Design and Implementation of a Smart Poker Environment

Mark McKinzie, Chris Reed, Eric Sun
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA
{mmckinzi, careed, ehsun}@andrew.cmu.edu

***Abstract*—A system capable of shuffling and dealing a deck of cards to four players in the typical order of a game of poker, calculating chip values when they are stacked in designated areas, detecting cards rank and suit when placed in designated areas, calculating statistics and leaderboards for players odds to win, chip values, etc. throughout a game of poker, and displaying players hands, chip totals and statistics to both the players and audience.**

***Index Terms*—Automation, Poker, Raspberry Pi, RFID**

I. Introduction

Professional poker is played under very different conditions than a typical home game. There are dealers whose job is to shuffle and deal, and keep track of whats happening during each phase of the game. The players are more adept at remembering their cards, counting chips, and generally keeping the pace of the game up. Televised games and tournaments are even more advanced; the players and the audience are able to see at a glance the current standings, as well as the number of chips currently in play. Viewers at home can even see each players cards, and automatically calculated statistics such as the probability of winning the current hand.

Our project was made to enhance the at-home experience of playing and watching a game of poker, making the everyday game feel more like a professional tournament. We believe that we can apply several different technologies to implement features typically only seen at the more professional levels of poker. Casual players will be able to set up and play a game with ease and have the tools to keep the game flowing easily. This is a must-have product for any person who enjoys playing poker games with friends in the comfort of their own home.

We feel that competition in this field is limited. For players who want the smoothest possible experience, the only real competition for our project is online poker. However we hope that our project appeals to players who still want the social experience that live poker provides. There are automatic card shufflers that are sold on their own, but ours is the only physical version of a poker game that shuffles, deals cards, and keeps track of every players chips and cards in real time while the game is being played. We believe that this, our tight integration with the game itself without straying from traditional live poker, is what sets us apart from any competing technologies or products.

The goals of this project are to:

1) Shuffle a full deck of cards and deal out hands to four players as well as the community cards in the proper order for Texas Hold 'Em
2) Calculate the exact monetary value of a stack of chips placed onto a detection zone within one second of placement
3) Detect the rank and suit of a playing card placed onto a detection zone within one second
4) Display player hands and chip values within one second of placement on both player E-Ink displays and audience displays
5) Calculate statistics for each player based off of their current hand and the community cards

II. Design Requirements

Our design is comprised of two primary physical parts. The first is a play area that will be placed in front of a player, or alternatively, in the center of the table where the pot and community cards are. It will house the chip detection, card detection, and e-ink screen. There will also be a central device that houses the shuffling and dealing mechanisms.

We will fabricate poker chips which when stacked can have their monetary value read. In order to accommodate this, we formulated an equation to calculate the resistance a chip should have based on its value. When the chips are stacked they will form a parallel circuit whose resistance can then be used to find the total value of the chips. To find the resistance an individual chip should have, we use:

$$\text{resistance} = \frac{100000}{\text{chip value}}$$

In order to diminish potential error from stacking many chips with a high tolerance, resistors with less than 1% tolerance were selected to embed in the chips. For reasonably sized stacks of chips with realistic values, this should eliminate errors. Testing will be done with varying numbers of chips and values of chips to ensure any stacking method will provide the correct monetary value of chips, within one second.

We will also modify existing playing cards with RFID tags to mark their rank and suit. We will then used RFID readers to read that data. There are 4 different suits, and 13 different ranks; we will use 2 bits to encode the suit and 4 bits to encode the rank. This means our RFID tags will need at least 6 bits of data storage. We will test that within roughly an inch of their intended locations, cards can be read 100% accurately within one second.

We will create an automated dealer modules that deals a controllable number of cards to any one of many trays surrounding the base. The automated shuffler should accept two stacks of cards near the top, and interleave them into a single stack that is able to be dealt by the dealer portion. It should interleave them such that no more than 3 cards from either pile occur sequentially in the shuffled deck. The modules will be controllable from the central computer.

We will add to each play area an e-ink display. It will show to each player their cards, total number of chips, overall rank, and indicate whether or not theyre the big blind or small blind. The e-ink screen should refresh in real time, as soon as the play area receives new information, within the constraints of the hardware (e-ink screens have relatively low refresh rates).

The above elements, minus the shuffling and dealing devices will be combined into a play area which incorporates all of the preceding requirements in an aesthetically and visually pleasing manner. In addition the play area top mimics the felt of an authentic poker table. It should be able to be placed either on a dedicated poker table, or any other flat surface.

Finally we will create a central computer that acts as a coordinator between all of the other elements. It will keep track of game state by pulling data from each of the play areas, and updating them with new game information. It will also provide two different web browser-accessible views. One is a table view for the players that will show overall standings, as well as statistics like the current ante, how many chips are in the pot, and what the result of the last hand was. The second is an audience view, like a TV broadcast might have. It will have all the same information that the table view does, but also additional statistics, like win probabilities for each player.

III. Architecture And Principle Of Operation

The game coordinator is the central brain of our project, with its instantiation on the central computer, the Raspberry Pi 3. It will operate on the Raspberry Pi Zeros in the player areas as well. The game coordinator will be in charge of keeping track of the game state, monitoring chip totals, player and community cards, player displays, and ensuring the shuffler and dealer provide the right number of cards to each area. It will also calculate game statistics, such as pot odds and win probabilities, to be displayed for the audience. The communication layer between the instantiations on the Pi Zero and Pi 3 will be written using the gRPC framework which will allow us to write two versions of the game coordinator that communicate seamlessly.

The chip counting subsystem of our project involves stacking chips with embedded resistors in parallel to allow totaling of these based on the total resistance value. A current shunt will measure the current flowing through the chip circuit which when combined with the voltage output by the Raspberry Pi Zero will allow us to calculate the total resistance. We will write a current shunt driver on top of an existing I2C driver in order to communicate the monetary chip value to the game coordinator running in the Raspberry Pi Zero embedded device.

The card detection subsystem involves RFID tags and readers in order to determine the rank and suit of cards in each players hand and the community cards. Each card will be given a distinct tag value which will be read by the reader embedded in the play area. We will write an RFID reader driver on top of an existing SPI driver which will communicate the card ranks and suits to the game coordinator running in the Raspberry Pi Zero embedded device. The RFID readers will be daisy chained to the e-ink displays since both use the SPI communication protocol.

The e-ink displays display information to each player, such as their chip total, cards in their hand, their overall rank at the table, and if they are big or small blind. These displays will communicate to the Raspberry Pi Zero through SPI, like the RFID reader, so they are daisy chained together. We will write an e-ink display driver on top of the same SPI driver that will be used for the RFID reader as well. These two drivers will work together to relay information between the game coordinator and the display.

The automated shuffler and dealer will use two high speed high torque DC motors to shuffle two stacks of cards into a single interleaved stack before utilizing two 360 degree rotational servo motors for dealing. One to rotate the module to the six designated dealing areas (four for players, one for burn cards, one for community cards) and the other to deal cards one at a time from the bottom of the shuffled deck. All of these motors will be driven by a motor driver (which can drive two DC or one servo motor at once). This motor driver will be controlled by either a PWM driver which will communicate with the game coordinator in the case of the servos, or simply a voltage output from the Raspberry Pi 3 for the DC motors.

The audience and player views will both show player information such as chip totals, player rank, and big and small blind locations. The audience view will have additional information including each players cards and statistics, such as pot odds and win probability. This information will be provided by the game coordinator to a web server that will provide a backend for both the player and audience screens.

IV. Design Trade Studies

In the poker chip subsystem, a design decision to use parallel resistance to calculate chip value rather than using computer vision was decided on. A driving reason behind this was to accommodate the ability to stack chips, which is the standard method of chip organization in all poker environments. In addition the circuitry implementation is fairly simple, whereas the computer vision approach would have been more complex, as well as requiring either multiple cameras, one for each chip area, or detecting in multiple zones. The embedded resistor chip is functional, though thorough testing has not been conducted as of yet. The test plan involves testing the resistance of individual chips, as well as stacks of like chips, which is how most poker players organize them, and finally stacks with multiple chip values to ensure any chip combination will read out accurately. The validation metric for this subsystem is 100%, meaning the correct monetary value of chips should always be correctly calculated.

We also had to make a design decision concerning how we were going to measure the resistance of each stack of chips. Originally we planned to measure the charge and discharge times of an RC circuit and use our embedded device to then calculate resistance based on those values and the capacitance of our capacitor. However, we had concerns about the accuracy of this system, so we also investigated using a current shunt, a type of current sensor. Ultimately we decided that although we would have to write an additional driver to use the current shunt (an I2C device) the extra accuracy would be more than worth it.

In the playing card subsystem both RFID tag reading and computer vision solutions were considered. Since the computer vision solution would have required some sort of indicator on the cards to reveal their value to the camera, which would also be visible to the players, the RFID reader route was chosen. Once choosing the RFID solution, further choices were needed to determine whether anti-collision would be used or simply having one card per reader area. Once again 100% accuracy is the validation for this subsystem, as misreading a card would greatly detract from the player experience, causing certain hands to be ruined.

In the play area subsystem we debated whether fabricating an actual table would be worth the work, but quickly determined that the main aspect of the play area that is necessary is a poker table felt mat type of surface, which can exist without a table surrounding it. Testing this subsystem will simply be a matter of ensuring all the data from the card and chip subsystems is read correctly by the embedded devices in each player area and the community area, then is relayed properly to the central computer.

The shuffler and dealer subsystem will have to be fabricated from scratch, as buying it off the shelf was really the only other option for the project. Whether all of this fabrication will be done using 3D printing or some will simply be manually cut and assembled is a design decision that still needs to be determined, as manual assembly will be cheaper but more time consuming. The final solution will likely involve some of both, 3D printing for the parts that need to be precise, and manual assembly for others. Testing of this subsystem will involve monitoring the interleaving of cards after shuffling, as well as ensuring 100% success of the dealer dealing the correct number of cards and to the correct tray for a player or community area.

For the computation, we had to make the decision whether to have a single central computer that all the sensors were connected to, versus embedding a device in each play area. Ultimately we decided that having a Raspberry Pi for each set of sensors was going to be the most effective. It will help us avoid significant difficulties associated with running so many SPI and I2C devices off of a single platform, and it also allows us to eliminate a wired connection between the play areas and the central computer. This is largely an improvement to the aesthetics of the project as opposed to the functionality, but when thinking of it as a potential consumer device, it is an important consideration.

Once we decided to embed computation with the sensors we also had to consider what method we were going to use to allow all of our devices to communicate. One option was to run a web service on each device and then allow them to query each other via a wireless network. This had the benefits of being modular and extremely flexible. However, we ultimately decided to use gRPC. While less flexible, it gives us the ability to call functions on remote devices as if they were local. This will make the code simpler and more readable, and allow us to write it as if it was a unified piece of software, as opposed to one part that runs in the play areas and one on the central computer.

Another tradeoff we had to consider was how to implement the different views that we expose to the players as well as the audience. Our options were either to implement them natively in Python and have them communicate with the game coordinator, or add an additional web server and write two web front ends. A native Python interface would be faster, and would incur less overhead because it would just be able to be displayed on a monitor connected to the Raspberry Pi. It would also be simpler to develop and deploy. A web-based approach would give us more flexibility in terms of technologies and likely produce a nicer looking final product. It would also force us to add an interface to the game coordinator, something that would involve more work but likely be useful and make our code better. Finally, and crucially, it would allow any number of people to view either of the views on any device with a web browser. Ultimately we felt that this flexibility was worth the extra overhead both in creating, and running the system, so we decided on implementing the web-based approach.

V. System Description

A. Player Area

1) Card Detection System: To build our card detection system, we purchased a deck of Bicycle standard playing cards along with MIFARE 13.56 kHz passive RFID stickers to modify the cards with. We adhered one RFID sticker on the center back of each playing card in order to facilitate our detection, and then encoded a bit code relevant to the cards respective rank and suit onto the RFID. To read and write the RFID stickers, we utilized an MFRC RC522 RFID reader sensor module that was compatible to be hooked up to our Raspberry Pi Zero we have in each Play Area. We attached the Raspberry Pi Zero to a T Cobbler to connect the Pis GPIO pins to the RFID chip, which allowed us to feed in the bit codes read from the RFID reader to the Raspberry Pi Zero to obtain the data on what cards were in the detection zone that respective Pi was in.

2) Chip Detection System: For Chip Detection, we started off purchasing a set of poker chips off of Amazon to modify. We designed our modified chips to have two concentric conductive areas, one in the center of the chip and another ring around that, connected with a resistor. This design was chosen so that we would be able to create a circuit of parallel resistors when we stacked the chips on top of each other, as typical poker players do. As we were drilling into the chip to hollow out sections for our resistor and conductive rings, we discovered that there was a small metal layer in the center of the poker chips. We had to adjust to buying lower quality poker chips so that they would solely be made of plastic, then sandwiched the low quality chips with the high quality ones to facilitate better contacting between stacked chips. After this adjustment, we were able to drill out sections in each chip to fill with our conductive rings and resistor. Once the chips were constructed, we placed sets of two conductive rings on our play area that were connected to the Raspberry Pi Zero in each player area. We placed an INA219 current shunt along the connection from the chip stack areas and the Raspberry Pi so that we would be able to know both the current of the stack of parallel resistors and the voltage the Pi was supplying. Then using R = V/I, the Pi was able to obtain the data of the total resistance of the chip stack, which the Pi is able to convert into monetary value of the chips using our conversion formula (value $= \frac{100000}{\text{resistance}}$).

TABLE I
CHIP VALUES AND RESISTANCES

Chip Value ($)	Resistance (k)
5	20
10	10
50	2

We chose the resistor values shown in Table I to be high resistance in order to reduce error in the chip value calculation. In addition these were selected because they are standard resistor values and these multiples fit well into our scheme for the monetary value of our chips.

3) E-ink Display System: We are placing a 2.13 inch flexible E-ink display HAT that is compatible with Raspberry Pis in each play area. This will be connected to the embedded Raspberry Pi Zero in each respective play area, and will be used to display the players hand, chip totals and rank, or the community cards and pot value.

4) Embedded Device System: We used a Raspberry Pi Zero in each player area in order to run and keep track of data from both our card and chip detection systems. Once it obtains the data of the players cards as well as the monetary value of the players chips, the Raspberry Pi Zero sends this information to the e-ink display in the player area to display this information to the player, as well as sending it to the central computer (Our Raspberry Pi 3B) for stat tracking as well as further statistical calculations. The embedded Pi runs a version of the game coordinator software that interacts with the game coordinator on the central computer via gRPC. The game coordinator will be written in python.

To communicate with the sensors, we will write a series of drivers that expose a simple interface to the game coordinator. The current shunt driver will be written on top of an existing I2C driver, and will allow the game coordinator to query the total current resistance of the system. This will allow us to move the configuration of chip values into the game coordinator. The RFID reader driver will work largely the same way, except it will be written on top of an existing SPI driver. It allows the game coordinator to query all of the tag data that is currently in range of the sensor. This means we can pull the mappings from tag data to cards into the game coordinator as well. Finally, we will write a driver for the e-ink display. This will allow the game coordinator to write out some of the sensor data, as well as information from the central computer, to the screen. The drivers will be written in python.

B. *Card Shuffling/Dealing*

1) Shuffling System: Our shuffling system is divided into two symmetric trays that will take in approximately half a deck of playing cards each. Each shuffling tray will contain a 2900 RPM DC motor which will each drive a rubber wheel. This rubber wheel will thrust the cards into the center of the module from both sides where they will combine into a full deck, alternating cards from each side. Once in the center of the shuffling module, the cards, now interleaved into a single full deck can be placed into the dealer tray and subsequently dealt into the proper player and community trays.

2) Dealing System: The dealing system will utilize two FS90R 360 degree continuous rotation micro servo motors. One motor will deal cards one at a time to each player slot and the community and burn slots in our dealing area. An attached wheel with rubber tread will allow the cards to be gripped and dealt easily. The other motor will rotate the entire dealing system to facilitate the dealing to each tray of the unit, consisting of a cylinder topped by the tray for the cards to rest in. The dealing motor and wheel will rest in the edge of the cylinder, whereas the rotation motor will be underneath the cylinder.

3) Enclosure: We designed an enclosure to house both the card shuffler and dealer in one unit in Solidworks. There is a one foot diameter circular base which has six sections with a tray for a card to be dealt into. In the base there is a six inch in diameter circular cut out where the dealing device will be housed, so that there is easy access for it to turn to every players card tray. There are also six vertical supports placed between each tray connecting an upper platform to the dealing base. This is where the shuffling system will be placed, with the cards it processes being deposited directly into the feeder tube of the dealing system. The dealer enclosure consists of a tray to hold the cards on top of a 4 inch diameter cylinder which will rotate to each dealing area. Most of the enclosure for our shuffler and dealer is hollow to save money on how much materials we use to produce it. Due to the limited budget we had, we were not able to produce the full model of our shuffler and dealer unit. We instead printed the dealing mechanism with two trays to deal cards into, as well as the two card trays to hold the decks when we shuffled them together.

We also designed enclosures for both the play area and community areas of our project. We laser cut wooden covers to put on top of the circuitry for the card detection and Raspberry Pis. The covers had etched card detection zones to show the user where to place their cards for proper detection, as well as an installed a conductive chip on the top where the players could stack their chips on top to track their monetary value.

4) Central Computing System: The central computer is a Raspberry Pi 3 Model B device. It is in charge of managing game state and driving the shuffling and dealing device. It will run a game coordinator that keeps track of all game information and pulls data from each of the play areas via gRPC. Like those devices, it will be connected to a wireless network so a wired connection is not required. The game coordinator will also calculate statistics, and send information like current standings back to each of the play areas for display. The game coordinator will be written in python.

The central computer will also control the shuffling and dealing device. We will write a software driver for the motor controller that will in turn drive the DC motors that shuffle, and the servo motor that deals the cards. The interface exposed to the game coordinator will let it specify the number of cards to be dealt and the direction in which to deal them. The driver will be written in python.

Finally, the central computer will provide the table and audience views that contain standings, statistics, and other game data. It will run a lightweight web service that pulls information from the game coordinator, and exposes an endpoint to access that data. We will also create two separate frontends, one for the table and one for the audience that query that endpoint. The web server will be written in python.

VI. Project Management

We divided up our schedule into each component we needed to implement for our project. We had six different components: the poker chips, playing cards, play area, shuffler/dealer, embedded devices and central computer. At the beginning of the project, we wanted to focus most on getting our chip detection and card detection working, as they are the main factors that go into playing a poker game. We wanted the physical parts and the software to run them to be done first so we will have plenty of time to test and refine these designs. One thing we had to adjust in our schedule was the timeline for the shuffler and dealer. We had to wait for our parts to come in before we could start implementing the cards and chips, so we pushed up the design of the shuffler and dealer to earlier in the project so we could have a Solidworks model done by the time we began start working on our other parts.

We split up primary responsibilities by part for this project. Marks primary responsibility was designing and constructing our modified poker chips. Chris primary responsibility was putting together the RFID playing cards. Mark and Chris shared the secondary responsibilities of designing and fabricating both the play areas and the shuffler and dealer. Erics primary responsibilities were the embedded devices, which run our play areas, and the central computer, which takes data from all the embedded devices and computes statistics.

Chris also compiled our bill of materials, keeping track of the parts we needed and their cost. We used every part we purchased, and you can see a breakdown of what parts we bought for each component in that document.

Our main risk for this project was not having our chips and cards be able to be detected with 100% accuracy. These two components are the main parts of a poker game, so we needed them to be fully functional and reliable for our project to be a success. We planned for this by focusing solely on getting those individual pieces working at the beginning of our project, so we had plenty of time for refining and testing. A secondary risk was the shuffler and dealer modules not properly combining together so the shuffler would drop cards straight into the dealer. We mitigated this risk in the end by optimizing the dealer and shuffler separately and though it added the manual step of moving the cards from shuffler to dealer, it was much more reliable than the combined module.

VII. Related Work

We took some inspiration for our dealer module from the Catanoumous project from a previous 18-549 class, which influenced our design to deal from the bottom of the deck rather than the top, as well as choosing a similar servo motor with wheel attachment for gripping the cards and dealing.

We also observed how a commercial automatic card shuffler operates in order to base our design off of something we know already functions well. A project done by former CMU students was to disassemble and analyze an automatic card shuffler, so we used this inside look into the shuffler to better our design. A student at USF created a card shuffler utilizing 3D printed parts which also influenced our card tray design.

VIII. Summary

Ultimately, while the individual components of the finished system met the design specifications, there was some difficulty involved in getting them to all work together, and the systems performance was varied. In large part, these differences were caused by hardware that was of a low quality, and software that was unable to fully make up for these deficiencies. The system had a few limits on its performance. First, card detection had a very limited range, and we experienced issues where cards were bumped outside of their areas, causing issues. Second, our chip detection experienced minor interruptions in performance where no chips were detected; these issues were caused by imperfections in the chips we fabricated. Finally, the shuffler and dealer did not work every time they were supposed to.

One simple approach to improving performance would be to research and purchase higher quality components. This might include things like high-accuracy stepper motors rather than the servos we purchased, RFID readers that had a larger range, and redundant current shunts so that multiple could be used to obtain a single more accurate measurement. We would also spend more money to fabricate all of our 3D printed components (with our limited budget we had to forgo some of them) and make the ones we did have out higher quality.

Another approach might be to try and compensate even more in software for hardware deficiencies. While we implemented options like repeated sampling and outlier removal, there are other more complicated options we could introduce.

A. Lessons Learned

One lesson we learned was to test hardware early enough to have extra time in case it was unexpected. This would have likely helped us avoid or mitigate some of the issues we faced. Another lesson would be to integrate hardware and software as quickly as possible. We waited too long to do that and it ended up being more work than we expected. Finally, we would have worked on our presentation for the demos more. We felt that while we were able to get a lot of work done in the week or so before them, our pitch was somewhat neglected.

REFERENCES

[1] https://www.archive.ece.cmu.edu/ ece549/spring16/team08/website/

[2] https://wiki.ece.cmu.edu/ddl/index.php/automatic_card_shuffler

[3] https://wiki.ece.cmu.edu/ddl/index.php/Automatic_card_shuffler_2

[4] https://www.instructables.com/id/Automatic-Card-Shuffler/

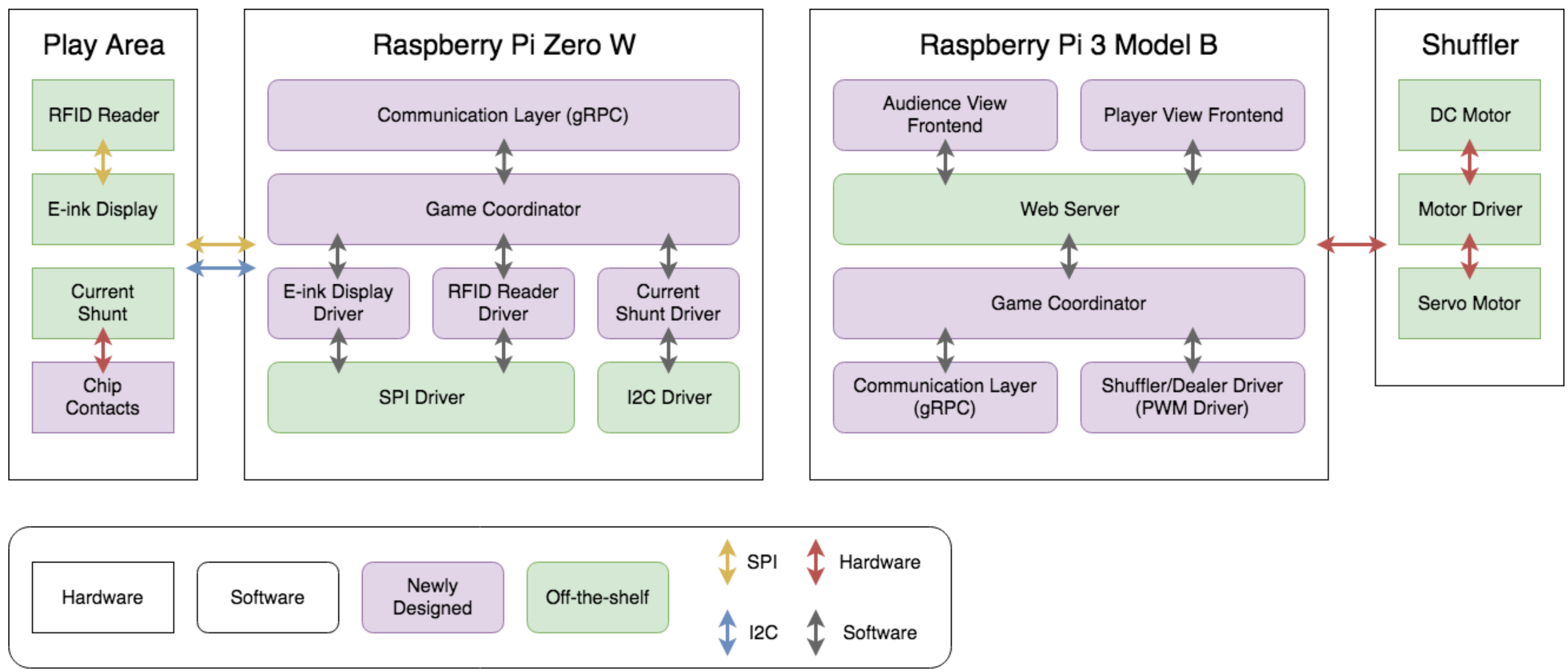

Fig. 1. Project block diagram.

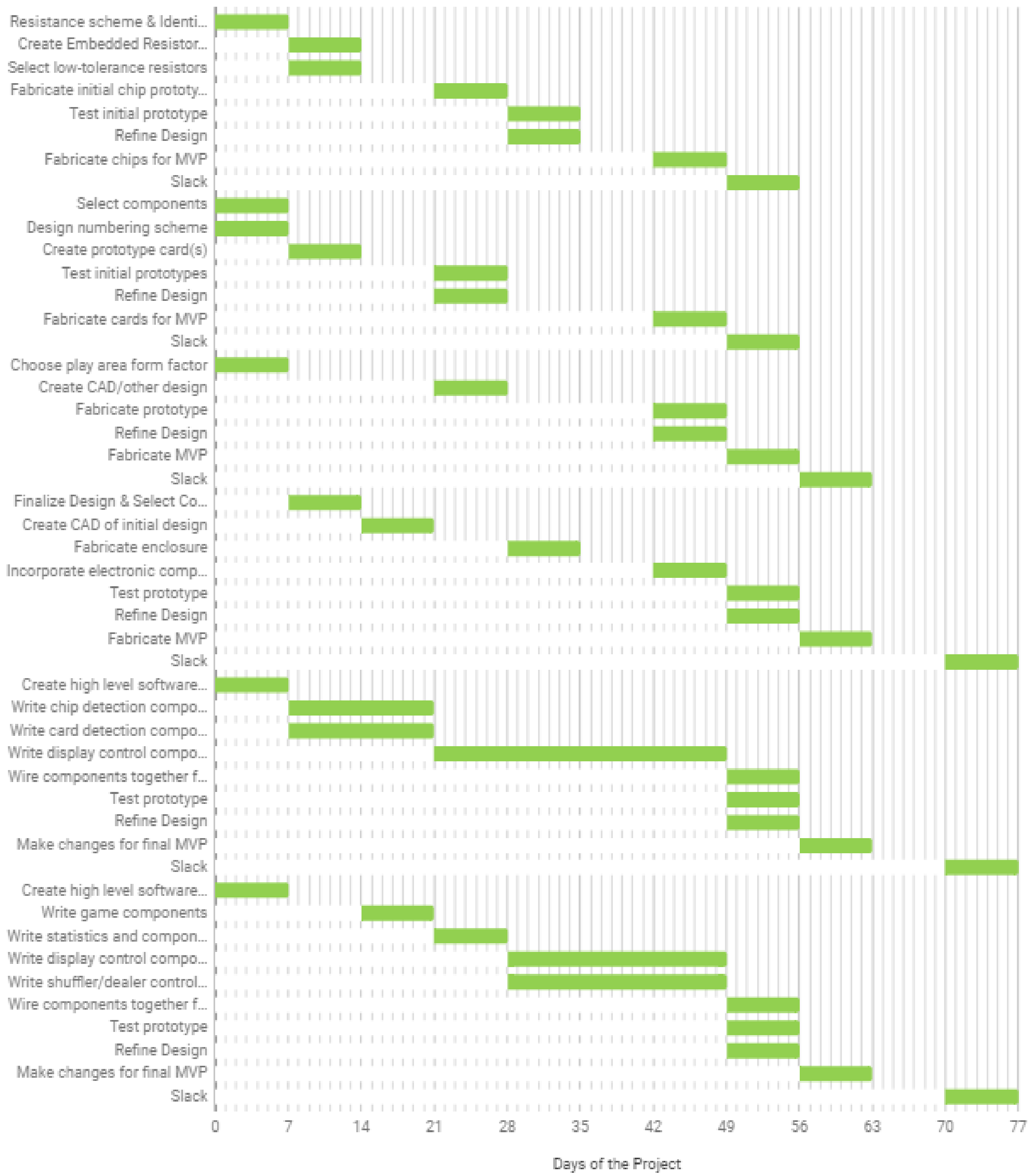

Fig. 2. Project schedule.

TABLE II
BILL OF MATERIALS

Requirement	Material	Quantity	Cost	Link
Card Reader				
(obtained)	RFID Reader/Writer	9	$45.96	[MFRC RC522 RFID-RC522](https://amzn.to/2GIBArN)
(obtained)	RFID Tags	100	$48.89	[13.56 MHz RFID Sticker](https://ebay.to/2toNuPz)
(obtained)	Deck of Cards	4	$9.39	[2 Pack of Cards](https://amzn.to/2BBySRs)
Chips				
(obtained)	20 kOhm resistor	20	$0.00	From the lab
(obtained)	10 kOhm resistor	20	$0.00	From the lab
(obtained)	2 kOhm resistor	20	$0.00	From the lab
(obtained)	Poker Chip Set	1	$16.15	[100 Piece Chip Set](https://amzn.to/2SJlns)
(obtained)	Poker Chip Set	1	$6.36	[100 Count Bicycle Chips](https://amzn.to/2SHcv2H)
(obtained)	Copper Tape	1	$11.90	[Copper Foil Tape](https://amzn.to/2SHcv2H)
Shuffler/Dealer				
(obtained)	Shuffler Motors	5	$9.75	[Shuffler Motor](https://bit.ly/2U8IytU)
(obtained)	Rotating Motor	1	$24.27	[High Torque Metal Gear](https://bit.ly/2SP9Vfl)
(obtained)	Gears	1	$8.99	[78 Piece Set](https://amzn.to/2Ixa9nm)
(obtained)	DC Motor Driver	3	$8.85	[Motor Driver](https://bit.ly/2GGcHhj)
(obtained)	Dealer Wheel	1	$24.98	[Micro Servo RC Wheel](https://amzn.to/2XlC90z)
(obtained)	New Shuffler Motor	2	$5.50	[Shuffler Motor](https://amzn.to/2DlZU0g)
	Enclosure Material	1	$134.83	3D Printing $0.35 per ounce
(printed)	Player Tray	1		
(printed)	Dealer Cylinder	1		
(printed)	Dealing Tray	2		
(printed)	Shuffling Tray	2		
(printed)	Dealer Bottom	1		
Play Area				
	Wooden Ply	2	$10.00	Art Store
Embedded Device				
(obtained)	E-ink Display	2	$42.98	[2 inch E-ink Display](https://bit.ly/2S49zgA)
(obtained)	Raspberry Pi Zero	2	$127.71	[Pi 0 Starter Kit](https://bit.ly/2BxOIwu)
(obtained)	Current Sensor	3	$45.04	[INA219 DC Current Sensor](https://bit.ly/2UWFh0K)
Central Computer				
(obtained)	Raspberry Pi 3	1		[Pi 3 Starter Kit](https://bit.ly/2GFq9kM)
Complete Project		1	$581.55	