Preliminary Design Review

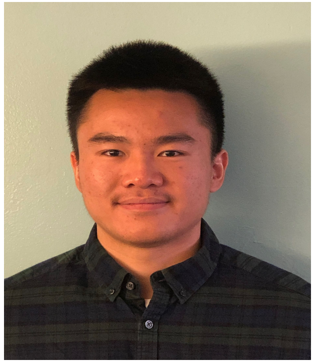
Tetra Board

SDP 2020

Team 7

October 9, 2019

Meet the Team



Aleck Chen, EE

James Gnall, EE

Varak Mouradian, EE

Vincent Nguyen, CSE

Professor Polizzi

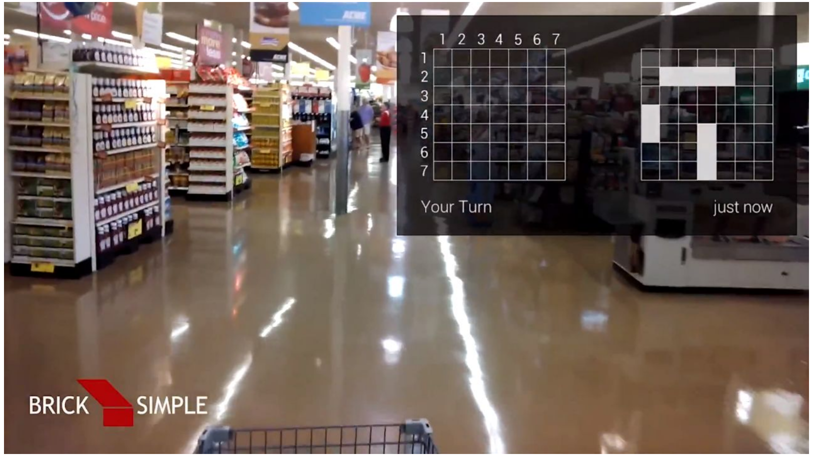
Problem

- 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

Current Solutions #1

GlassBattle:

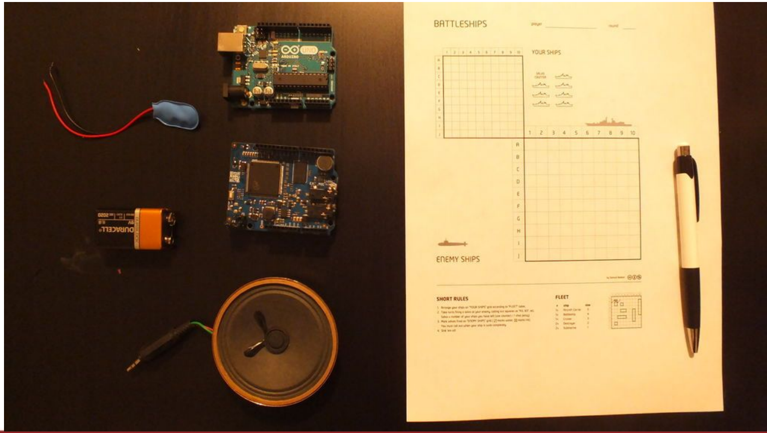
- App for Google Glass that allows you to play voice controlled Battleship while completing daily tasks
- Only Player vs Player mode with randomly generated ship placement on 7x7 grid
- Command library limited to (x,y) configuration instead of traditional alphanumeric grid



Current Solutions #2

Voice-based Battleship Game for Arduino:

- Arduino programmed for battleship with speaker and mic that allows for audio input of commands
- Does not have a board, requires one to keep track of moves on a piece of paper
- Single Player, can only play against a CPU



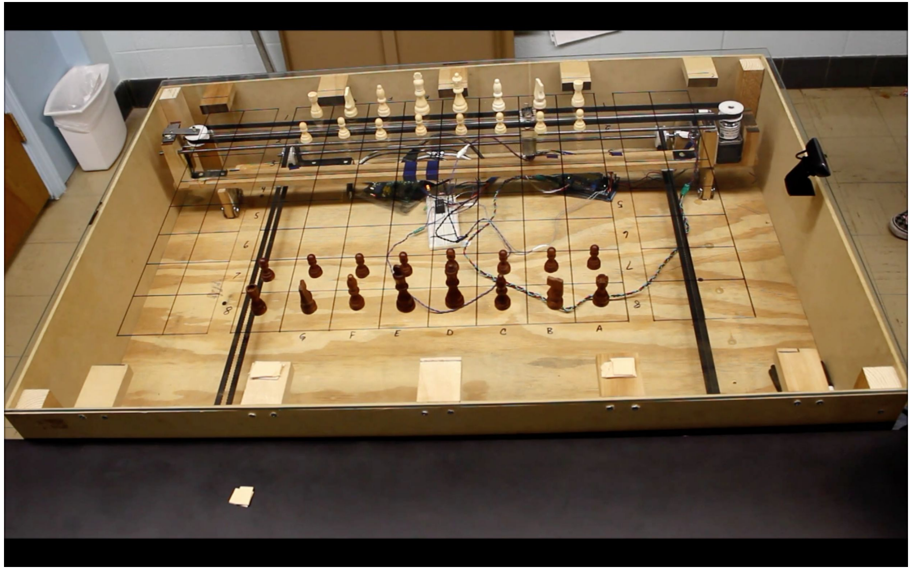
Our Solution: Voice Controlled Battleship

- 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 green/red/blue 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

Design Alternative #1

Voice-controlled Chessboard:

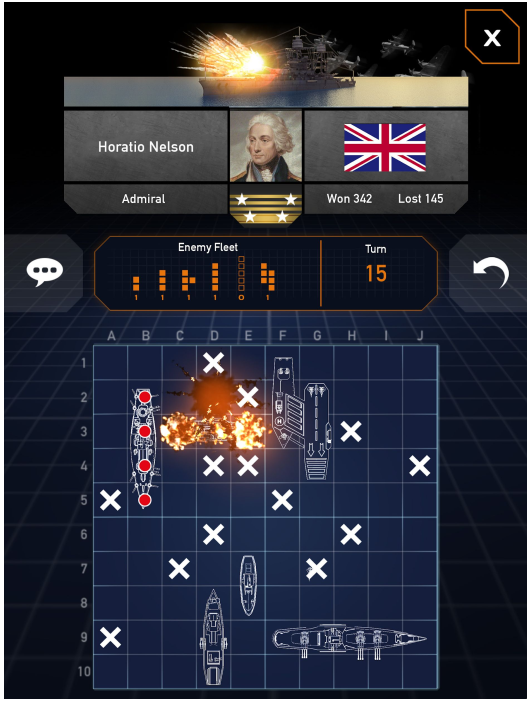
- 8x8 voice controlled board
- Pieces are transported by magnetic mechanism located under the board
- Utilizes microphones in order to take in the user voice
- The mechanism only allows movement of pieces during the game and is not able to reset pieces for a new game



Design Alternative #2

Fleet Battle:

- Game on Androids which supports both one and two player gameplay
- Game relies on touchscreen input from players which may not be welcoming for people with poor motor skills
- LEDs are a cheaper substitute for physical pieces



Board Layout

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

	1	2	3	4	5	6	7	8
A								
B								
C								
D								
E								
F								
G								
H								

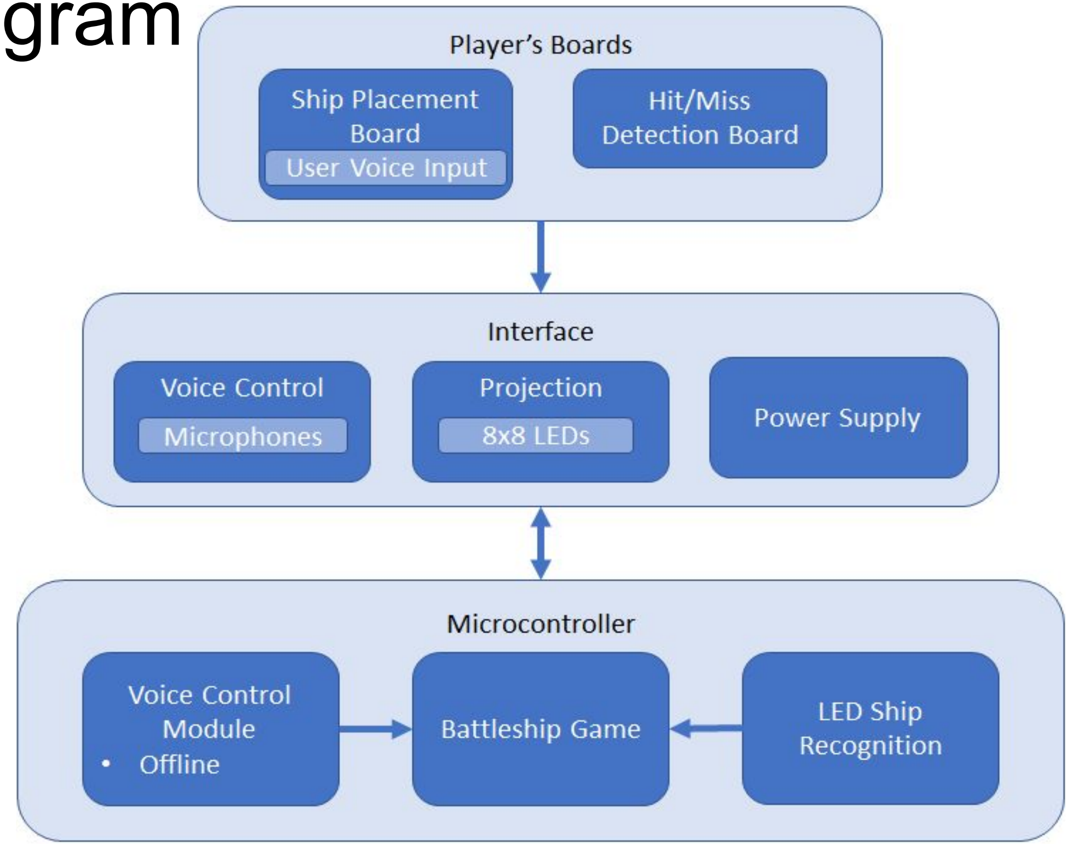
System Specifications

- 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)
- 2 microphones (1 for each player)
- 1 speaker to play movement result
- Players take turns saying commands into their microphones to initiate moves
- 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

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

Block Diagram



Microcontroller

Requirements:

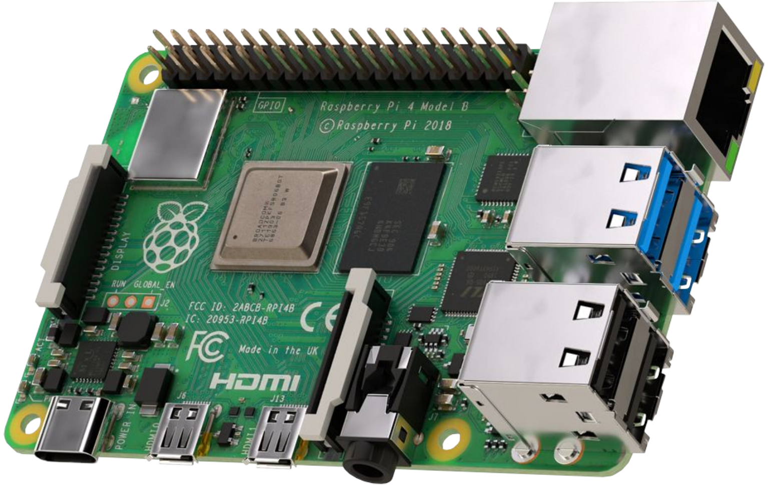
- Handle real-time speech recognition
- Store voice recognition library and game data
- Require 3.2V for LEDs

Proposed Solution:

- Raspberry Pi 4 - Model B

Alternative:

- Development Board with SoC
  - ARM Cortex Processor



Hardware

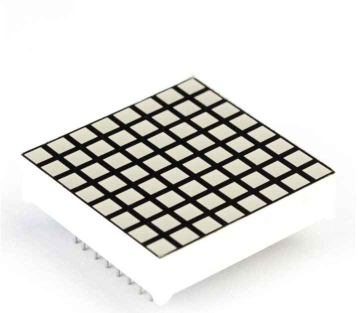
- 1 Microcontroller
- 4 8”x8” Proto boards to wire the LED matrix
  - Wiring LEDs to the board (256 total)
  - PCB will replace one of these boards for FDR
- 1 PCB
  - Needs to output a minimum of 3.2V to LEDs
  - Will implement 5V power supply
- Each board of LEDs is addressable by Maxim7219

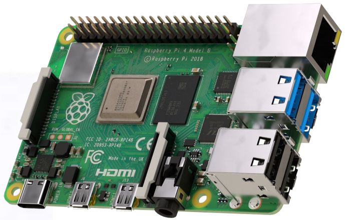
Software

- Analyze and process user input using voice recognition
- Battleship Play
    - Quick Play and Regular Play
    - Two modes:
        - Two Players
        - Single Player
            - CPU with randomly generated moves based on difficulty settings (easy, medium and hard)

Costs

- LEDs = ~$30
    - 64 LEDs for 4 boards = 256 LEDs
    - 256/100 = 3 lots for ~$10 each
- PCB = ~$20
- 4 Proto Boards = ~$140
- Plastic enclosure = ~$20
- Raspberry Pi 4B = ~$50
- Two-sided suitcase = ~$30





Total Cost = $290

Responsibilities

Varak	Aleck	Vincent	James
Design the PCB that will be used for FDR, help plan/connect the microcontroller with the LED array	Design and implement the LED array on the suitcase	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

Potential Challenges

- Voice Control Capability
- Aesthetic
- Portability

MDR Goals

- 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

Post-MDR Applications

- Complete two player mode
- Plans to implement more games to the board
  - Connect Four
  - Tic Tac Toe
  - Tetris
  - Our own game
- Normal play mode

Thank you

Questions?