

Jack of All Trades

Team A7

Miya, Mason, Rachel

Use Case/ Problem Statement


With our system, it is possible to play card games over the internet with physical cards.

ECE Areas:


Software and Hardware



Game Requirements

- Play with physical cards
 - Plays the games Go Fish, Euchre, and Rummy
 - Multiplayer support up to 5 players per game
 - Be able to input any card for game logic
 - Ability to have concurrent games
- 
- A decorative graphic at the bottom of the slide consisting of a series of blue dots arranged in a wavy pattern that spans the width of the page.

Quantitative Requirements

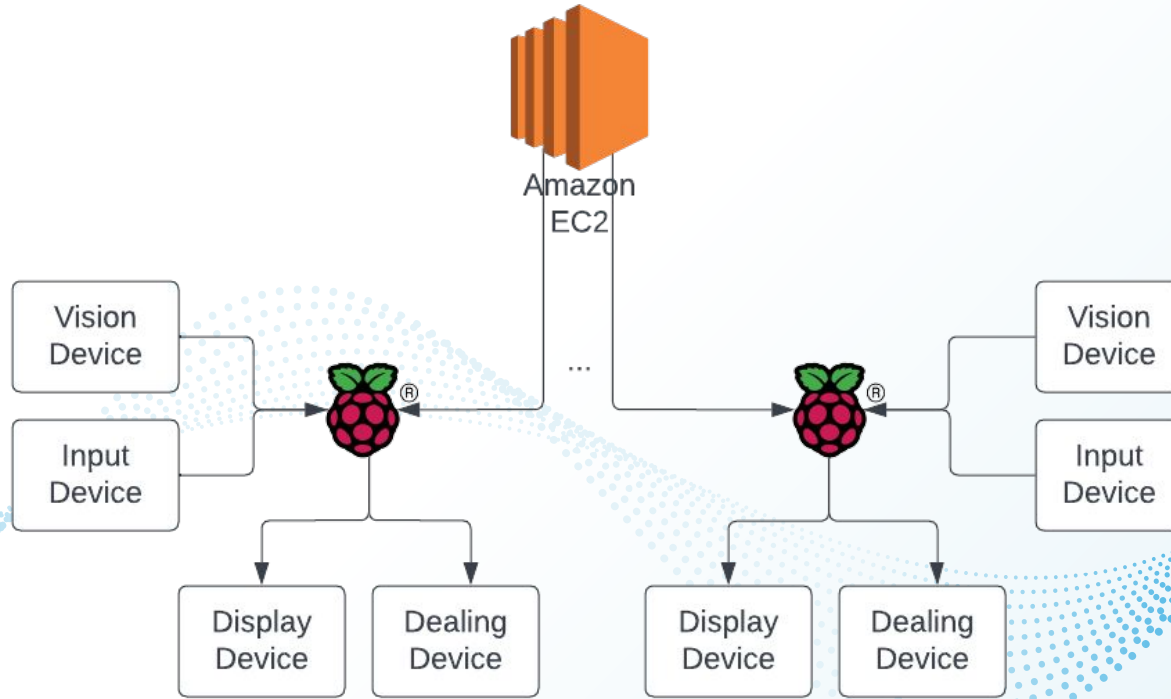
- A 18" x 24" playing/vision area
 - Playing/vision area updates are done at least once per second
 - When dealing cards are emitted at least once per second
 - The full physical device is smaller than a shoebox (14 in x 10 in x 5 in) and lighter than 10lbs
- 
- A decorative graphic consisting of a series of blue dots arranged in a wavy pattern, starting from the left side of the slide and extending towards the right, below the list of requirements.

Technical Challenges

- Real-time signal handling
- Classification latency
- Identifying unique devices to facilitate multiple players in distinct games
- Interfacing between the multiple physical devices (input, display, and dealing)



Approach/Architecture



Vision Device and Classification

Hardware:

- Raspberry Pi Camera Module 3 Wide

Software:

- OpenCV using Python to detect the card played by the user
- Tensorflow to train an ML model to identify the card's suit and number

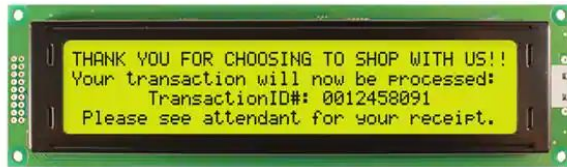
Input Device

- 10 key keyboard for user input of cards
- Repaint some keys to represent suits



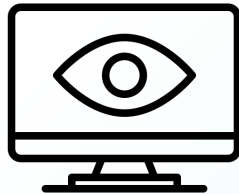
Output Devices

- 40×4 character LCD Screen to display commands and prompts involved in games
- A mechatronic dealing device is out of scope, but to keep the physical card aspect, we will use a receipt printer



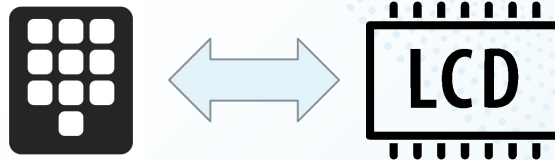
Unit Testing

1. *Thermal printer*: Be able to print the correct cards with corresponding suit and number with a consistent size
2. *Camera/Computer Vision*: Properly identifies card(s)
3. *Small keyboard*: Inputs are properly received and buffered.
4. *LCD Screen*: Displays text, then special characters like suits
5. *Implementing game logic for different games*: Go fish, Euchre, Rummy
6. *EC2/Networking*: Concurrency and logic tests.



Integration Testing

1. *Software device-level supervisor*: Services interrupt from peripherals in a timely manner without dropping any signals.
2. *Keyboard/Screen Coupling*: Keypresses appear on screen within our latency targets.
3. *Server/device Network Protocol*: The device supervisor is able to send game state update messages to the server in a timely manner, and the server can reconstruct a matching local game state. The reverse is also true, the server can send commands to the device, which are serviced in a timely manner. The string representation of the game state is shown on the device screen.



Division of Labor

ML track: Rachel

Hardware track: Mason & Miya

Software track: Mason & Miya (& Rachel)