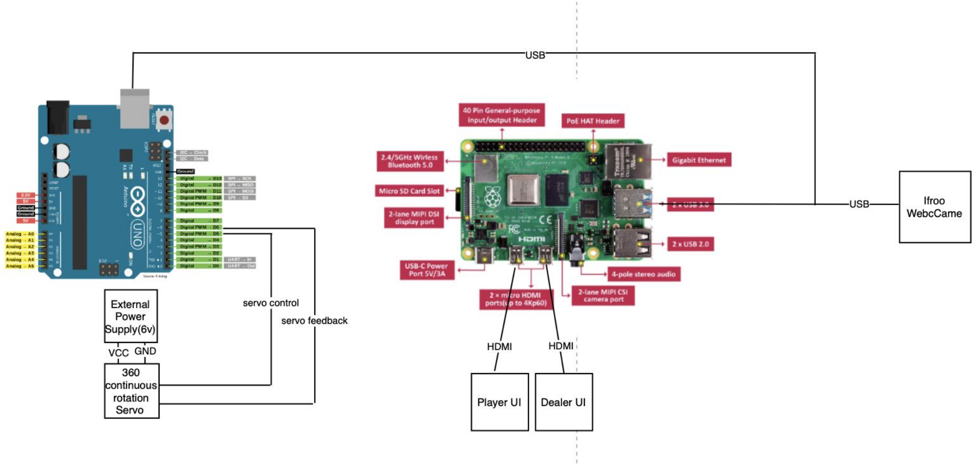
Team E8: Smart Poker Table Final Presentation

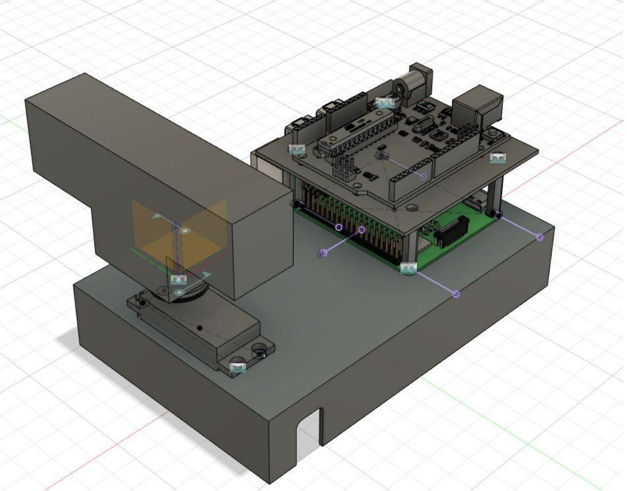
Zongpeng (Steve) Yu, Brandon Hung, Patrick Kollman

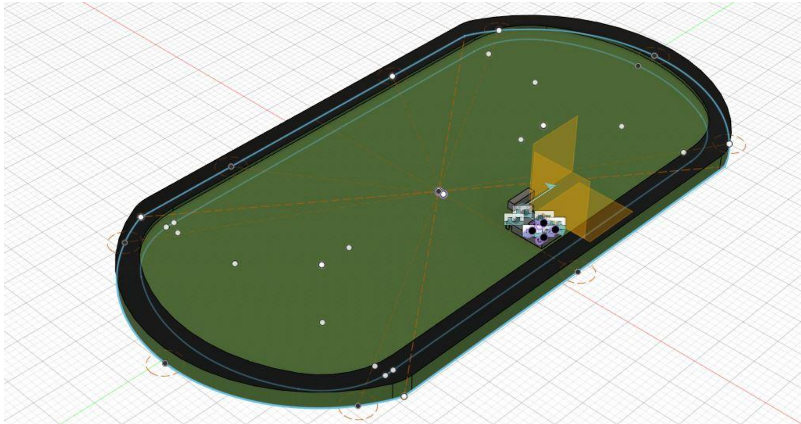
Application Area

- Display poker statistics (pot size and player stack sizes), that are available in online poker games, to real life poker players in a casino
- CV provides relatively robust and fast stack height/color reading
- Intuitive software for dealer to keep track of the game state

Solution Approach





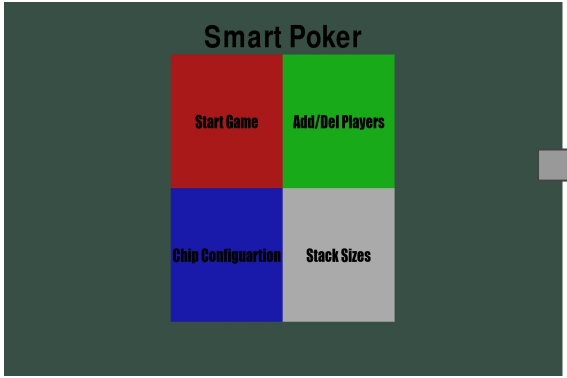


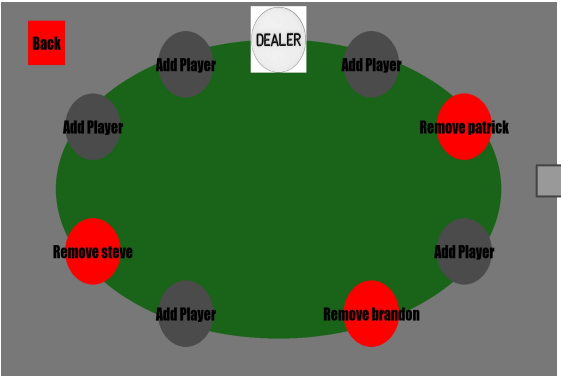
Complete Solution

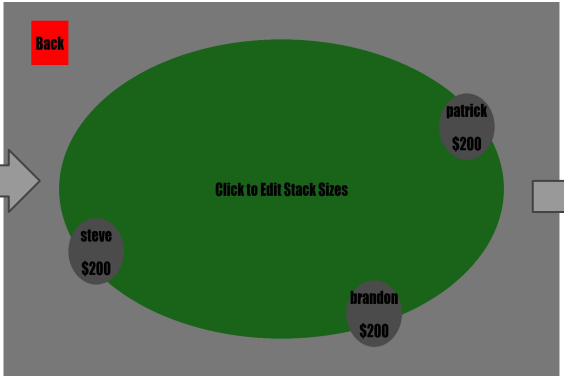
In this setup

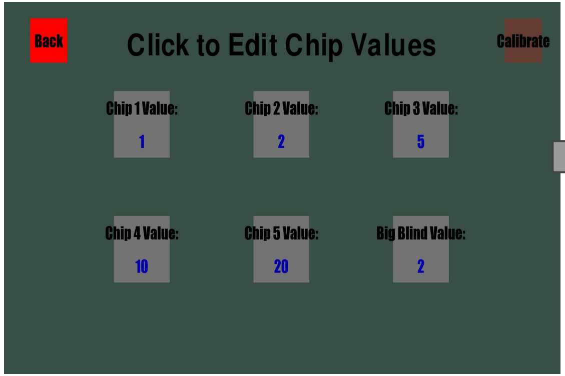
- Brandon’s stack was $84
- CV algorithm detect $86
- Accuracy of attempt: 97.7%

Complete Solution









Chip 1 Value: 1

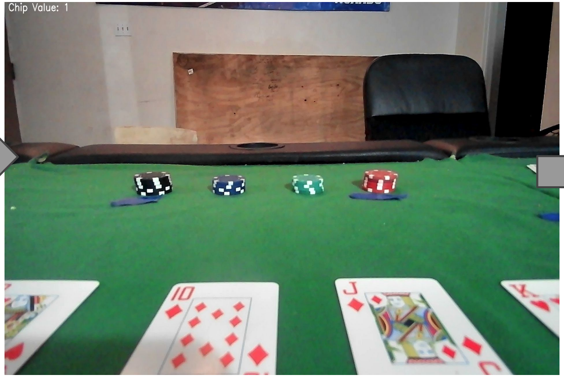
Chip 2 Value: 2

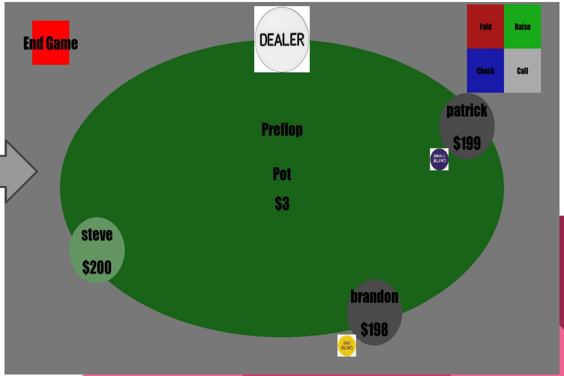
Chip 3 Value: 5

Chip 4 Value: 10

Chip 5 Value: 20

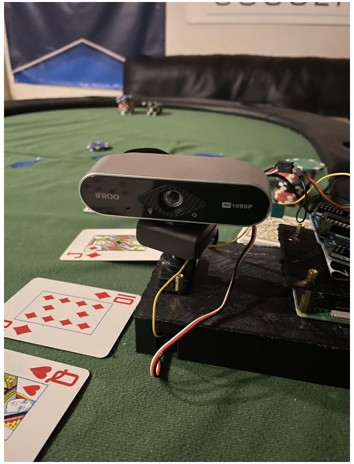
Big Blind Value: 2





Testing, Metrics and Validation: Servo

Original Metrics	Test Inputs	Method	Result
Camera Angle offset +/-4 degrees	Player position sent by dealer SW through serial	Compare the result with the protractor; 20 trials	Average error was 1 degree from target
Camera rotate to spot with 5 secs	Player position sent by dealer SW through serial	Use a Timer (20 trials)	Average time within 1 seconds



Testing, Metrics and Validation: Computer Vision

- Original metric: +/- 10% of real stack value
- Results:

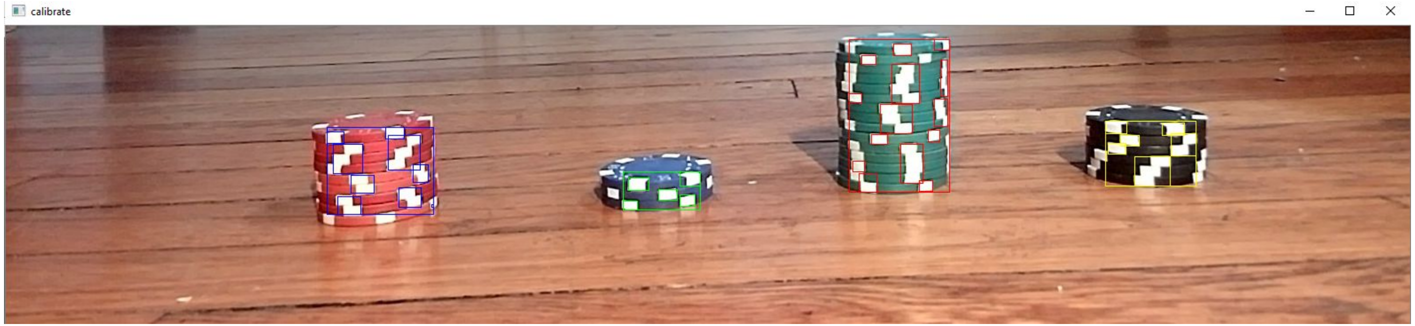
Stack Value	Percentage within +/- 10%	Mean Error in Percentage	Tests Ran
$46.00	80.00%	6.49%	5
$84.00	80.00%	8.68%	5
$150.00	43.18%	18.48%	7
$210.00	31.86%	22.46%	6

Source of error: not standardized sizes, chip widths varied significantly and aspect ratios were ruined when resizing pictures

CV Tradeoffs

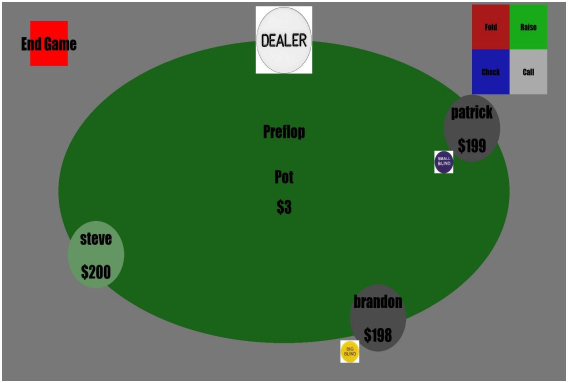
- Original
  - Algorithm: Color blob detection
    - Pros:
      - Simple to implement
      - Can scan multi-colored stacks
    - Cons:
      - Strongly influenced by lighting
      - Less accurate
      - Inconsistent due to image noise despite preprocessing
      - Requires chip height in pixels be known

- Final
  - Algorithm: Checker grouper and detector
    - Pros:
      - More accurate/robust
      - Resistant to noise
      - Automated estimation of chip sizes
    - Cons:
      - Complex to implement
      - Accuracy depends on calibration
      - Calibration routine more complicated
      - Requires chips have white checkers
      - Cannot scan multi-colored stacks



Testing, Metrics and Validation: Software

- Original Metrics:
    - **100%** accurate simulation required
    - Dealer UI learnable in less than **5** minutes
- Tests:
    - Robust gameplay and intentionally trying to break the software
    - Introduced Dealer UI to 5 different roommates
- Results:
    - **100%** accurate simulation, no bugs found
    - Average Time for the 5 roommates to learn and master the Dealer UI was **2 minutes 55 seconds**
- Design Tradeoffs:
    - Player UI and Dealer UI combined into one subsystem



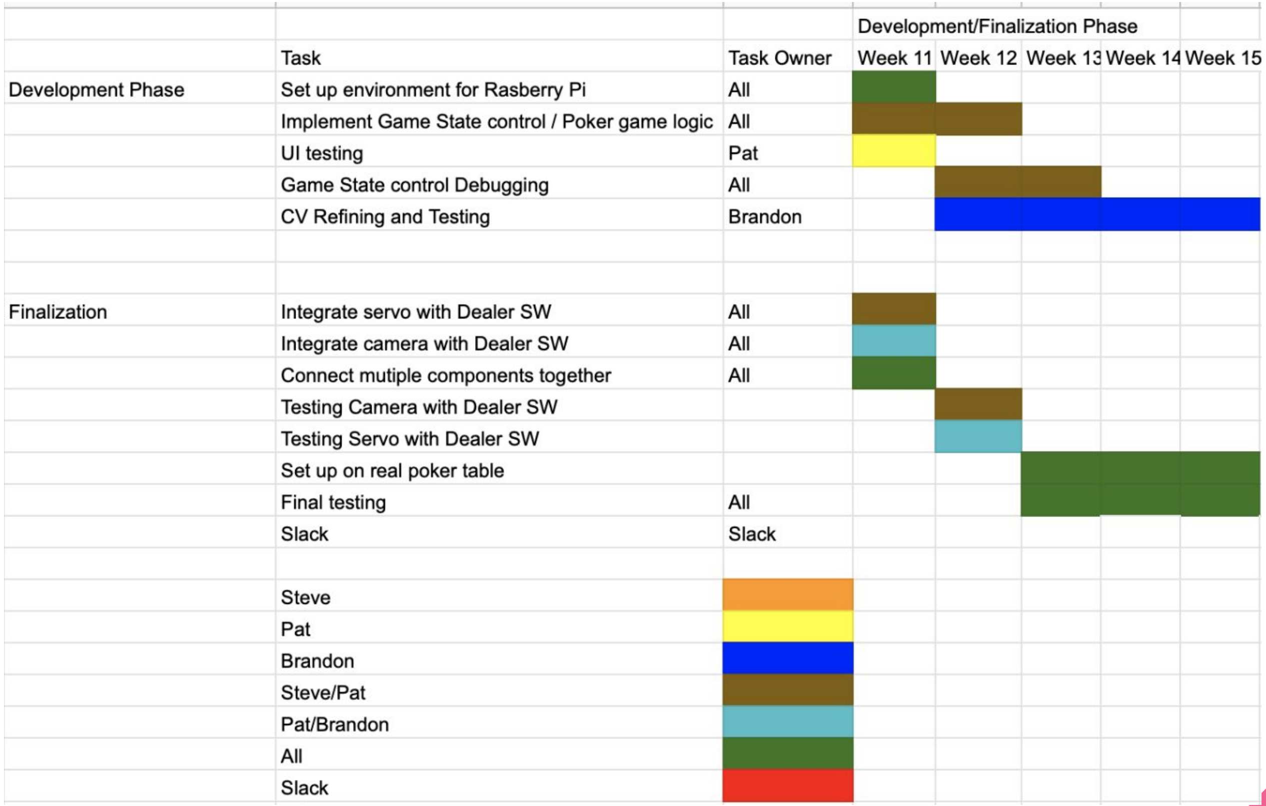
Testing, Metrics and Validation: Overall System

- Tested each function of the system by playing multiple rounds using it:
  - Tested display update using timer
  - Tested that previous Servo, Software, and CV performance hold
- Results:
  - Display/game state update time: < 1 second
  - Previous subsystem performance maintained
  - Few bugs from CV
- Design Tradeoffs made
  - Bets: placed at a designated area on the table
  - Color temperature: held constant and high

Metrics and Validation Summary

Component	Testing Strategy	Metrics	Results
Servo Motor	Input: unique ID assigned to each player Output: servo rotating to the specified position	Camera angle offset within +/- 4 degrees; camera should rotate to the position within 5 seconds	Success: Average error = 1 degrees Success: Average time = 1 sec
Computer Vision	Input: unoccluded picture of bet Output: total value of bet	Verify +/-10% of total stack value	Partial Success: Accuracy goes down with height
Software	Input: Game state inputs entirely simulated in software Output: game state, stack sizes, and pot size update correctly in the simulation	100% accurate simulation required; the simulation should update correctly given the inputs it receives	Success: 100% accurate simulation Success: 2:55 avg learning time
Overall System	Input: dealer software controls Output: servo position and updated game state, stack, and pot size to monitor display	After integration, the previous subsystem metrics hold and the display is updated within 5 seconds	Partial success: Integration was successful and the previous subsystem metrics besides CV hold

Project Management



			Development/Finalization Phase				
	Task	Task Owner	Week 11	Week 12	Week 13	Week 14	Week 15
Development Phase	Set up environment for Rasberry Pi	All	🟩				
	Implement Game State control / Poker game logic	All	🟫	🟫			
	UI testing	Pat	🟨				
	Game State control Debugging	All		🟫	🟫		
	CV Refining and Testing	Brandon		🟦	🟦	🟦	🟦
Finalization	Integrate servo with Dealer SW	All	🟫				
	Integrate camera with Dealer SW	All	▨				
	Connect mutiple components together	All	🟩				
	Testing Camera with Dealer SW			🟫			
	Testing Servo with Dealer SW			▨			
	Set up on real poker table				🟩	🟩	🟩
	Final testing	All			🟩	🟩	🟩
	Slack	Slack					
	Steve	🟧					
	Pat	🟨					
	Brandon	🟦					
	Steve/Pat	🟫					
	Pat/Brandon	▨					
	All	🟩					
	Slack	🟥					