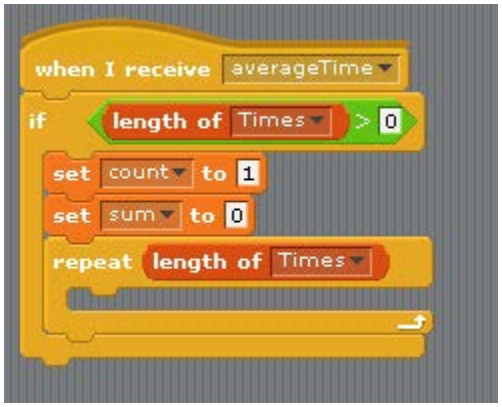


4. Put a repeat 10 block inside the if statement.

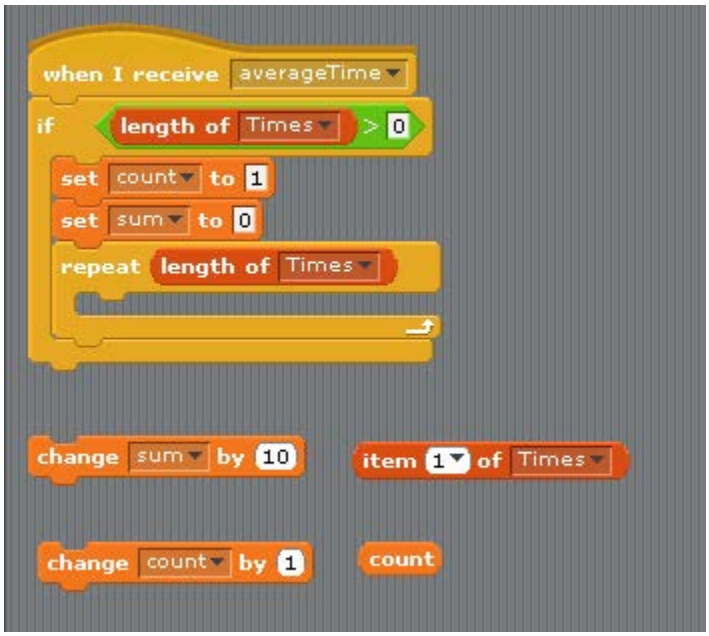


5. Put a 'length of Times' block inside the Repeat. This is like saying (in python):

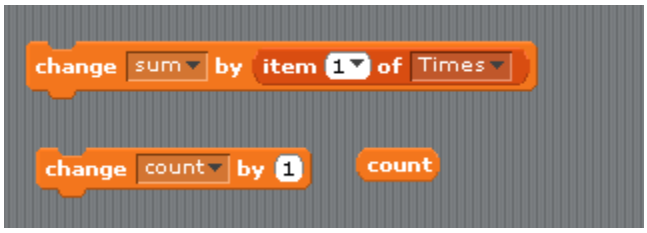
```
for count in range(1, len(Times)):
```



6. Put the following blocks in the space below the if statement (Do not connect yet)

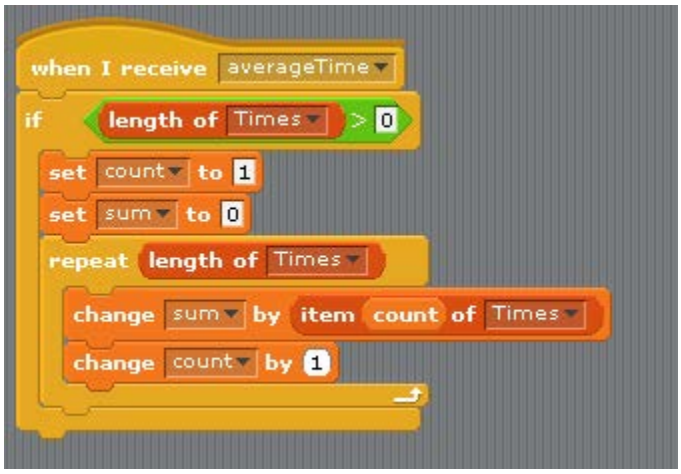


7. What we want to do here is add one Times list item at a time to the sum value. To do this, place the 'item 1 of Times' into the 'change sum by 10' block'.



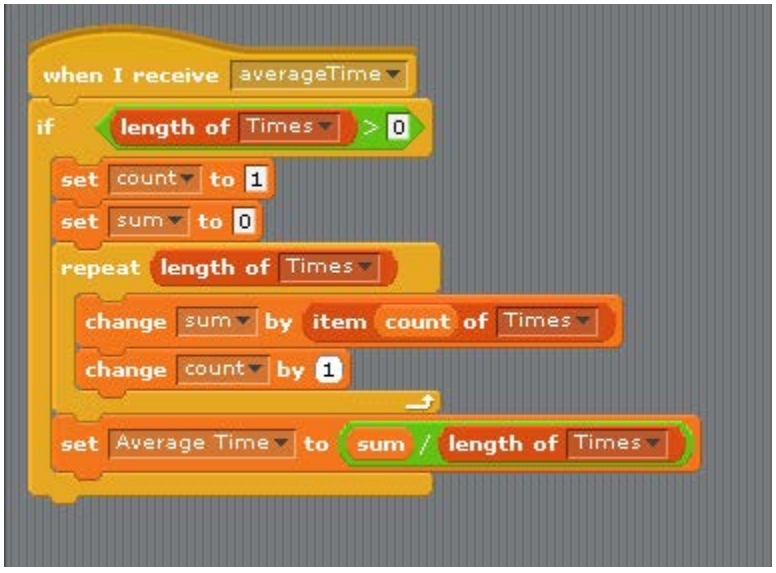
8. Place the 'count' (acting like an index variable) into the 'item of Times' block.

9. Place both these blocks into the ‘repeat length of Times’ block.



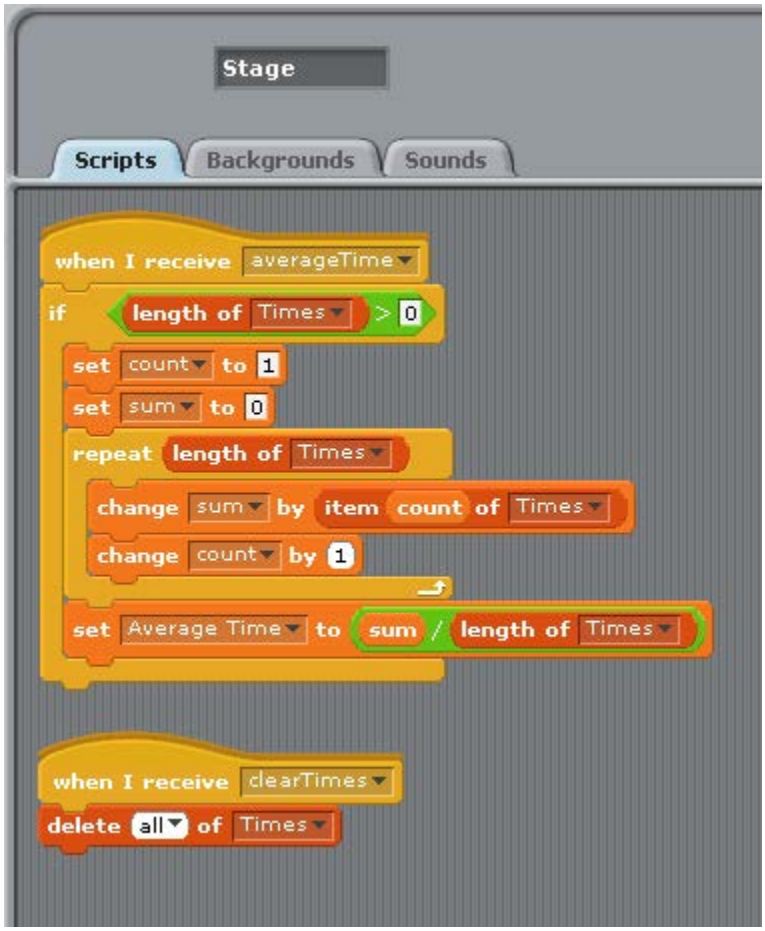
10. When this code runs, the sum variable will hold all the values in the List ‘Times’ added together. To Calculate the Average Time, we will devide the sum by the length of the Times list:

11. Place this block below the ‘repeat length of Times’ block and inside the if.



12. Add a 'When I receive 'clearTimes'' block to call the command to delete all values of the list when this message is broadcasted.

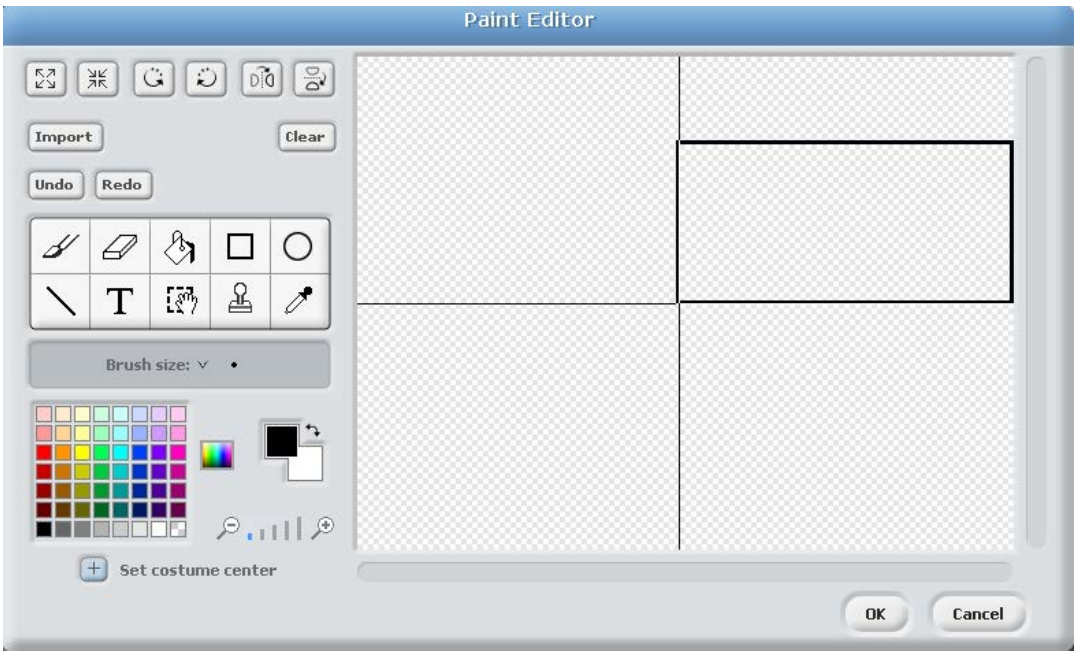
13. The entire script for the stage should look like this:



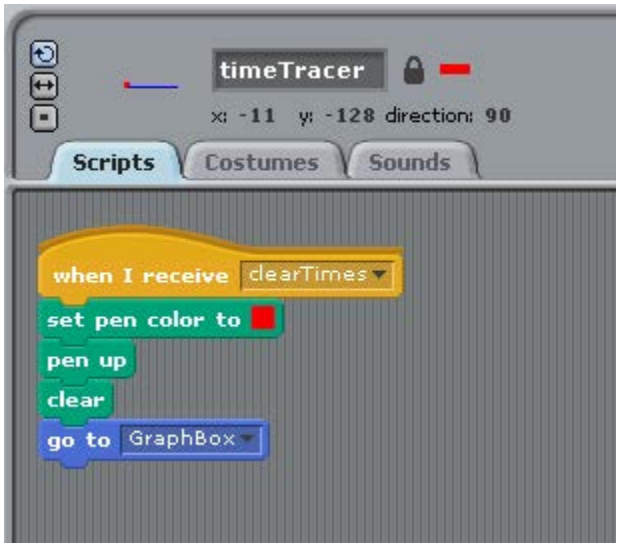
Part 5: Writing the Scripts for the time Tracer.

The timeTracer Sprite creates a line graph in the GraphBox representing the values of the reaction times stored in the Times List. The timeTracer uses the lower left hand corner of the GraphBox as the origin. (Most computer programming systems use the upper left hand corner.)

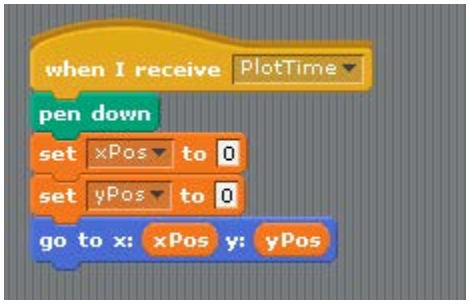
1. To start, go to the GraphBox and set the costume center to be the lower left hand corner.



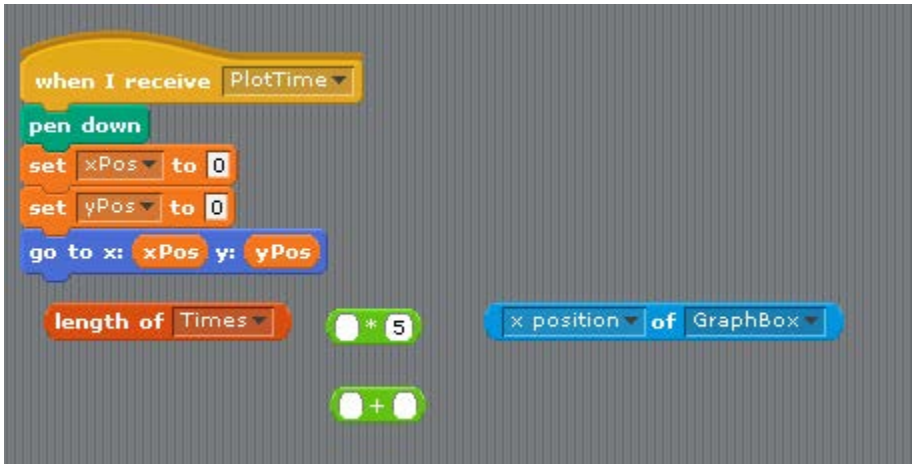
2. Now go to the timeTracer Sprite. Add the following Scripts to initialize the Sprite:
    a. 'when I receive 'clearTimes'' to listen to the message.
    b. Set pen color to red
    c. 'pen up' to stop the drawing process
    d. 'clear' to clear any graph from the screen
    e. 'go to GraphBox' to go back to the origin of the graph.



3. The 'PlotTime' message will advance the timeTracer to map out the next value in the list. Assemble the following scripts to start building the 'PlotTime' actions:

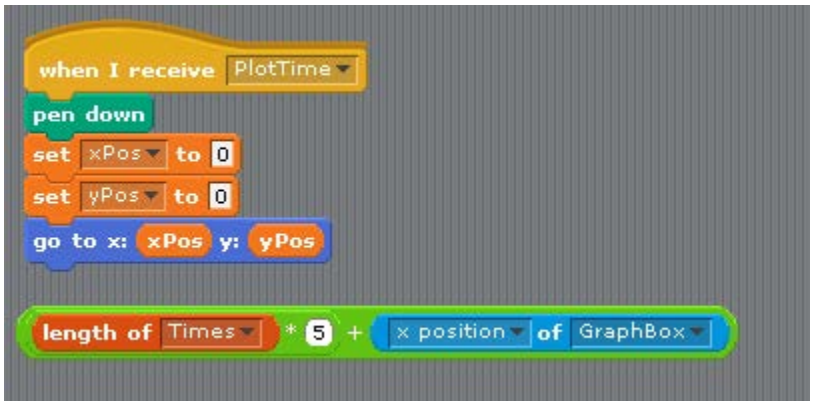


4. Put the following script blocks below but do not assemble:
    a. Length of Times (Total number of items in the Times List)
    b. Multiply block (*)
    c. 'x position of graphBox' from sensing
    d. Addition block (+)

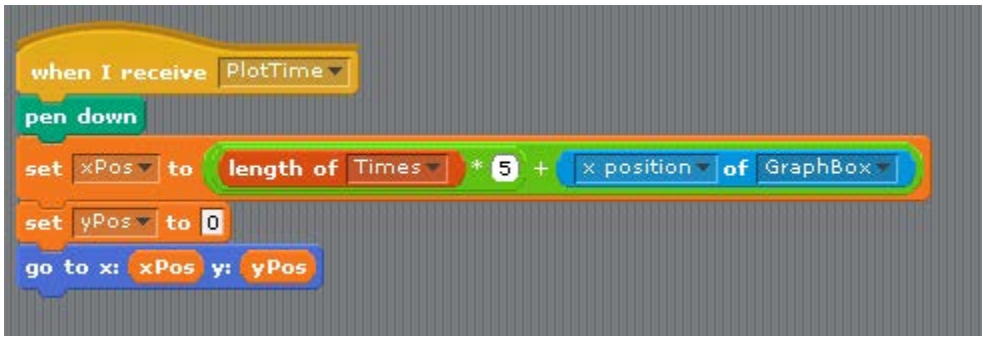


5. We will use these blocks to calculate the X position of the line graph point. The length of Times multiplied by 5 will describe the length of the intervals between data points in the graph (the x axis). The x position of the GraphBox will provide the starting point in order to have the line graph reside within the GraphBox. Assemble the Scripts as shown below:

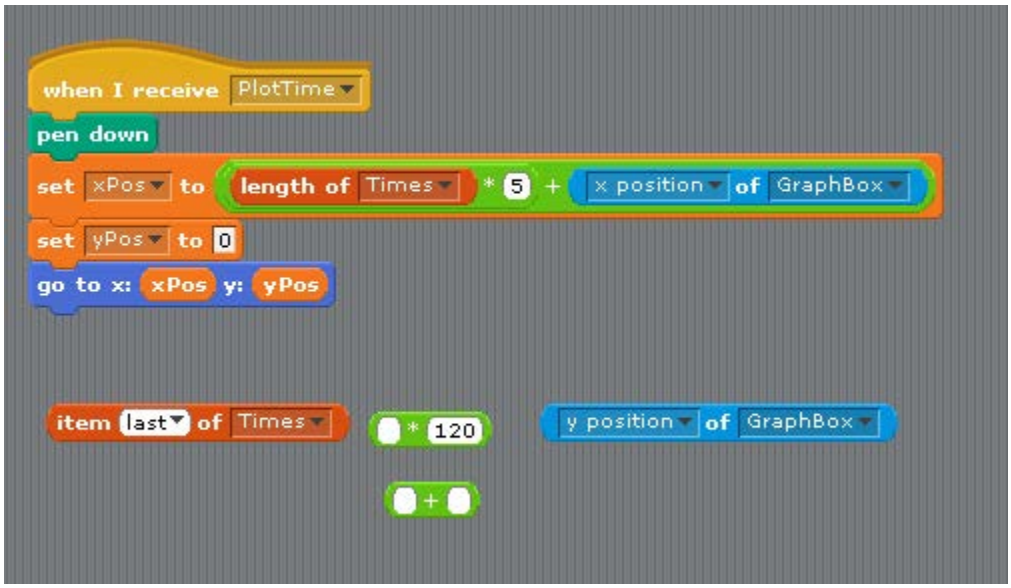
    ((length of Times) * 5) + (x position of GraphBox)



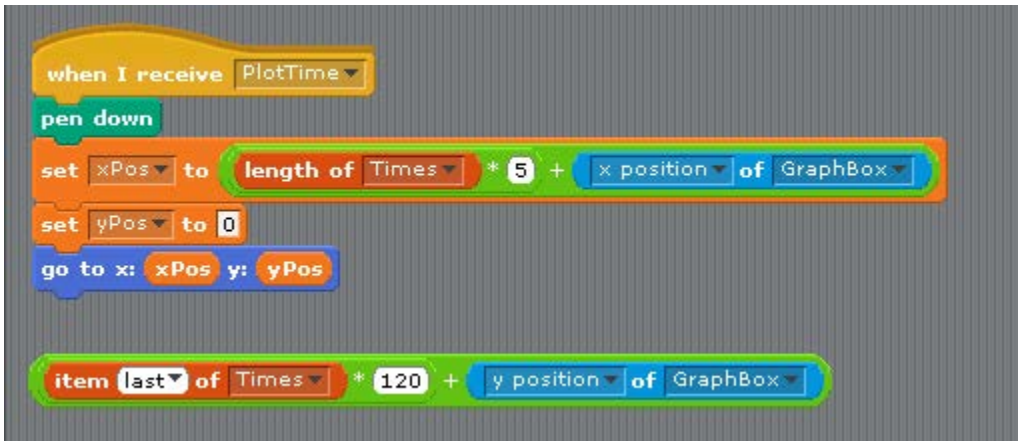
6. Place this block set into the ‘set xPos’ to 0 block:



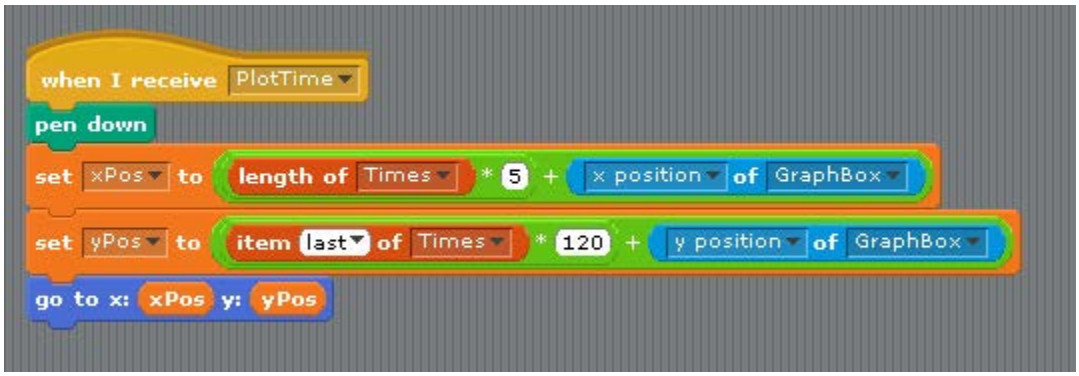
7. We will calculate the y Position in a similar manner. Place the following blocks below the script:



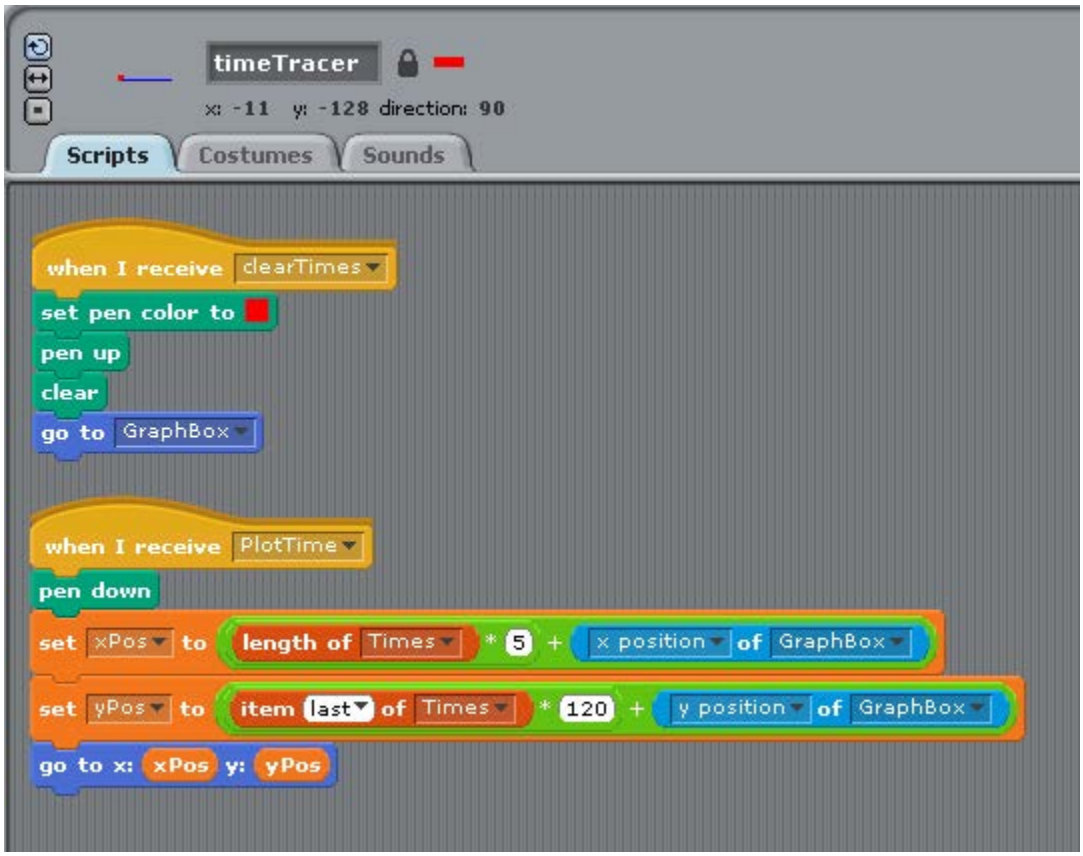
8. Using the last item of the Times List, we will calculate the Y Position. The number 120 comes from the height of the GraphBox so 1 second would be at the top of the box. Assemble the Scripts as shown here:



9. Place this block in the 'set yPos to 0' block:



10. The Entire timeTracer Scripts should look like this:

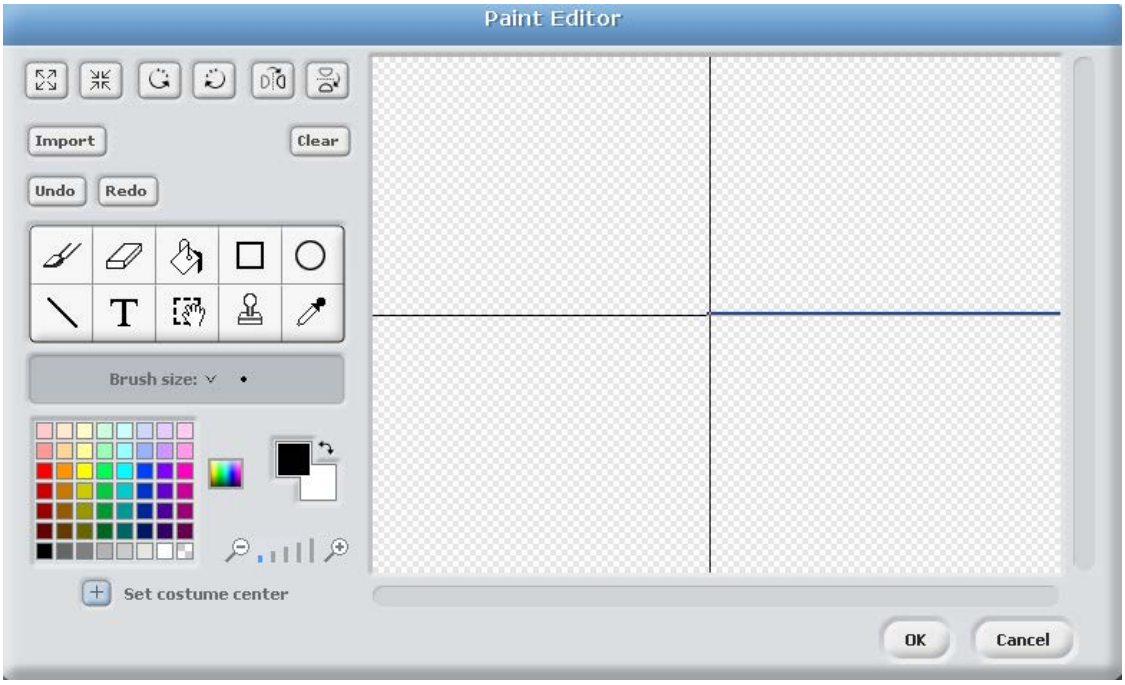


11. Click the Green Flag and then press the 's' key. The computer should 'pop' and then press the spacebar or the PicoBoard Button. The cat should then show the result and the first time spot should graph. If you continue to press 's' and to time trials, the graph should begin to show mapping your times.

Part 6: Writing the Scripts for the Average Line

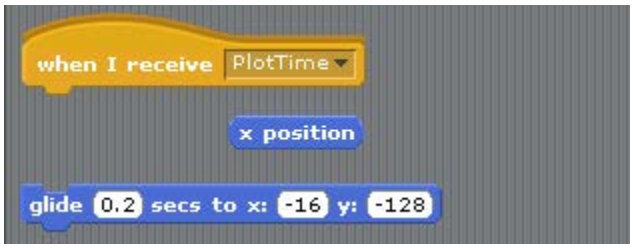
These scripts take the Average Time value from the Times list and use this value to position a blue line on the graph indicating the Average.

1. Make sure the costume center of the Average Sprite is on the left side:



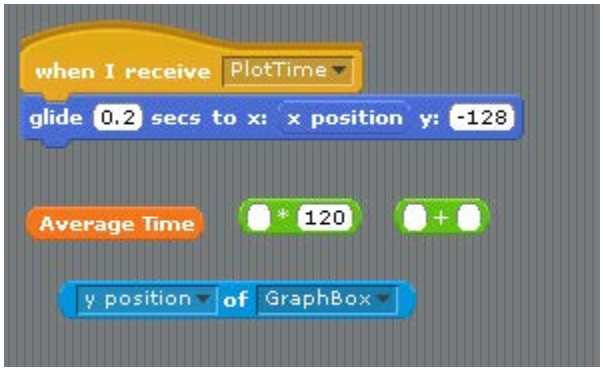
2. Place the Following Scripts to initialize the Average line when the 'clearTimes' message is broadcast.

3. Place the following Scripts:
    a. ‘when I receive ‘PlotTime’’
    b. Glide 0.2 secs to x: y:
    c. X Position

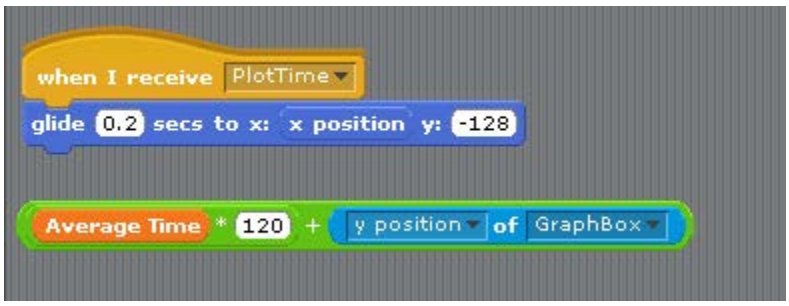


4. Assemble the Scripts as shown:

5. Bring the following Blocks into the scripts area:
    a. Average Time Variable
    b. Multiply Block
    c. Addition Block
    d. Y position of GraphBox from sensing

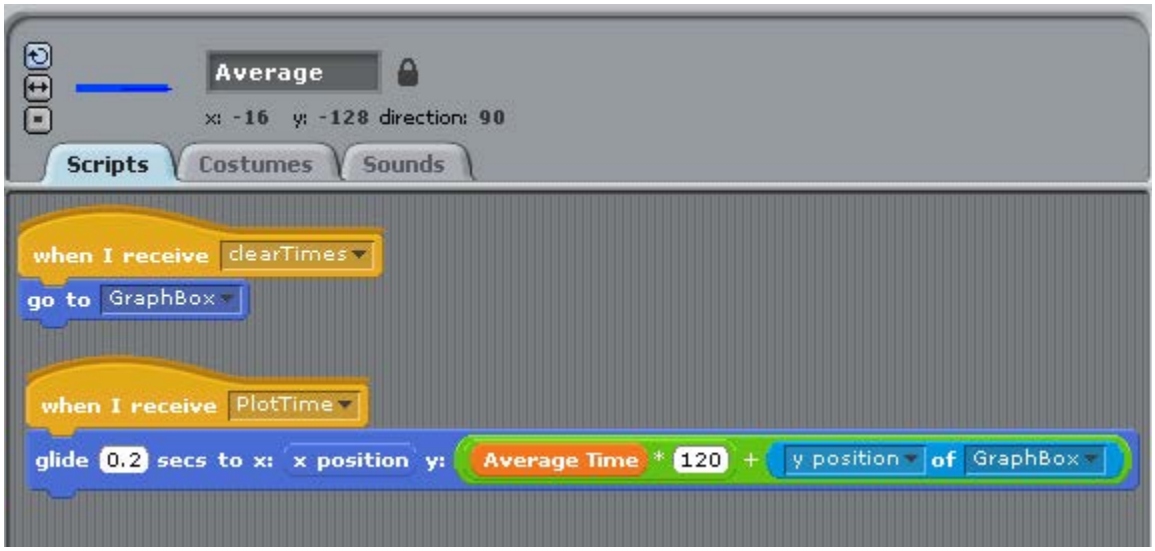


6. Assemble these blocks as shown to calculate the y position.



7. Place this script into the y: of the glide block:

8. The complete Script for the Average line should look like this:



9. All Done! Click the Green Flag and then press 's'. Repeated tries will create a graph of your reaction times with the average. Try using other controls on the PicoBoard to vary the game. (You could make a version of "Bop it" By having the computer ask the user to slide the slider, push the button, cover the light . . .) See this website for an example:

http://www.amazon.com/Parker-Brothers-7789-Bop-It/dp/B001RNC1GK/ref=sr_1_1?ie=UTF8&qid=1373074759&sr=8-1&keywords=bop+it