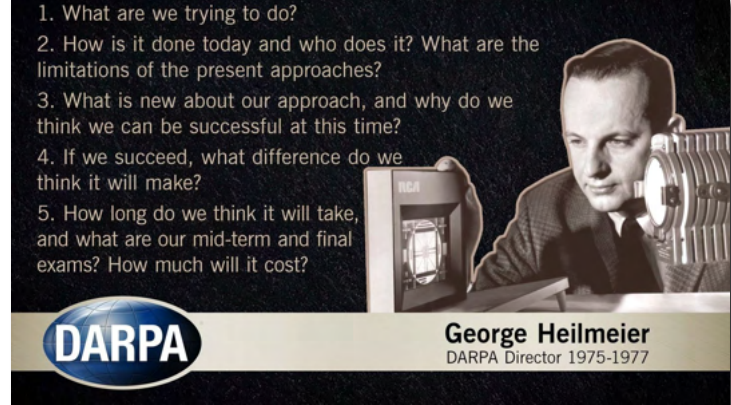


Team E1 – Gambletron 9000

Samuel Istvan, Benjamin Berger, **Andrei Da Silva**



Add your 12 slides after this slide... [remember, 12 min talk + 3 min Q/A]

For more information about formatting or importing slides see:

<https://gsuite.google.com/learning-center/products/slides/get-started/>

Make sure to cover

(refer to the Final Presentation Guidance):

- Use Case & Requirements - Remind us who is your customer, what product do they want to buy and quantitative reqs [keep it short]
- Quantitative Design Reqs - Map quantitative use case to quantitative design requirements [also keep it short]
- Solution Approach Block Diag. - Remind us of solution path(s) ... and social, cultural, global .. factors; any changes since design ?
- System Implementation - e.g., pictures, screenshots, video (make sure that there is CMU access to play any media)
- Testing, Verification and Metrics - What tests did you run ? How many tests ? What were the results ? e.g. Graphs, quantitative tables, ...
- Project Management - WHO is did WHAT and HOW did you address challenges ?
- Lessons Learned - Summarize what you learned

Consider that this slide already works as a introduction slide so use your first slide wisely

Remember – Relevant figures (and tables) can be worth
“a thousand words”!

Use Case

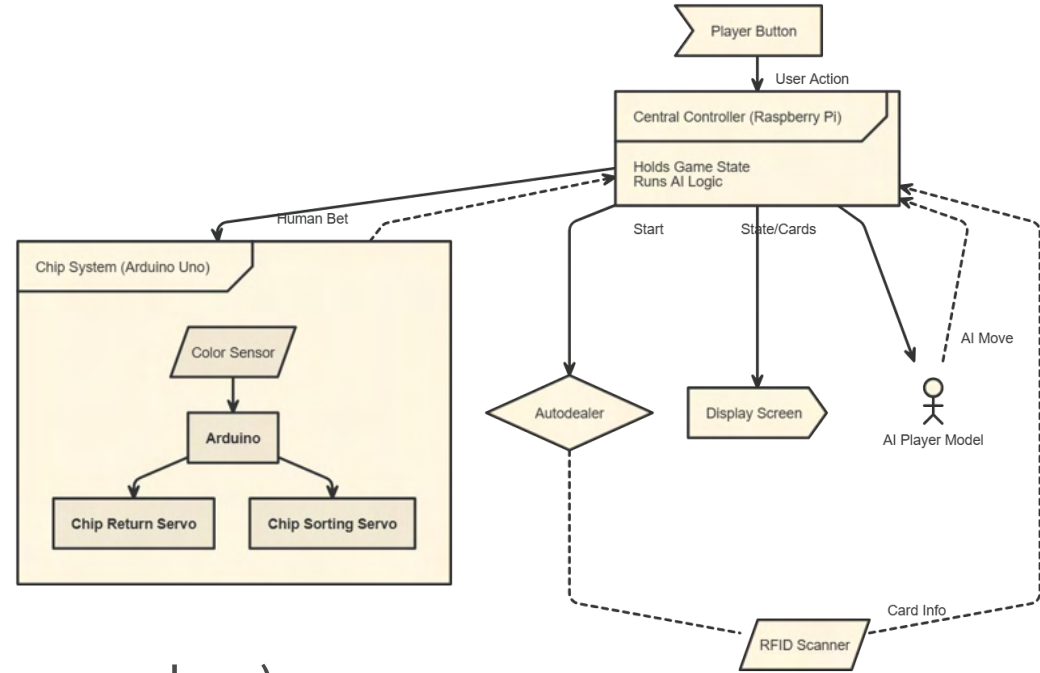
- At-Home Texas Hold'em Poker
- Works with 1-6 human players
- Automated Dealing
- Fit for all levels of play

Quantitative Requirements

Use Case Requirements	Performance Criteria	Design Requirement	Performance Criteria
Enforce Texas Hold'em rules	100% accuracy	Rule enforcement	100% accuracy
Card Identification	100% accuracy	RFID Accuracy	$\geq 99.9\%$
Chip Identification	100% accuracy	Color sensing Accuracy	$\geq 99.9\%$
AI Response Time	<30 seconds	RFID read latency	<100ms
Dealing Speed Time	<15 seconds		

Solution Approach

- Central Controller (RPI)
- RFID based card tracking
- Automated dealer module
- AI poker agents
- Game state and rules engine
- Player interface (screen, buttons, speakers)



Changes to Solution Approach

Chips Return Changes:

- No longer sending chips to each individual.
- We will be sending chips to a central chip return pot.

Card Dealer Changes:

- Card dealer shoots cards out too quickly.
- Card dealer no longer has a single RFID reader attached to it.
- RFID readers are carefully replaced underneath the landing area of the cards.
- It is possible that a player may have to manually tag their cards.

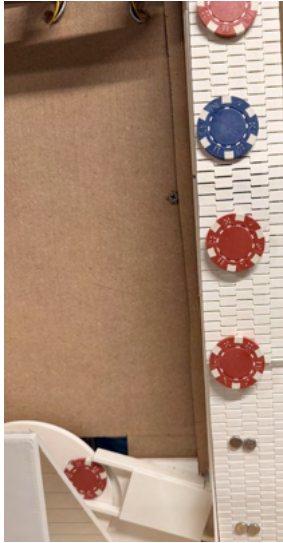
Complete Solution - Chip Collection



1) Slot entering chip system



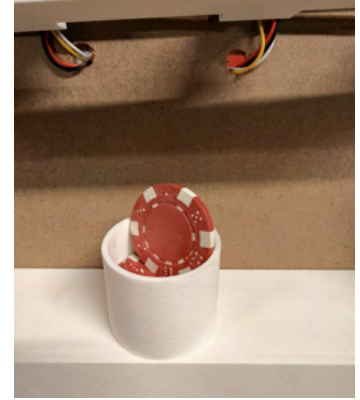
2) Chip hopper puts chips into line



3) conveyor belt to color sensing



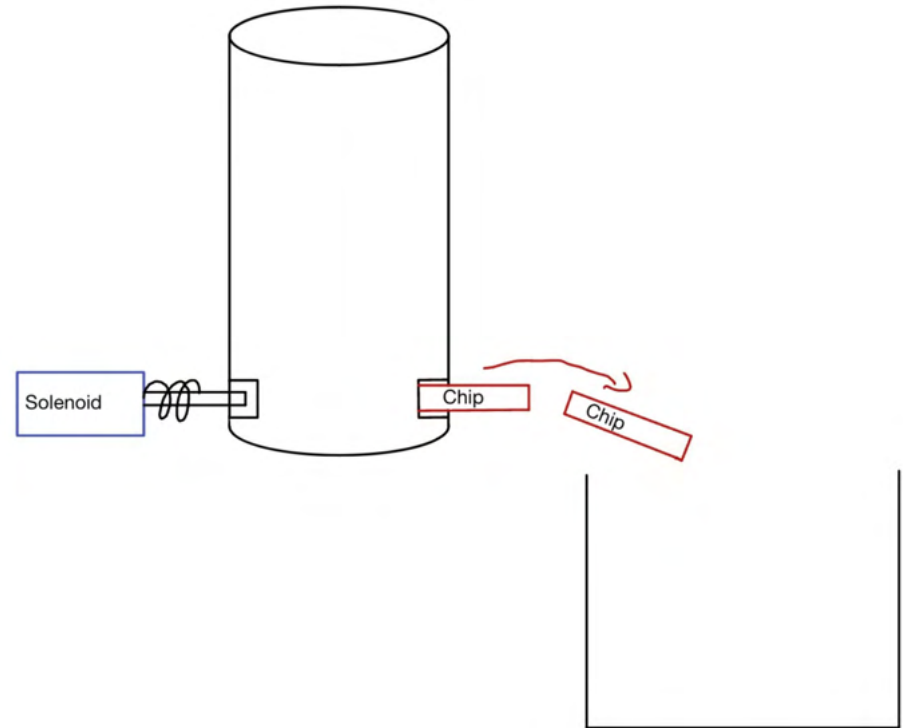
4) Color Sensing



5) Stacks of sorted chips

Complete Solution - Chip Return

- Arduino receives a chip value it must output
- Figures out what combination of chips is needed to pay out that value
- Solenoids at the base of the stack will push out needed chips



Complete Solution - Dealing

- Dealing will be handled by the automated card dealer (Dealtron).
- Each card is equipped with a unique ID mapped RFID sticker.
- To ensure game accuracy, the code limits players to two distinct cards.
- An RFID reader is placed beneath each player's card landing spot.
- Dealing begins when the Raspberry Pi signals the start of a new round.
- Players must manually reset the cards at the start of each round if they wish to comply with proper poker rules.



Complete Solution - Automated Players

- Offline blueprint trained via Monte Carlo CFR in C++, scales from 100K to 12.4 billion iterations at the highest difficulty, matching the original Pluribus paper
- 169 lossless preflop buckets (suit isomorphism) + 200–500 postflop buckets via k-means on hand strength, positive potential, and negative potential features
- Fine-grained action abstraction: up to 12 bet sizes preflop, with pseudo-harmonic mapping for off-tree opponent actions
- Real-time depth-limited subgame solving during play, maintains Bayesian beliefs over 1,326 possible opponent hands, updated after every observed action
- 4 continuation strategies at search leaf nodes (unbiased, fold/call/raise-biased) to bound opponent play beyond the search horizon
- C++ engine with 64-shard lock-free info set store, Linear CFR discounting, and negative-regret pruning

Game Frontend Display

- The central screen displays game state, player bets, pot size, and community cards.
- Screen is touch screen which allows players to manually make corrections as necessary.



Project Management

[illegible]

Testing and Verification

Dealtron Dealing Speed

The average dealing time was 4.23 seconds, based on 15 tests.

AI Response Time (100K hands)

Avg	54ms
p95	1.4s
Max	17s

Color Sensor Testing

Chip Color	Chips read	Correct readings	Misreads	Accuracy
White	150	150	0	100%
Red	150	149	1	99.3%
Blue	200	197	3	98.5%

Lessons Learned

- Stretched ourselves too much on our ability with largely mechanical project.
- We should play to our strengths more often.
- Calculate for busy schedules for weeks leading up to carnival and carnival.
- Be more prepared for hardware limitations, and be adaptable to the amount of resources actually available