Jeffrey Greblick, Chandler Woo, and Youwen Xia

November 18, 2014

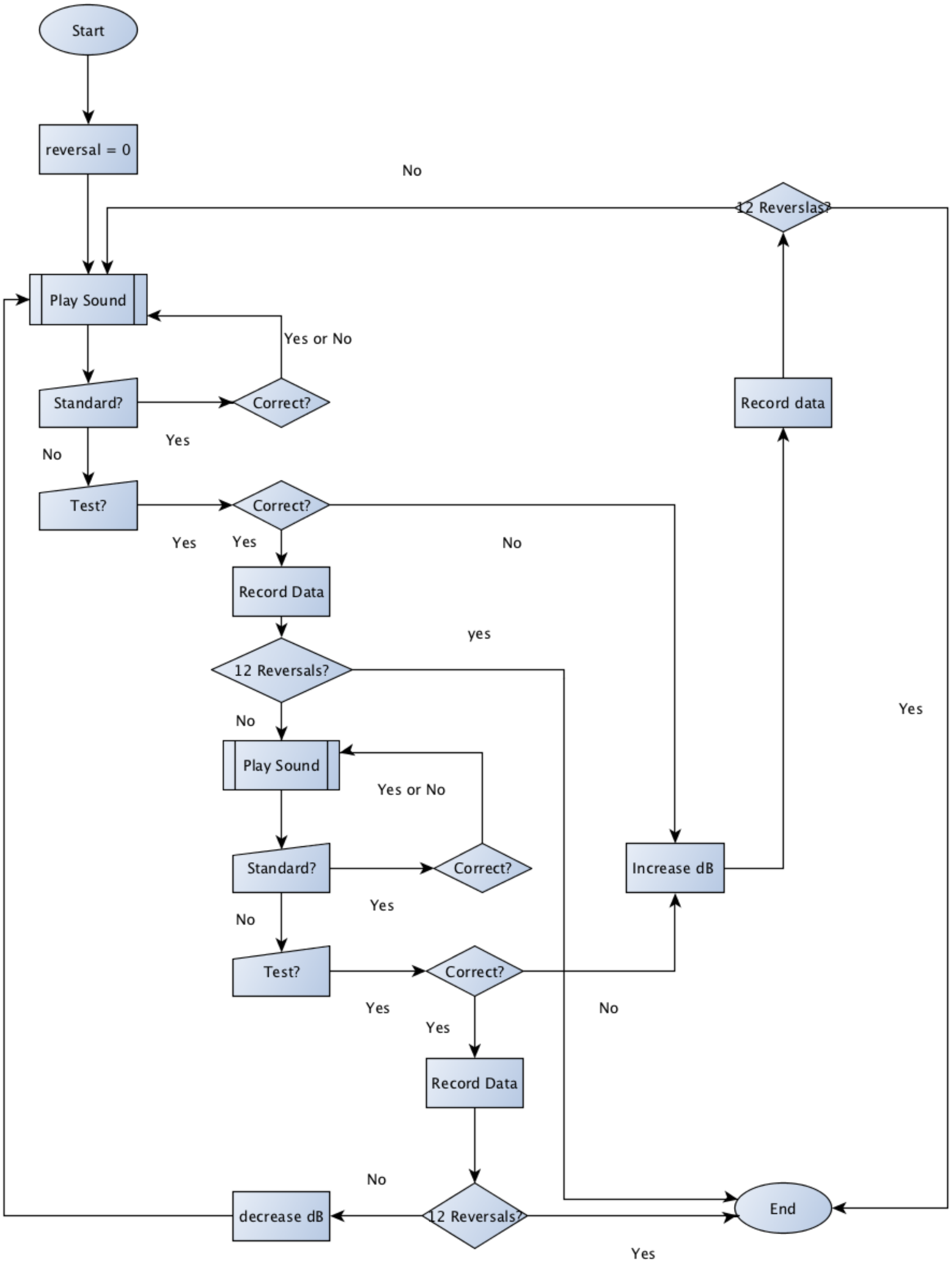
MATLAB Final Project

Children's Hearing Test

Children have been known to lose interest quickly when given menial tasks. This trend has always been a problem for doctors administering hearing tests that require repetitive data collection. When children lose focus or stop caring, the data may be corrupted leading to a misdiagnosis. This is a serious problem when dealing with children who have autism, as they frequently suffer from sensory disorders such as auditory dislaterality, inability to block out background noise, and hypersensitivity to sound (Waterhouse, 2014). However, an article from the *Journal of Autism and Developmental Disorders* produced some interesting findings regarding the attentional capacities of children with autism. When compared to a control group of children, it was discovered that children with autism struggled more than the average when asked to switch quickly between tasks. However, they were able to stay locked in and focused on a single repetitive task as good or better than an average child (Pascualvaca, 1998). Because the hearing test we have been asked to create is also repetitive, an autistic child should be able to produce accurate results, as long as the program runs properly and keeps the child adequately engaged.

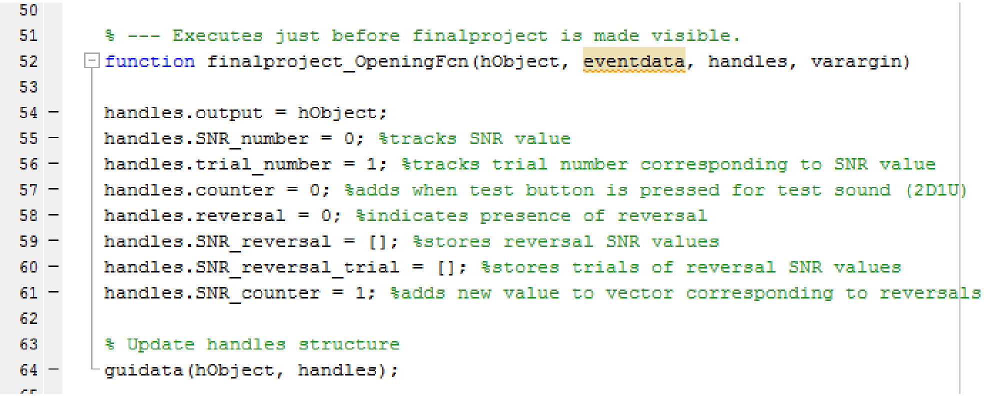
The hearing test we were asked to generate had to comprise of two different sounds, one a short burst of noise at a standard decibel level (standard noise). The second must be a combination of the standard noise and a sinusoidal wave tone (test noise). The test subject must decide if they think the noise they hear is the standard noise or the test noise. At first, the subject should be able to differentiate the two easily because the ratio of the sinusoidal wave tone to the standard noise will be one to one. As the user continues to correctly identify the test noise, the sinusoidal wave tone will decrease by one decibel, for every two correct responses. After a number of trials the test noise will become increasingly difficult to identify. When the subject fails to recognize the test noise, then the decibel will increase by one. This type of algorithm is called a 2 down 1 up (2D1U) setup and is a transformation of a simple up or down staircase method (Levitt, 1971). Eventually, the test noise and the standard noise will become indiscernible by the test subject and their correct and incorrect responses will begin to oscillate. This oscillation is called a sequence of reversals, and by finding the average decibel at which the correct responses in the oscillation occur, we will be able to find the "threshold of discrimination." This threshold determines the child's hearing abilities. We must then produce a plot of the decibel level vs. accurate responses, with the threshold of discrimination labeled.

Flow Chart


Start
reversal = 0
No
12 Reverslas?
Play Sound
Yes or No
Standard?
Correct?
Record data
Yes
No
Test?
Correct?
Yes
Yes
No
Record Data
yes
12 Reversals?
Yes
No
Play Sound
Yes or No
Standard?
Correct?
Increase dB
Yes
No
Test?
Correct?
Yes
Yes
No
Record Data
No
decrease dB
12 Reversals?
End
Yes

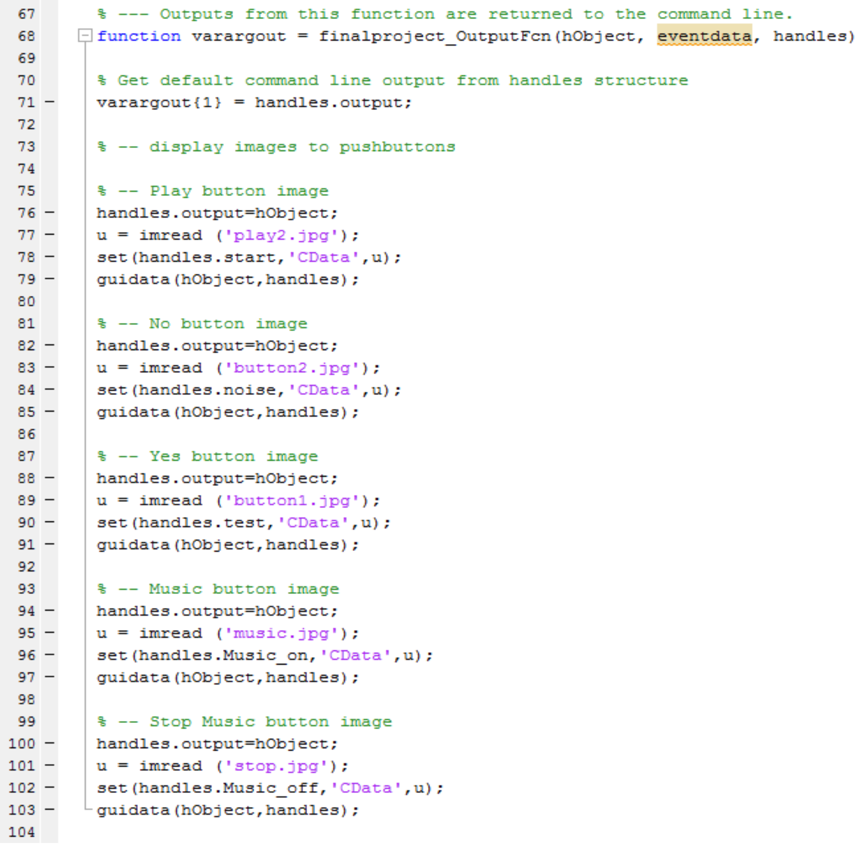
Code

Preset Variables (Note that brackets indicate vectors and handles.trial_number and handles.SNR_counter are 1 because MATLAB has the 1 indice rather than a 0 indice)



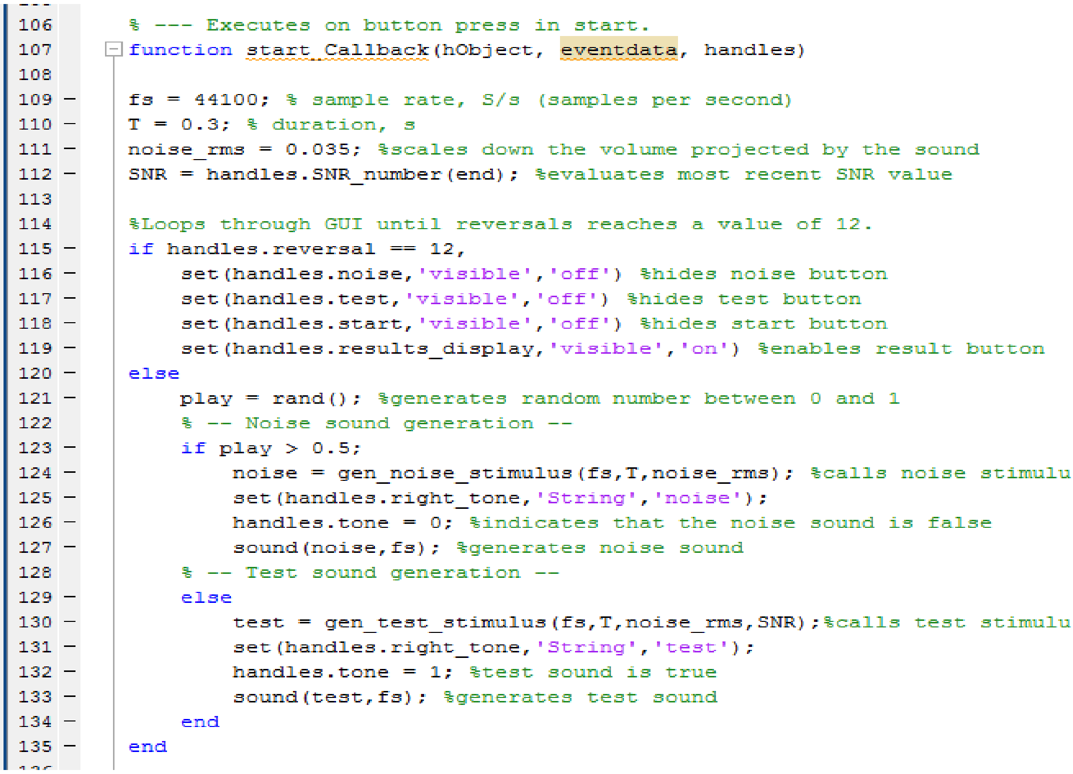
```
% --- Executes just before finalproject is made visible.
function finalproject_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;
handles.SNR_number = 0; %tracks SNR value
handles.trial_number = 1; %tracks trial number corresponding to SNR value
handles.counter = 0; %adds when test button is pressed for test sound (2D1U)
handles.reversal = 0; %indicates presence of reversal
handles.SNR_reversal = []; %stores reversal SNR values
handles.SNR_reversal_trial = []; %stores trials of reversal SNR values
handles.SNR_counter = 1; %adds new value to vector corresponding to reversals

% Update handles structure
guidata(hObject, handles);
```

Output variables to display image within pushbutton



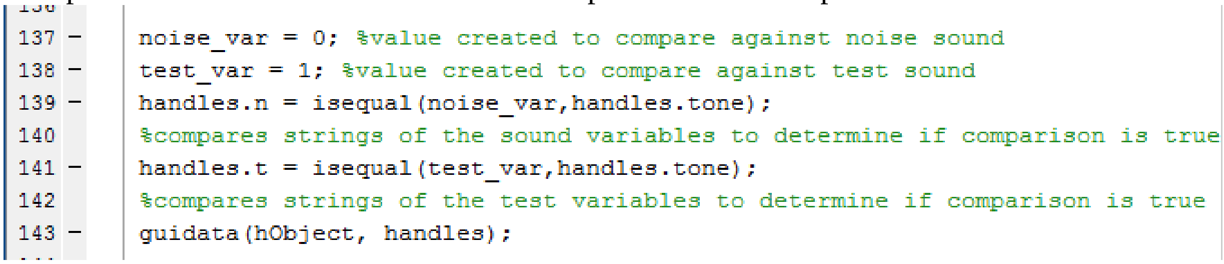
```
% --- Outputs from this function are returned to the command line.
function varargout = finalproject_OutputFcn(hObject, eventdata, handles)

% Get default command line output from handles structure
varargout{1} = handles.output;

% -- display images to pushbuttons

% -- Play button image
handles.output=hObject;
u = imread ('play2.jpg');
set(handles.start,'CData',u);
guidata(hObject,handles);

% -- No button image
handles.output=hObject;
u = imread ('button2.jpg');
set(handles.noise,'CData',u);
guidata(hObject,handles);

% -- Yes button image
handles.output=hObject;
u = imread ('button1.jpg');
set(handles.test,'CData',u);
guidata(hObject,handles);

% -- Music button image
handles.output=hObject;
u = imread ('music.jpg');
set(handles.Music_on,'CData',u);
guidata(hObject,handles);

% -- Stop Music button image
handles.output=hObject;
u = imread ('stop.jpg');
set(handles.Music_off,'CData',u);
guidata(hObject,handles);
```

Initiation of Reversal Condition and Noise Generation



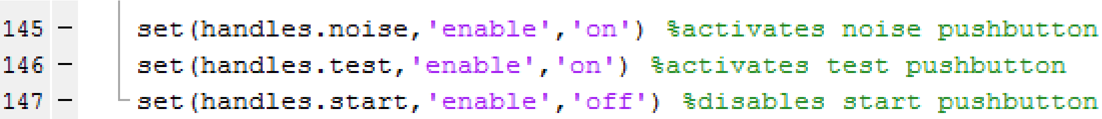
```
% --- Executes on button press in start.
function start_Callback(hObject, eventdata, handles)

fs = 44100; % sample rate, S/s (samples per second)
T = 0.3; % duration, s
noise_rms = 0.035; %scales down the volume projected by the sound
SNR = handles.SNR_number(end); %evaluates most recent SNR value

%Loops through GUI until reversals reaches a value of 12.
if handles.reversal == 12,
    set(handles.noise,'visible','off') %hides noise button
    set(handles.test,'visible','off') %hides test button
    set(handles.start,'visible','off') %hides start button
    set(handles.results_display,'visible','on') %enables result button
else
    play = rand(); %generates random number between 0 and 1
    % -- Noise sound generation --
    if play > 0.5;
        noise = gen_noise_stimulus(fs,T,noise_rms); %calls noise stimulus
        set(handles.right_tone,'String','noise');
        handles.tone = 0; %indicates that the noise sound is false
        sound(noise,fs); %generates noise sound
    % -- Test sound generation --
    else
        test = gen_test_stimulus(fs,T,noise_rms,SNR);%calls test stimulus
        set(handles.right_tone,'String','test');
        handles.tone = 1; %test sound is true
        sound(test,fs); %generates test sound
    end
end
```

Compares between noise/test sound and user-pressed noise/test pushbutton



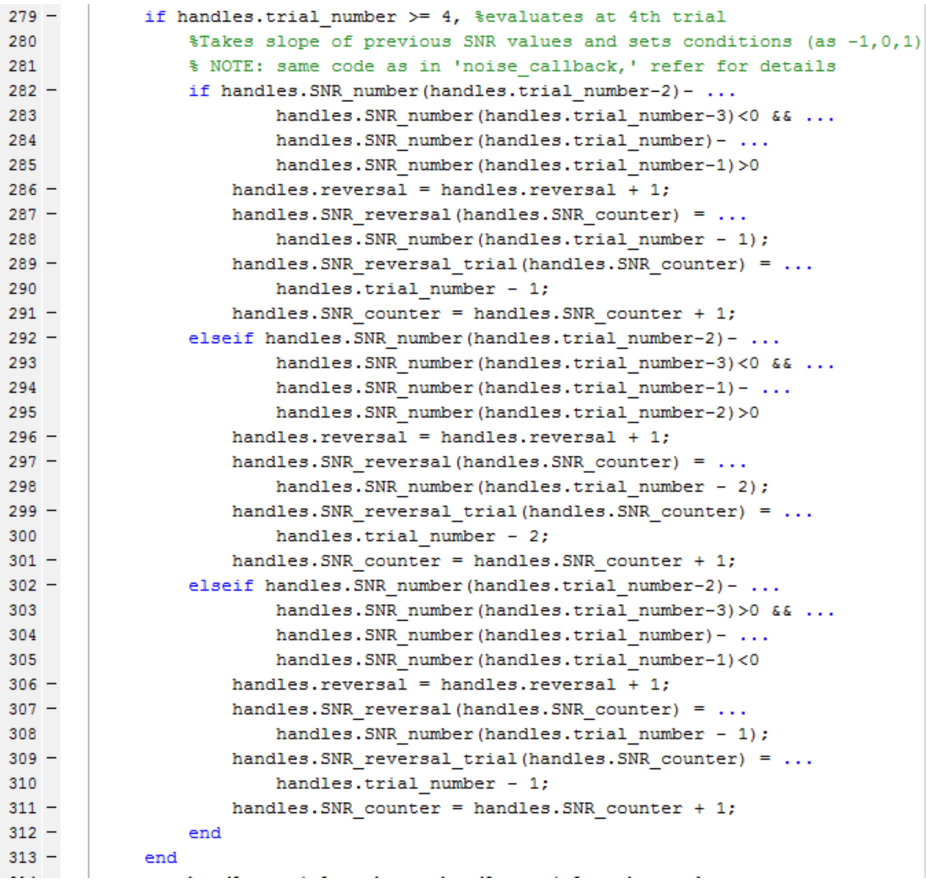
```
noise_var = 0; %value created to compare against noise sound
test_var = 1; %value created to compare against test sound
handles.n = isequal(noise_var,handles.tone);
%compares strings of the sound variables to determine if comparison is true
handles.t = isequal(test_var,handles.tone);
%compares strings of the test variables to determine if comparison is true
guidata(hObject, handles);
```

Enables/Disables buttons so user cannot press sound or noise/test pushbutton twice



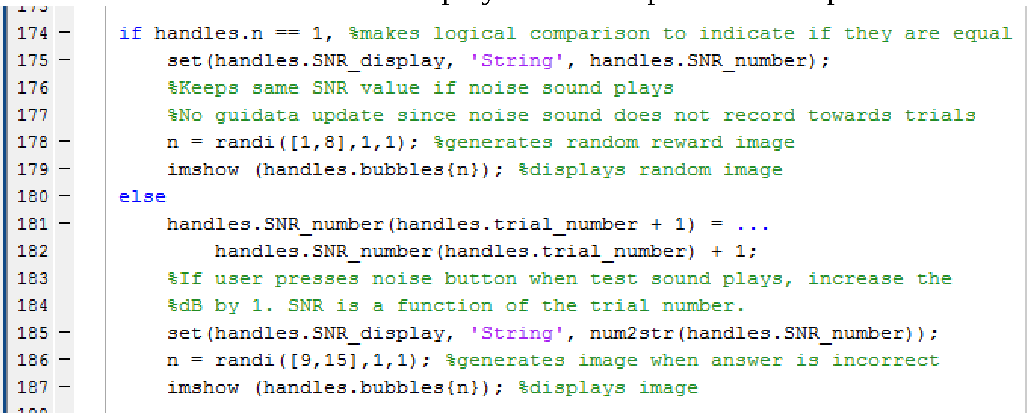
```
set(handles.noise,'enable','on') %activates noise pushbutton
set(handles.test,'enable','on') %activates test pushbutton
set(handles.start,'enable','off') %disables start pushbutton
```

Reversal Counter and Conditions



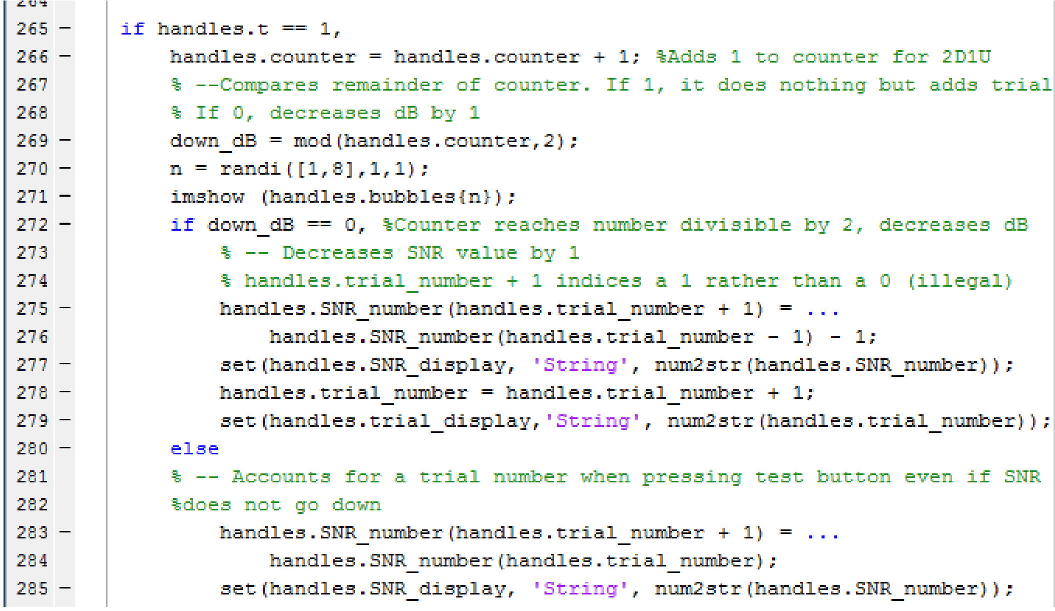
```
if handles.trial_number >= 4, %evaluates at 4th trial
    %Takes slope of previous SNR values and sets conditions (as -1,0,1)
    % NOTE: same code as in 'noise_callback,' refer for details
    if handles.SNR_number(handles.trial_number-2)- ...
            handles.SNR_number(handles.trial_number-3)<0 && ...
            handles.SNR_number(handles.trial_number)- ...
            handles.SNR_number(handles.trial_number-1)>0
        handles.reversal = handles.reversal + 1;
        handles.SNR_reversal(handles.SNR_counter) = ...
            handles.SNR_number(handles.trial_number - 1);
        handles.SNR_reversal_trial(handles.SNR_counter) = ...
            handles.trial_number - 1;
        handles.SNR_counter = handles.SNR_counter + 1;
    elseif handles.SNR_number(handles.trial_number-2)- ...
            handles.SNR_number(handles.trial_number-3)<0 && ...
            handles.SNR_number(handles.trial_number-1)- ...
            handles.SNR_number(handles.trial_number-2)>0
        handles.reversal = handles.reversal + 1;
        handles.SNR_reversal(handles.SNR_counter) = ...
            handles.SNR_number(handles.trial_number - 2);
        handles.SNR_reversal_trial(handles.SNR_counter) = ...
            handles.trial_number - 2;
        handles.SNR_counter = handles.SNR_counter + 1;
    elseif handles.SNR_number(handles.trial_number-2)- ...
            handles.SNR_number(handles.trial_number-3)>0 && ...
            handles.SNR_number(handles.trial_number)- ...
            handles.SNR_number(handles.trial_number-1)<0
        handles.reversal = handles.reversal + 1;
        handles.SNR_reversal(handles.SNR_counter) = ...
            handles.SNR_number(handles.trial_number - 1);
        handles.SNR_reversal_trial(handles.SNR_counter) = ...
            handles.trial_number - 1;
        handles.SNR_counter = handles.SNR_counter + 1;
    end
end
```

Increases dB of SNR if test sound plays and noise pushbutton is pressed



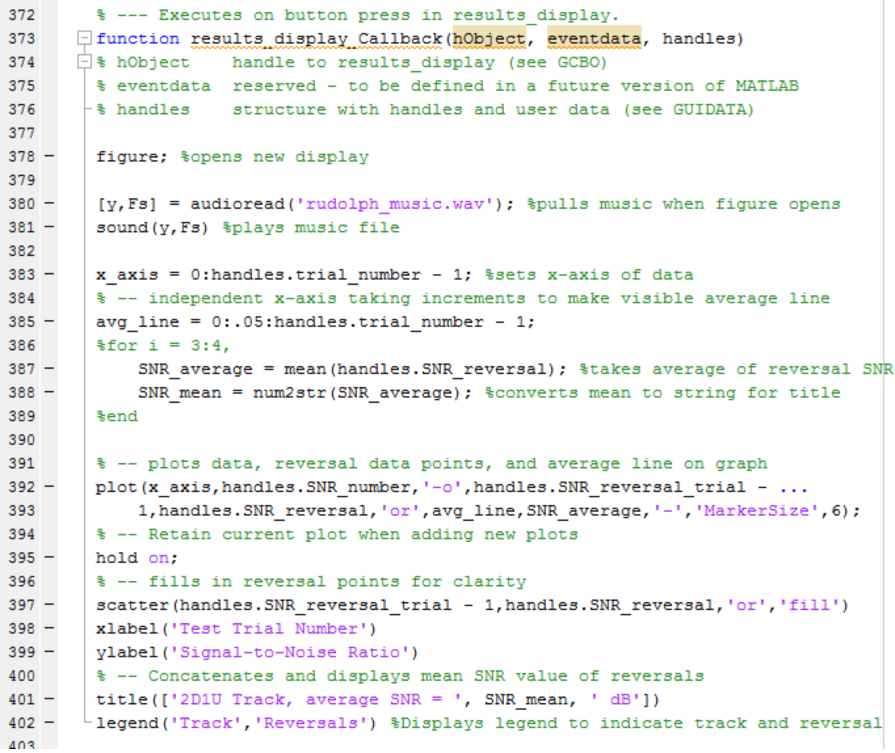
```
if handles.n == 1, %makes logical comparison to indicate if they are equal
    set(handles.SNR_display, 'String', handles.SNR_number);
    %Keeps same SNR value if noise sound plays
    %No guidata update since noise sound does not record towards trials
    n = randi([1,8],1,1); %generates random reward image
    imshow (handles.bubbles{n}); %displays random image
else
    handles.SNR_number(handles.trial_number + 1) = ...
        handles.SNR_number(handles.trial_number) + 1;
    %If user presses noise button when test sound plays, increase the
    %dB by 1. SNR is a function of the trial number.
    set(handles.SNR_display, 'String', num2str(handles.SNR_number));
    n = randi([9,15],1,1); %generates image when answer is incorrect
    imshow (handles.bubbles{n}); %displays image
```

2D1U operator



```
if handles.t == 1,
    handles.counter = handles.counter + 1; %Adds 1 to counter for 2D1U
    % --Compares remainder of counter. If 1, it does nothing but adds trial
    % If 0, decreases dB by 1
    down_dB = mod(handles.counter,2);
    n = randi([1,8],1,1);
    imshow (handles.bubbles{n});
    if down_dB == 0, %Counter reaches number divisible by 2, decreases dB
        % -- Decreases SNR value by 1
        % handles.trial_number + 1 indices a 1 rather than a 0 (illegal)
        handles.SNR_number(handles.trial_number + 1) = ...
            handles.SNR_number(handles.trial_number - 1) - 1;
        set(handles.SNR_display, 'String', num2str(handles.SNR_number));
        handles.trial_number = handles.trial_number + 1;
        set(handles.trial_display,'String', num2str(handles.trial_number));
    else
    % -- Accounts for a trial number when pressing test button even if SNR
    %does not go down
        handles.SNR_number(handles.trial_number + 1) = ...
            handles.SNR_number(handles.trial_number);
        set(handles.SNR_display, 'String', num2str(handles.SNR_number));
```

Graph figure code



```
% --- Executes on button press in results_display.
function results_display_Callback(hObject, eventdata, handles)
% hObject    handle to results_display (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

figure; %opens new display

[y,Fs] = audioread('rudolph_music.wav'); %pulls music when figure opens
sound(y,Fs) %plays music file

x_axis = 0:handles.trial_number - 1; %sets x-axis of data
% -- independent x-axis taking increments to make visible average line
avg_line = 0:.05:handles.trial_number - 1;
%for i = 3:4,
    SNR_average = mean(handles.SNR_reversal); %takes average of reversal SNRs
    SNR_mean = num2str(SNR_average); %converts mean to string for title
%end

% -- plots data, reversal data points, and average line on graph
plot(x_axis,handles.SNR_number,'-o',handles.SNR_reversal_trial - ...
    1,handles.SNR_reversal,'or',avg_line,SNR_average,'-','MarkerSize',6);
% -- Retain current plot when adding new plots
hold on;
% -- fills in reversal points for clarity
scatter(handles.SNR_reversal_trial - 1,handles.SNR_reversal,'or','fill')
xlabel('Test Trial Number')
ylabel('Signal-to-Noise Ratio')
% -- Concatenates and displays mean SNR value of reversals
title(['2D1U Track, average SNR = ', SNR_mean, ' dB'])
legend('Track','Reversals') %Displays legend to indicate track and reversal
```

Limitations and Potential Improvements

As a group, we struggled on a few areas of the code. The first area we struggled on was randomizing the sound. It took some time to understand the syntax of the rand() function in the context of the GUI. The next area we struggled on was creating an algorithm that produced results depending solely on responses to the test noise. This essentially meant that responses to the standard noise were irrelevant and actually simplified some of the code in the end. Proceeding on, we met our next roadblock with the 2D1U setup. 2D1U requires the test subject to correctly identify the test noise twice in a row in order to decrease the wave tone a decibel. If the user chooses the standard noise when the test noise is playing, the decibel will increase by one. Plotting the data was the final hurdle in the coding process. Once the code had been written for the 2D1U reversals, generating the plot was fairly straight forward, but ensuring that only the final six reversals were used to calculate the threshold of discrimination was not so simple. Looking back, we could improve our code by condensing it quite a bit. There are instances of repetitive code that could be shortened. Also we could have used global variables rather than using all handles. This would have made the project easier to follow for a programmer reading our code.

Works Cited

H. Levitt, Transformed Up-Down Methods in Psychoacoustics, J. Acoust. Soc. Amer., 1971, Vol. 49, 469–472.

Pascualvaca D., Fantie B., Papageorgiou M., “Attentional Capacities in Children with Autism: Is There a General Deficit in Shifting Focus?” *Journal of Autism and Develpmental Disorders,* Vol. 28, No.6, Dec. 1998

Waterhouse, Stella. “Sensory Disorder” *autisimtoday.com*, Nov. 13, 2014.