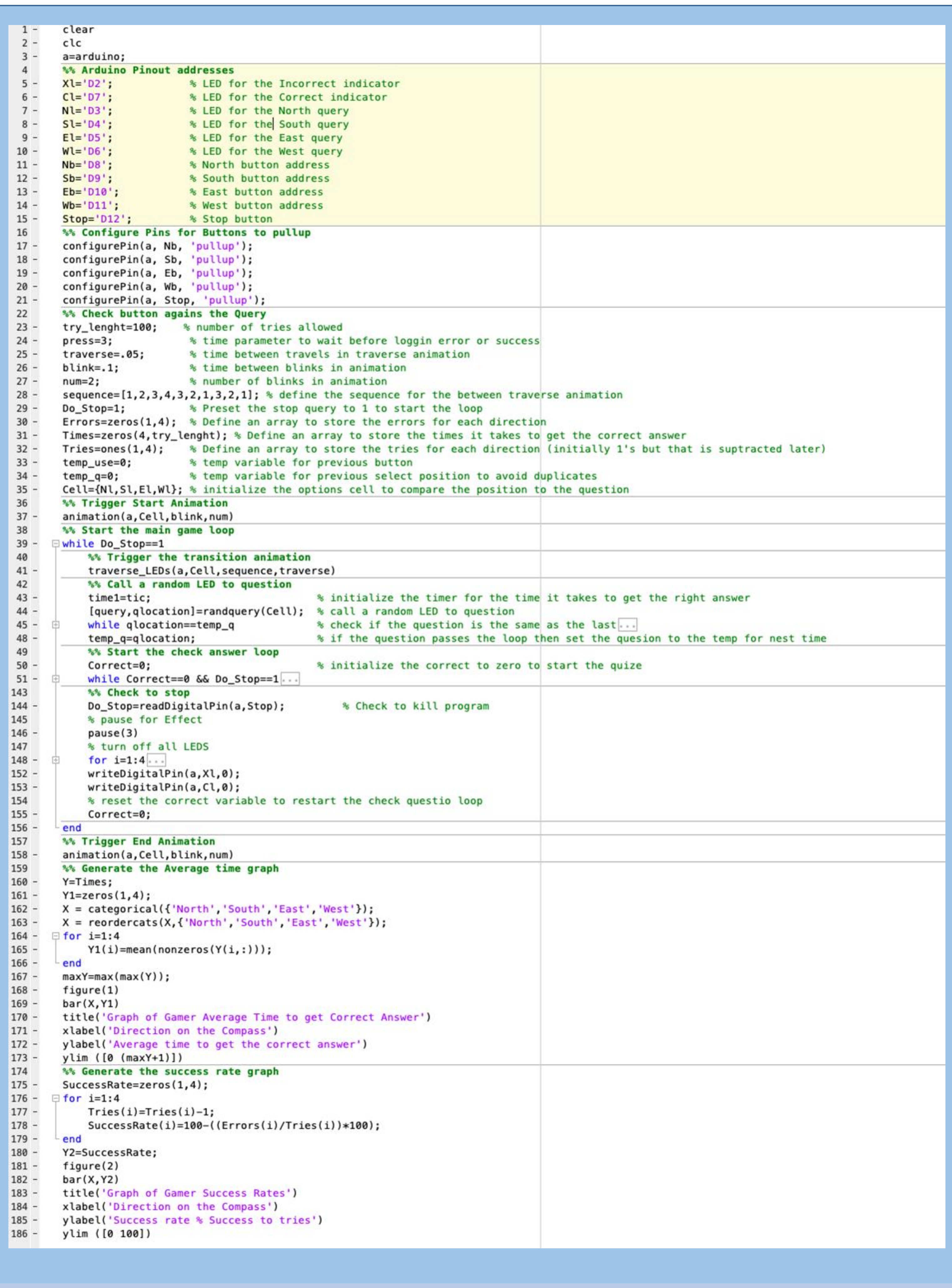
# Compass Game

Michael OHara
Tolga Kaya

ENGR 200 - Computational Methods in Engineering

## Abstract

The Compass Game is a device designed to help Kindergarteners learn how to read compass directions. The base has four LEDs with symbols for each direction on them to indicate which direction to find. There are also two LEDs to indicate if the provided answer is correct or incorrect. Next to the LEDs is a button that stops the game. The highlight of the game is the select dial and the compass face. This is where the gamer chooses his or her answer to the question by turning the dial. The game is designed to track how long it takes the gamer to get the correct answer. The controller is an Arduino that is connected to MATLAB to display gamer stats. And the end of the game, MATLAB displays the average time the gamer took to get the correct answer and his or her success rate per question.



```matlab
clear
clc
a=arduino;
%% Arduino Pinout addresses
Xl='D2';             % LED for the Incorrect indicator
Cl='D7';             % LED for the Correct indicator
Nl='D3';             % LED for the North query
Sl='D4';             % LED for the South query
El='D5';             % LED for the East query
Wl='D6';             % LED for the West query
Nb='D8';             % North button address
Sb='D9';             % South button address
Eb='D10';            % East button address
Wb='D11';            % West button address
Stop='D12';          % Stop button
%% Configure Pins for Buttons to pullup
configurePin(a, Nb, 'pullup');
configurePin(a, Sb, 'pullup');
configurePin(a, Eb, 'pullup');
configurePin(a, Wb, 'pullup');
configurePin(a, Stop, 'pullup');
%% Check button agains the Query
try_lenght=100;     % number of tries allowed
press=3;             % time parameter to wait before loggin error or success
traverse=.05;        % time between travels in traverse animation
blink=.1;            % time between blinks in animation
num=2;               % number of blinks in animation
sequence=[1,2,3,4,3,2,1,3,2,1]; % define the sequence for the between traverse animation
Do_Stop=1;           % Preset the stop query to 1 to start the loop
Errors=zeros(1,4);  % Define an array to store the errors for each direction
Times=zeros(4,try_lenght); % Define an array to store the times it takes to get the correct answer
Tries=ones(1,4);     % Define an array to store the tries for each direction (initially 1's but that is suptracted later)
temp_use=0;          % temp variable for previous button
temp_q=0;            % temp variable for previous select position to avoid duplicates
Cell={Nl,Sl,El,Wl}; % initialize the options cell to compare the position to the question
%% Trigger Start Animation
animation(a,Cell,blink,num)
%% Start the main game loop
while Do_Stop==1
    %% Trigger the transition animation
    traverse_LEDs(a,Cell,sequence,traverse)
    %% Call a random LED to question
    time1=tic;                           % initialize the timer for the time it takes to get the right answer
    [query,qlocation]=randquery(Cell);  % call a random LED to question
    while qlocation==temp_q             % check if the question is the same as the last...
    temp_q=qlocation;                    % if the question passes the loop then set the quesion to the temp for nest time
    %% Start the check answer loop
    Correct=0;                           % initialize the correct to zero to start the quize
    while Correct==0 && Do_Stop==1...
    %% Check to stop
    Do_Stop=readDigitalPin(a,Stop);         % Check to kill program
    % pause for Effect
    pause(3)
    % turn off all LEDS
    for i=1:4...
    writeDigitalPin(a,Xl,0);
    writeDigitalPin(a,Cl,0);
    % reset the correct variable to restart the check questio loop
    Correct=0;
end
%% Trigger End Animation
animation(a,Cell,blink,num)
%% Generate the Average time graph
Y=Times;
Y1=zeros(1,4);
X = categorical({'North','South','East','West'});
X = reordercats(X,{'North','South','East','West'});
for i=1:4
    Y1(i)=mean(nonzeros(Y(i,:)));
end
maxY=max(max(Y));
figure(1)
bar(X,Y1)
title('Graph of Gamer Average Time to get Correct Answer')
xlabel('Direction on the Compass')
ylabel('Average time to get the correct answer')
ylim ([0 (maxY+1)])
%% Generate the success rate graph
SuccessRate=zeros(1,4);
for i=1:4
    Tries(i)=Tries(i)-1;
    SuccessRate(i)=100-((Errors(i)/Tries(i))*100);
end
Y2=SuccessRate;
figure(2)
bar(X,Y2)
title('Graph of Gamer Success Rates')
xlabel('Direction on the Compass')
ylabel('Success rate % Success to tries')
ylim ([0 100])
```

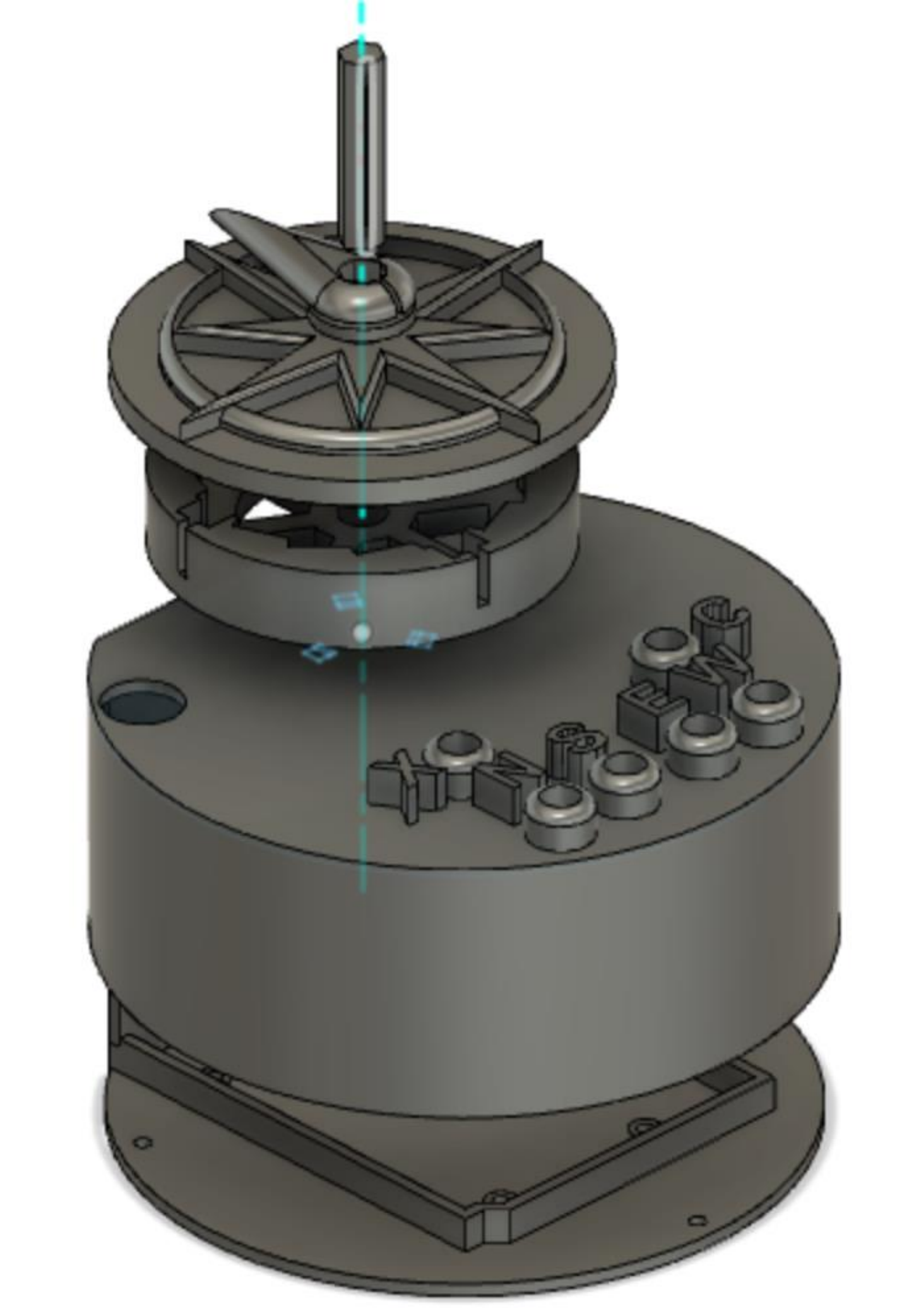
## Methods and Materials

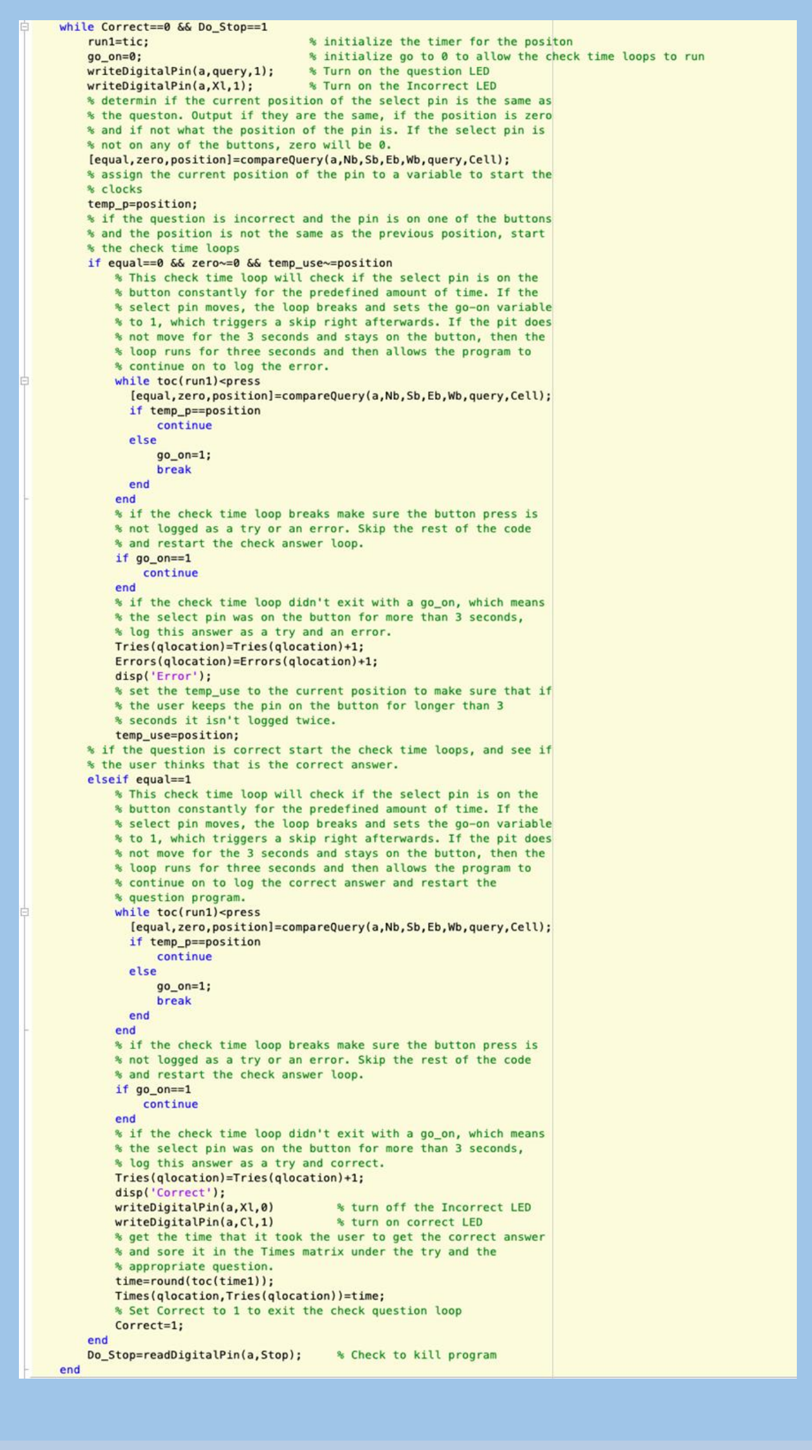
The game's housing, the gearbox for the dial, the dial, the spring, and compass face were all designed by me from scratch in Autodesk Fusion 360. I printed the all the non-electronic parts in the IDEA Lab and wrote all the code in MATLAB. Special thanks to Professor Tolga who made this project possible and helped develop it, Cedric Bleimling the Manager of the IDEA Lab, and Trevor Neal who helped me print the parts. The non electronic parts were all finished in ENGR-125 this past winter. You can see the PCB wiring on the right as well as the implementations.

## Code

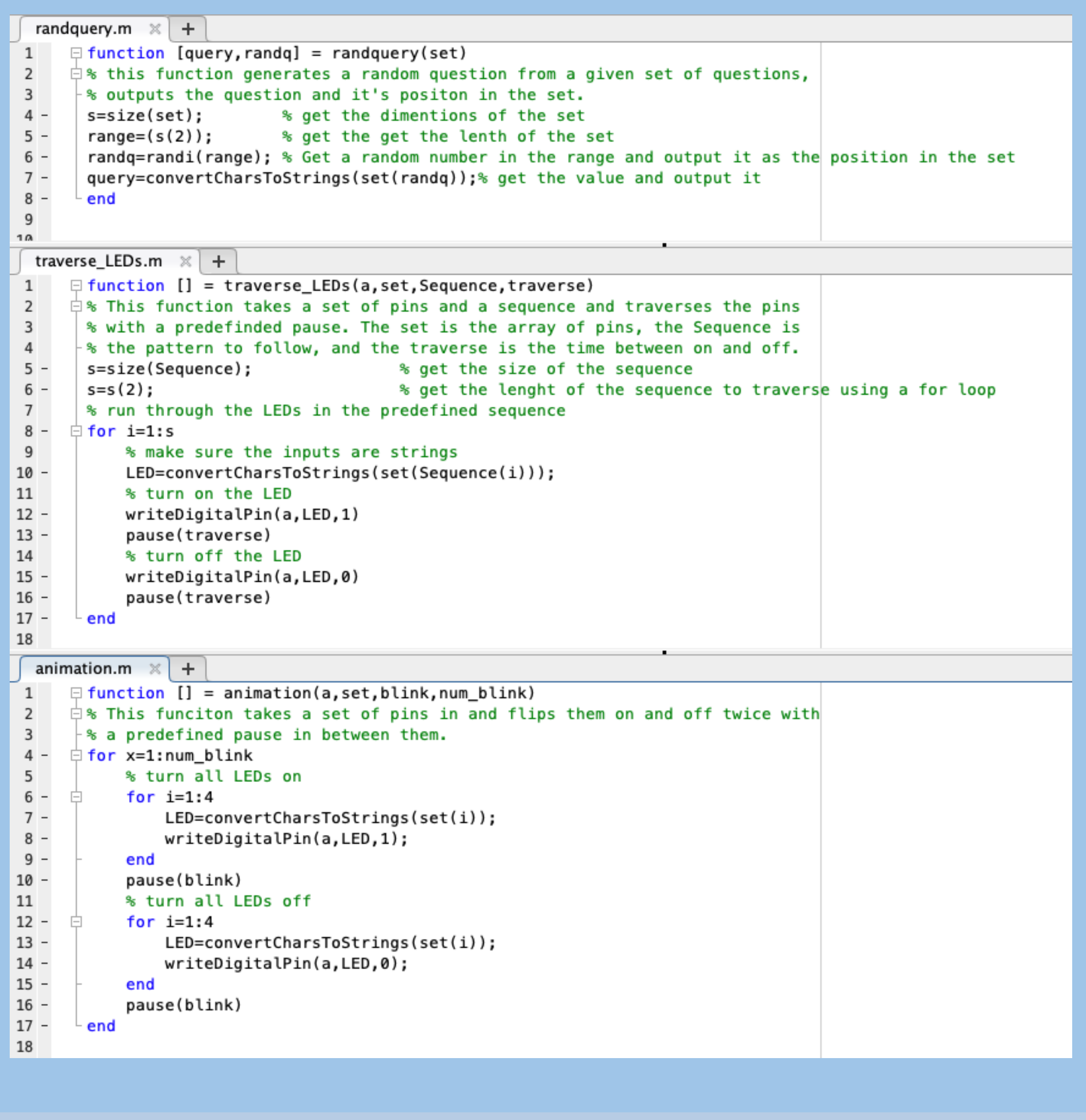
The code of this project was the hardest part. At the start of the game, after a brief animation, the gamer receives a random direction from the LEDs as a question. They can then try to find the correct direction using the bronze dial on the compass top.The game is coded to prevent the same question from being asked twice in a row, and it keeps asking the question until the gamer gets it right. If the gamer keeps the dial on the wrong answer for more than 3 seconds, the answer is logged as incorrect. And if the gamer stays on the correct answer for more than 3 seconds, the answer is logged as correct and a green light turns on. Then it selects a new random question and goes on. While the gamer is playing, the Arduino times how long it takes to get the correct answer. When the gamer is finished they can press the white button to stop the game. In MATLAB the program finishes by displaying the gamer's stats in the form of average time to get each answer, and percent correct.

The main code with comments is displayed to the left. I wrote three functions to do the job which are on the right. There is also a section of the main code on the right (in yellow) that is minimized on the left.





```matlab
while Correct==0 && Do_Stop==1
    run1=tic;                       % initialize the timer for the positon
    go_on=0;                        % initialize go to 0 to allow the check time loops to run
    writeDigitalPin(a,query,1);     % Turn on the question LED
    writeDigitalPin(a,Xl,1);        % Turn on the Incorrect LED
    % determin if the current position of the select pin is the same as
    % the queston. Output if they are the same, if the position is zero
    % and if not what the position of the pin is. If the select pin is
    % not on any of the buttons, zero will be 0.
    [equal,zero,position]=compareQuery(a,Nb,Sb,Eb,Wb,query,Cell);
    % assign the current position of the pin to a variable to start the
    % clocks
    temp_p=position;
    % if the question is incorrect and the pin is on one of the buttons
    % and the position is not the same as the previous position, start
    % the check time loops
    if equal==0 && zero~=0 && temp_use~=position
        % This check time loop will check if the select pin is on the
        % button constantly for the predefined amount of time. If the
        % select pin moves, the loop breaks and sets the go-on variable
        % to 1, which triggers a skip right afterwards. If the pit does
        % not move for the 3 seconds and stays on the button, then the
        % loop runs for three seconds and then allows the program to
        % continue on to log the error.
        while toc(run1)<press
          [equal,zero,position]=compareQuery(a,Nb,Sb,Eb,Wb,query,Cell);
          if temp_p==position
              continue
          else
              go_on=1;
              break
          end
        end
        % if the check time loop breaks make sure the button press is
        % not logged as a try or an error. Skip the rest of the code
        % and restart the check answer loop.
        if go_on==1
            continue
        end
        % if the check time loop didn't exit with a go_on, which means
        % the select pin was on the button for more than 3 seconds,
        % log this answer as a try and an error.
        Tries(qlocation)=Tries(qlocation)+1;
        Errors(qlocation)=Errors(qlocation)+1;
        disp('Error');
        % set the temp_use to the current position to make sure that if
        % the user keeps the pin on the button for longer than 3
        % seconds it isn't logged twice.
        temp_use=position;
    % if the question is correct start the check time loops, and see if
    % the user thinks that is the correct answer.
    elseif equal==1
        % This check time loop will check if the select pin is on the
        % button constantly for the predefined amount of time. If the
        % select pin moves, the loop breaks and sets the go-on variable
        % to 1, which triggers a skip right afterwards. If the pit does
        % not move for the 3 seconds and stays on the button, then the
        % loop runs for three seconds and then allows the program to
        % continue on to log the correct answer and restart the
        % question program.
        while toc(run1)<press
          [equal,zero,position]=compareQuery(a,Nb,Sb,Eb,Wb,query,Cell);
          if temp_p==position
              continue
          else
              go_on=1;
              break
          end
        end
        % if the check time loop breaks make sure the button press is
        % not logged as a try or an error. Skip the rest of the code
        % and restart the check answer loop.
        if go_on==1
            continue
        end
        % if the check time loop didn't exit with a go_on, which means
        % the select pin was on the button for more than 3 seconds,
        % log this answer as a try and correct.
        Tries(qlocation)=Tries(qlocation)+1;
        disp('Correct');
        writeDigitalPin(a,Xl,0)         % turn off the Incorrect LED
        writeDigitalPin(a,Cl,1)         % turn on correct LED
        % get the time that it took the user to get the correct answer
        % and sore it in the Times matrix under the try and the
        % appropriate question.
        time=round(toc(time1));
        Times(qlocation,Tries(qlocation))=time;
        % Set Correct to 1 to exit the check question loop
        Correct=1;
    end
    Do_Stop=readDigitalPin(a,Stop);     % Check to kill program
end
```



```matlab
function [query,randq] = randquery(set)
% this function generates a random question from a given set of questions,
% outputs the question and it's positon in the set.
s=size(set);          % get the dimentions of the set
range=(s(2));         % get the get the lenth of the set
randq=randi(range); % Get a random number in the range and output it as the position in the set
query=convertCharsToStrings(set(randq));% get the value and output it
end
```

```matlab
function [] = traverse_LEDs(a,set,Sequence,traverse)
% This function takes a set of pins and a sequence and traverses the pins
% with a predefinded pause. The set is the array of pins, the Sequence is
% the pattern to follow, and the traverse is the time between on and off.
s=size(Sequence);                 % get the size of the sequence
s=s(2);                           % get the lenght of the sequence to traverse using a for loop
% run through the LEDs in the predefined sequence
for i=1:s
    % make sure the inputs are strings
    LED=convertCharsToStrings(set(Sequence(i)));
    % turn on the LED
    writeDigitalPin(a,LED,1)
    pause(traverse)
    % turn off the LED
    writeDigitalPin(a,LED,0)
    pause(traverse)
end
```

```matlab
function [] = animation(a,set,blink,num_blink)
% This funciton takes a set of pins in and flips them on and off twice with
% a predefined pause in between them.
for x=1:num_blink
    % turn all LEDs on
    for i=1:4
        LED=convertCharsToStrings(set(i));
        writeDigitalPin(a,LED,1);
    end
    pause(blink)
    % turn all LEDs off
    for i=1:4
        LED=convertCharsToStrings(set(i));
        writeDigitalPin(a,LED,0);
    end
    pause(blink)
end
```

## Conclusions

This was a really fun project. It is pretty and functional. I learned so much about designing and prototyping. One design defect is that the dial only goes in one direction. This affects the game's times to get each answer. That is something that could be improved in future versions. All in all it was a lot of work but rewarding. Thank You!

## Contact

Michael OHara
oharam5@mail.sacredheart.edu