Mid-year Design Review

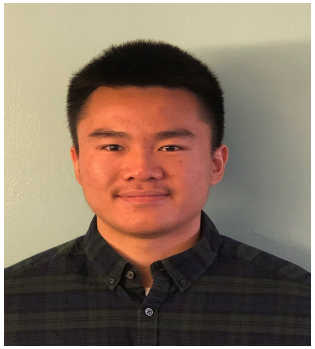
Tetra Board

SDP 2020

Team 7

December 4, 2019

The Team



Aleck Chen,
EE

James Gnall,
EE

Varak Mouradian,
EE

Vincent Nguyen,
CSE

Professor Polizzi

Problem

- Battleship is a classic game that has evolved from pencil and paper game into a fully fledged board game
- The current iteration requires physical pieces of ships and pegs to be able to play
- Requires precise movement that otherwise may be impossible for those with poor motor skills

Our Solution: Voice Controlled Game Board

- Eliminate as much of the physical involvement required to play as possible
- Utilize an I/O system that implements microphones, a speaker, and lights to command the actions of the game
- Replace physical pieces with LEDs that will light up depending on the status of that tile
- Have a “Quick Play” option that implements a random board layout to be quickly and easily set up without the opponent hearing where you put your boats

System Specifications

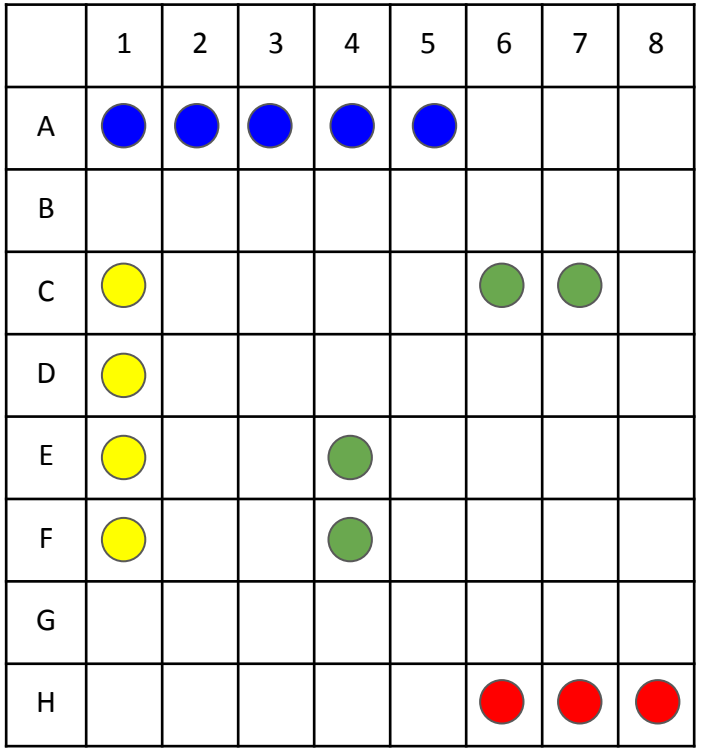
- 4 boards (2 for each player)
  - 1 board is for your own moves
  - 1 board is for opponent's moves
- Each board contains 8x8 alphanumeric grid with 64 LEDs (256 total)
- 1 microphone
- 1 speaker to play movement result
- Controlled using a Raspberry Pi
- Housed in a double sided briefcase

Battleship Gameplay Specifications

- Raspberry Pi will keep track of the status of both player's boards
- LEDs will provide result of each moves depending on whether it was a hit or a miss
  - Results of moves are displayed for both players on their respective boards
- Speakers notify players of game events
- 2 Player game option
- CPU with 3 levels of difficulty
- Ability to exit or quit out of game
- Ability to manually or automatically set ship locations

Board Layout

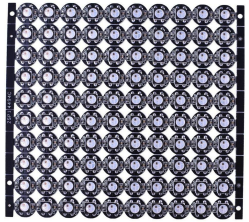
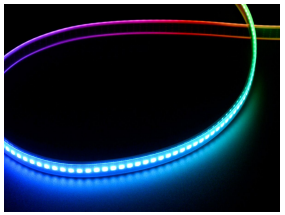
#	Class of Ship	Size
1x	Carrier	5
1x	Battleship	4
1x	Destroyer	3
2x	Submarine	2



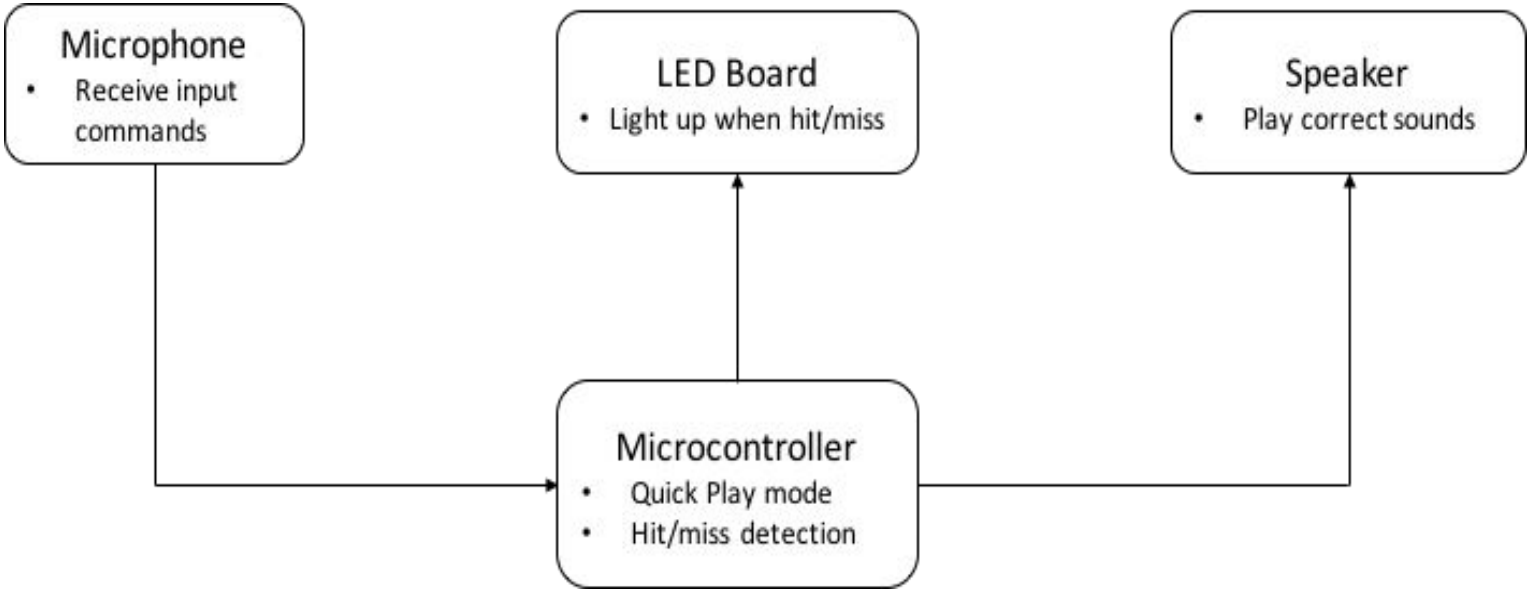
Microphone Input/Speaker Output

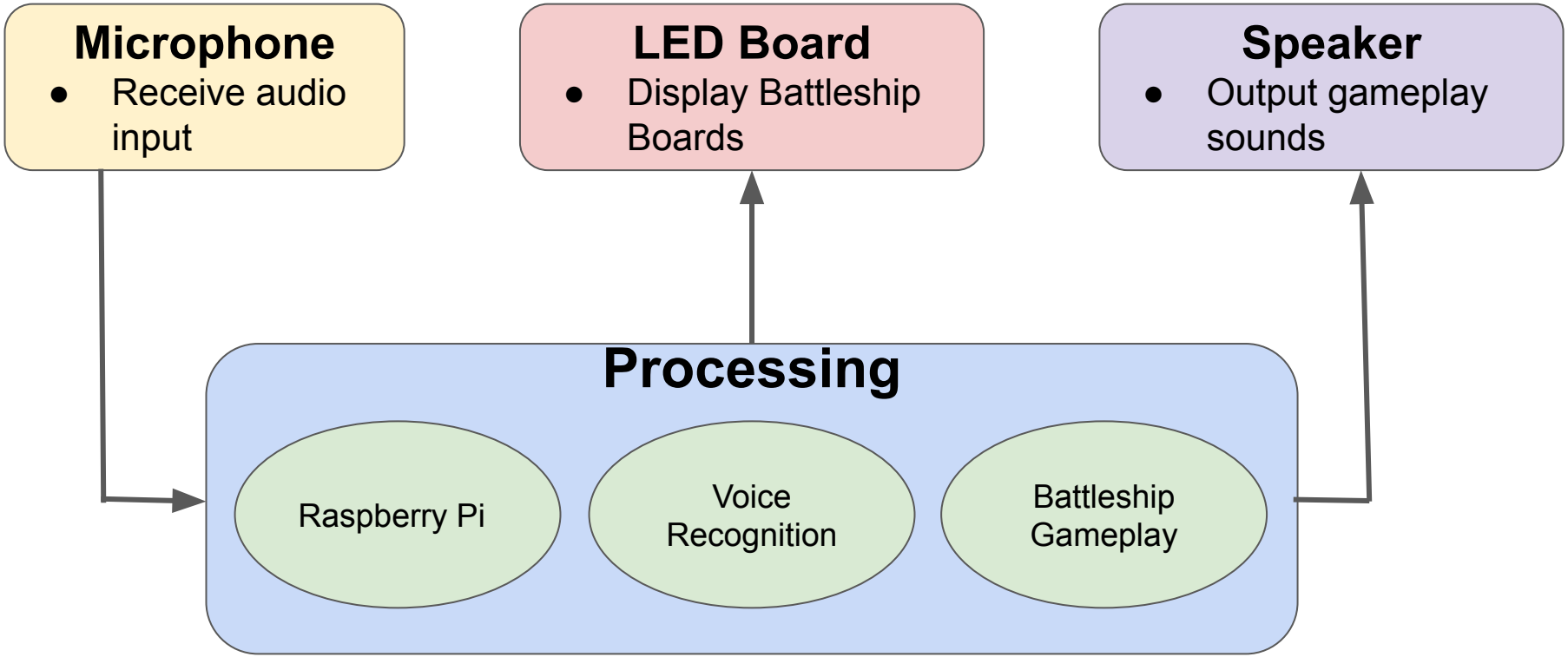
- Speaker
  - “Hit” - announces a hit
  - “Miss” - announces a miss
  - “Sunk X” - announces a the particular ship that has been sunk
  - “Player X Wins” - announce winner of the game
- Microphone
  - “Alpha 3” - flash square at A3
    - “Confirm” - to confirm move
    - “Cancel” - to cancel a move
  - “Surrender” - announces player has given up and ends the game
  - “Reset” - game will reset
    - “Quick Play” - randomizes ships on each board
    - “Regular Play” - initializes ships to be placed based on user input

LED Comparison

	Diodes	Neopixels	Dotstar
			
Addressable	No	Yes	Yes
Multiplexing Included	No	Yes	Yes
Compatible with Raspberry Pi	Yes	No	Yes

Block Diagram




Microphone
Receive audio input
LED Board
Display Battleship Boards
Speaker
Output gameplay sounds
Processing
Raspberry Pi
Voice Recognition
Battleship Gameplay

Microcontroller

Requirements:

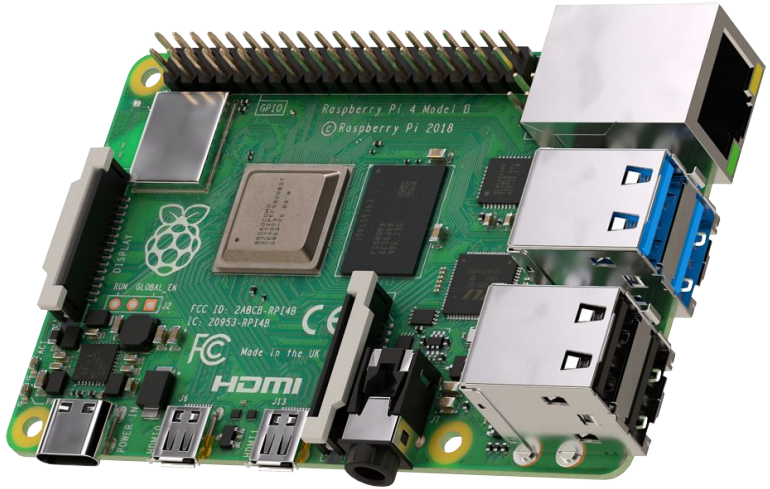
- Handle real-time speech recognition
- Store voice recognition library and game data

Proposed Solution:

- Raspberry Pi 4 - Model B

Alternative:

- Development Board with SoC
  - ARM Cortex Processor



Hardware

- 1 Microcontroller
- 4 8”x8” boards to wire the LED matrix
  - Wiring LEDs to the board (256 total)
- 1 PCB
  - Voltage Regulator
  - Will implement external 5V power supply
- Each LED is addressable by Raspberry Pi

Board Power Consumption

- 128 LEDs from 2 boards (ship board and attack board)
- A strip of 60 LEDs consumes 9W of power with a 5V power supply and 1.8A of current at maximum brightness.
  - Our implementation utilizes these LEDs at 10% brightness
  - Power for each LED is 30mW
  - Total current through each board is 180mA
  - Starting setup will consume 0.48W (16 LEDs for all ships)
  - Max power consumed from both boards is 3.84W
  - Mixed colors will use less than 30mW per LED

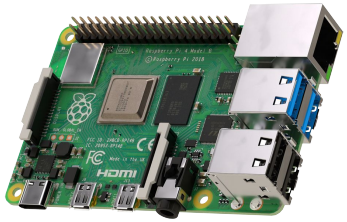
PCB Design

- Construct Voltage Regulator
- External power source will be 5V wall adaptor
- PCB will distribute 5V to all 4 boards and microcontroller

Altium Designer

Costs

Raspberry Pi 4	$48.01
Dotstar LEDs x2	$49.90
HDMI	$5.98
2-sided Briefcase	$27.16
Microphone	$12.99
Adaptor	$7.99
Micro SD Card	$9.40
Testing Components	$23.87
Total	$185.30



Responsibilities

Varak	Aleck	Vincent	James
Design the PCB that will be used for CDR, help plan/connect the microcontroller with the LED array. Took the lead on administrative tasks	Design and implement the LED array on the suitcase. Helped debug gameplay code.	Program/Implement voice control software to recognize moves when spoken into the mic. Help program gameplay and layout of the board	Program the gameplay to receive instructions from the voice recognition software and execute that command on the board. Help with planning of the hardware layout

Gantt Chart


20-Jan
27-Jan
3-Feb
10-Feb
17-Feb
24-Feb
2-Mar
9-Mar
16-Mar
23-Mar
30-Mar
6-Apr
Speaker Refinement (Vincent)
Ability To Manually Set Ships (Vincent)
Offline Speech Recognition (Vincent + James)
CPU Difficulty Levels (James)
2nd Player Option (James)
PCB Design (Varak)
Suitcase Implementation (Varak + Aleck)
3rd and 4th Boards (Aleck)
Connect 4 (Aleck)
Tic Tac Toe (Aleck)
CDR Preparation (All)
CDR Presentation (All)
Project Refinement (All)

MDR Deliverables

- The ability to play one player against the CPU
- Speakers are able to project sounds based on the current game being played
- Board correctly lights up whether it is a hit or a miss
- Quick play mode

MDR Deliverables

Goal	What is left to do
The ability to play against CPU	
Quick Play mode	
Board lights up correctly on hit or miss	
Speaker projects sounds without voice commands	Have speaker and voice command working properly at the same time

Proposed CDR Deliverables

- Implement Player vs. Player mode
- Implement different levels of CPU difficulty
- Ability to manually set location of ships
- Addition of new games
  - Connect Four
  - Tic Tac Toe

Demo

https://www.youtube.com/watch?v=DewwipwgW8k

Thank you

Questions?