AutoCA(r)D

Marco Túlio Giachero Pajaro and Theodoro Haddad

Introduction

Context

The AutoCA(r)D is an automated and programmable approach to shuffling cards. This device is controlled by user inputs in terms of shuffling settings from a bank of algorithms available on our website. In this way, the device can be adjusted depending on the rules of a given card game, making it dynamic and customizable.

This project consists of a digital simulator for shuffling cards and a 3D printed card shuffler that connects to our simulator and orders the cards in the set output from our algorithm. This way, we can create the random order of cards first using the parameters that the user wants and then translate that into our physical deck of cards.

Motivations and Goals

As avid Truco players, we found a mutual interest in card games during a conversation in one of our Honors Seminar learning circles exercises. From that, we both recognized that cheating is a big problem when shuffling a card deck in unofficial Truco matches. In this way, we saw in that problem an opportunity to combine our knowledge in Electrical Engineering and Computer Science to come up with a device capable of adjusting the

impartiality of a shuffle by user inputs. Therefore, what we want to accomplish with our prototype is to manipulate the “fairness” of a given shuffle, from 0 to 100%, with variable shuffling parameters. In other words, the user will be able to set a shuffle pattern to their liking or randomize them, so that card counting is encouraged or discouraged, depending on what their objectives are.

Applications

This project has multiple different applications both in the physical and theoretical sense. Firstly, the most obvious one is to use it as a shuffler for any card game. Secondly, to explore randomness that could be generated in a computer and have customizable parameters that then can translate it into the order of a physical deck of cards. As a consequence of the adjustable “fairness” nature of our project, the second application our system has is to be used as a card counting tool. Additionally, another important dimension that our project touches on is combining software and hardware to improve accessibility. In this case specifically, the AutoCA(r)D can also be used for card players with motor impairments or other conditions that make it hard for them to physically shuffle cards effectively.

Outline of Problem Addressed

The main problem we tried to address was exploring the discrepancies between real world and simulation randomness. That is because by nature, computers are deterministic machines, which means that in order to generate random numbers they still have to follow a certain pattern or algorithm. This results in something called “pseudo random” numbers.

In order to explore randomization with a physical card shuffler we had to make use of 3D printed parts to house the cards themselves, motors to move them, and a microcontroller to translate the outputs of the randomization algorithms processed and hosted on our website to the rotation of the motors. This was a challenge that lay at the intersection of Computer Science, Electrical and Mechanical Engineering, making it a cross-functional project that made us use skills from both of our majors and beyond.

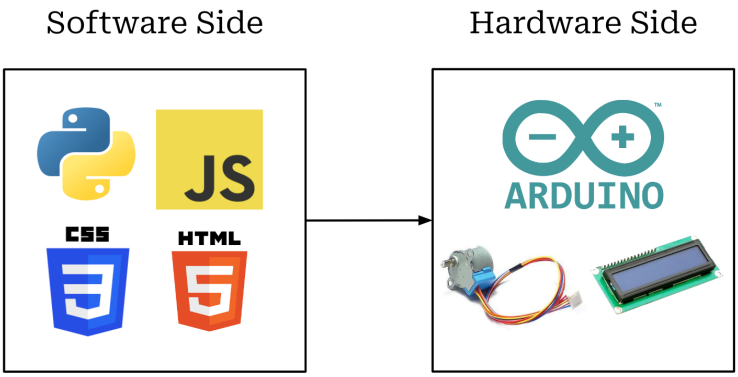
Our project was very peculiar in the sense that not only we had to work with the physical design of the system, that would be responsible for reorganizing the order of the deck of cards, we also had to ensure compatibility with the shuffling methods used. That means that the system has to run smoothly as theory suggests, otherwise it would be unrealistic to use the device as a card counting aid. In this way, we decided to program a simulator on our website to use it as a testing tool to compare the order of the cards from our physical device to the theoretical one according to the algorithms used.

Next, we decided to give emphasis to the theory of shuffling methods and how that translates into code. In this way, we resorted to computational math papers published that explored the ideas of randomization to conceptualize in a high level view what our algorithms should be able to perform. While building and testing our online card simulator, we explored a pseudorandom algorithm from the Math library in JavaScript that had its limitations. Then, we came across another algorithm that we found to be the closest to real-world randomness we could find, called the Fisher-Yates algorithm for shuffling cards. With this algorithm, we got closer to the "fairness" that we were looking for in shuffling cards, but still had to deal with pseudo random issues that are bound to happen when simulating randomness using a computer, since at the end of the day every algorithm is somewhat deterministic.

In this way, from the simulations we ran on our website, we noticed that we were able to accomplish a decent enough randomization with the Fisher-Yates algorithm that in a real world scenario, in a non-professional card game level, it would be really difficult to distinguish if the cards were shuffled by a human or a machine.

Methods

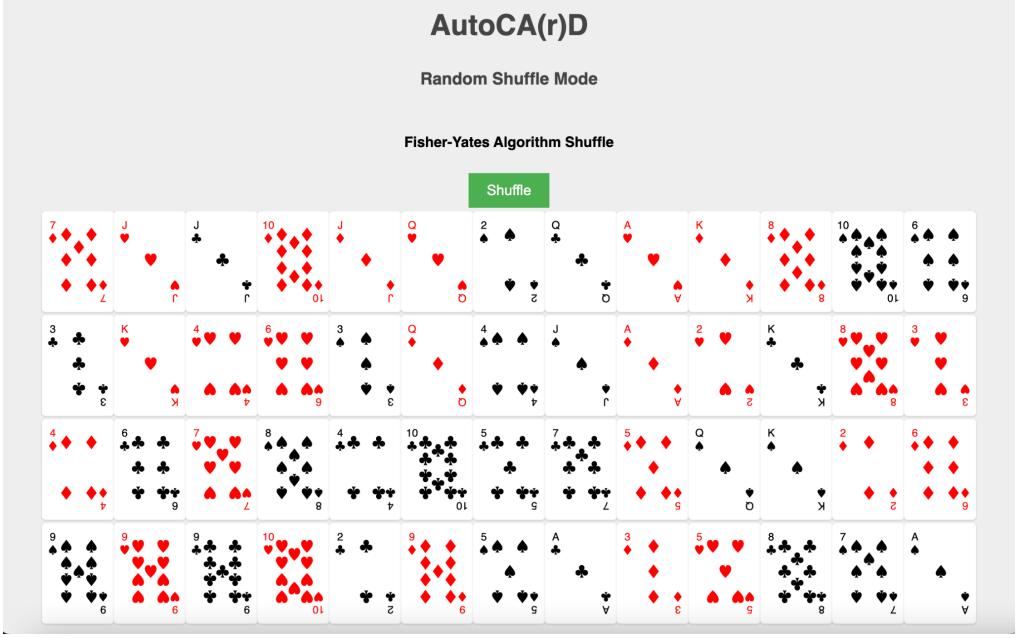
System Design



This image highlights our system integration for the card shuffler that we built. Here, we have the software side which deals with the randomization exploration and algorithms as well as the hardware side, which deals with the physical shuffling of the deck of cards using motors as well as matching the card order that our software side decided on.

Website

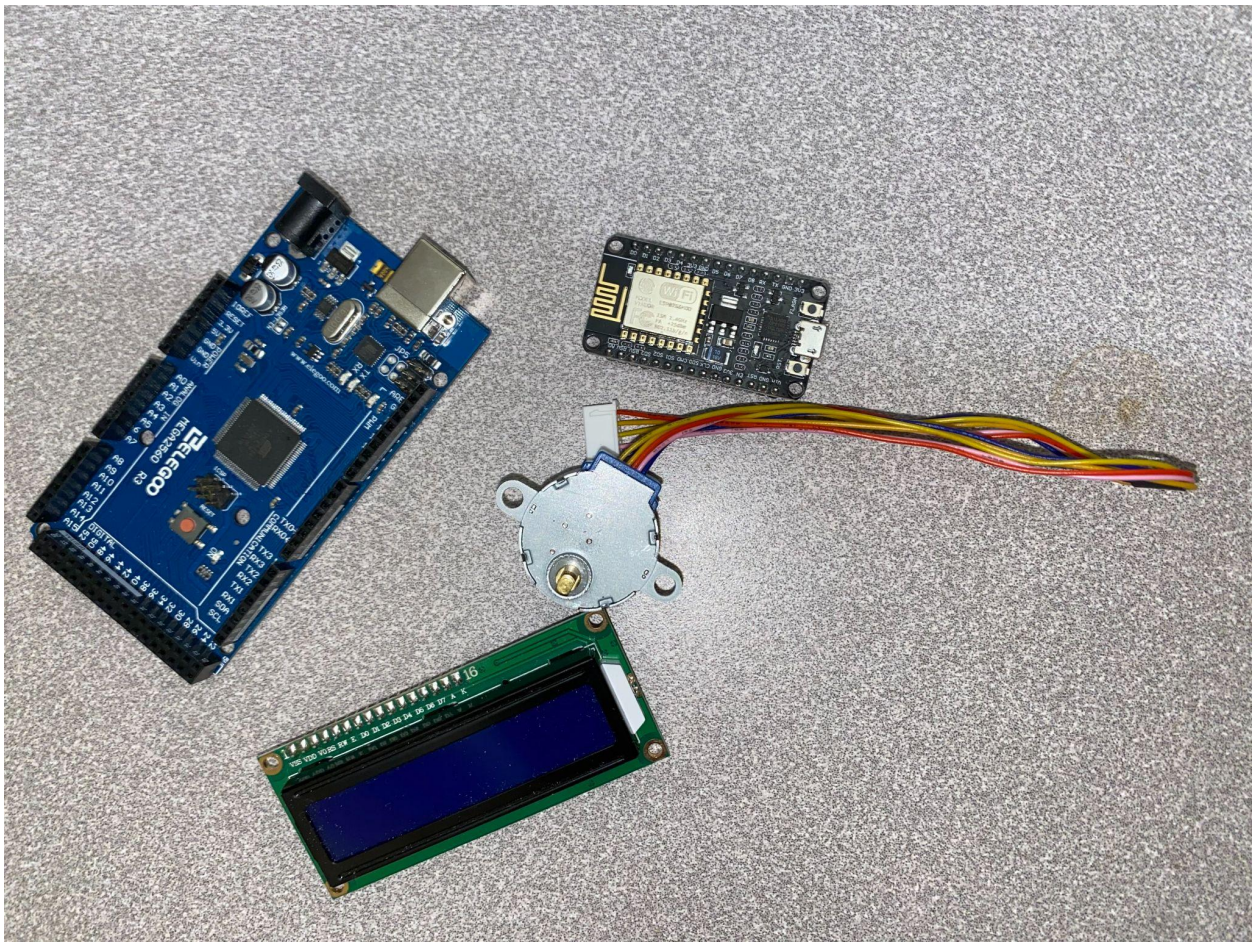
Our website was responsible for hosting the user interface, responsible for configuring the system, as well as the simulator. The latter was built using JavaScript to simulate the card shuffler and to implement the Fisher-Yates algorithm. Then, we used HTML and CSS to build a user interface (picture bellow) where the user can choose which algorithm they would like to simulate, then display a deck of cards in the order that the shuffler will set the cards to be set in the physical deck.



Hardware

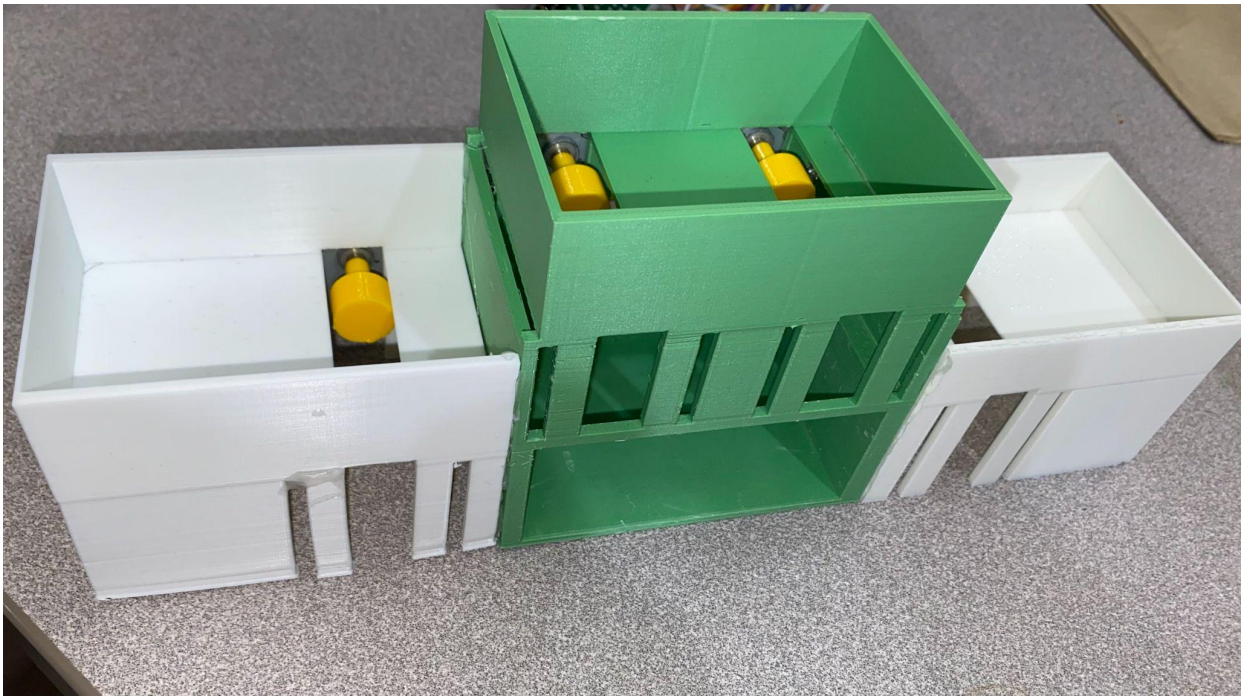
Finally, for our physical model of the card shuffler, in terms of microcontrollers, we used an ESP 8266 to communicate with the server using its wifi module, as well as Arduino Mega 2500 to convert the output of the website into rotations of the motors. The two microcontrollers communicated using UART serial protocol. In terms of motors, we decided to use stepper motors, which required us to use proper PWM on the Arduino terminals so that we could accurately move just one card at the time, instead of the whole deck. Lastly, we also programmed a 2x16 LCD screen so that we could inform the users of the status of

the system (in progress/ done).



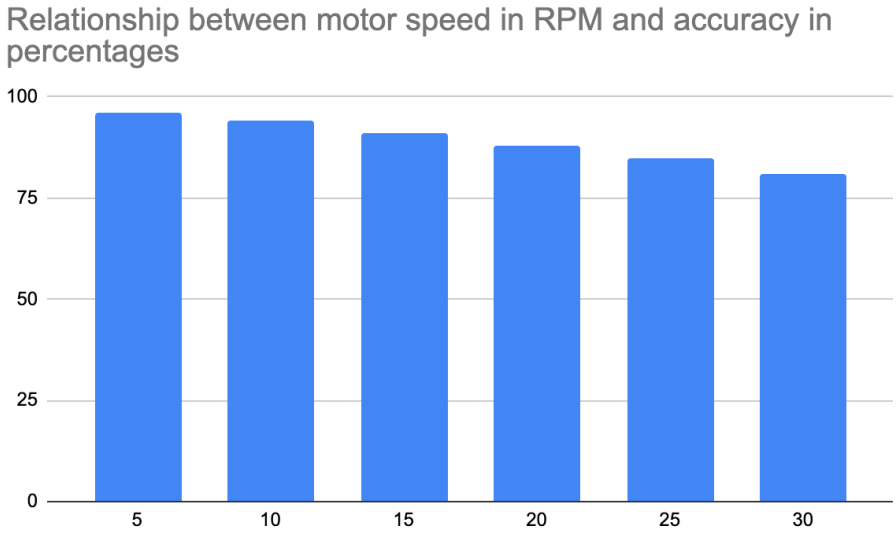
3D Printing

In order to create a compartment that could house the system, we empirically design on Fusion 360 versions of what the system could look like, also taking inspiration from similar existing projects. Each attempt was tested in a sort of trial and error attempt looking for improvements we could make in the design. Finally, when we passed the proof of concept stage, we designed the final version which we stuck with for the remaining duration of the project.



Results

Shuffling Speed and Accuracy



One of our main objectives while building this project was to make sure that our card shuffler performed accurately and rapidly enough such that it could be used in real life card games. In the graph above, we can observe the tradeoff between shuffling speed in terms of RPM and our percent accuracy (here, we define accuracy as the number of cards that matched our algorithm's output). In general, the trend that we see is that the faster you shuffle, the less accurate our shuffle becomes, so this can be used per user discretion depending on which of these two factors they value the most for their specific application.

Challenges and Improvements

Some of the challenges we faced were mostly related to the kind of resources we had available at the time. For instance, really good quality stepper motors are quite big, so we would encounter way too many mechanical difficulties implementing those to our design. In this way we decided to use smaller ones, that as a consequence are of lower quality; which then resulted in the challenge to accurately tuning them to turn just enough to move a single card from one compartment to the next one.

Another challenge we faced was in terms of the resolution of the 3D printers we had available. Initially, the slit of 0.24mm (width of a single card) was not being printed due to the extrusion of too much support material from the 3D printer. We were able to fix that problem by adding filets to the slits in Fusion 360. We also had problems with the size of the 3D printer we had available, that is because we were able to borrow one Creality Ender 3 from the 3D printing club which has a bed of 235 x 235 mm in area; the basis of our model was 280.4 x 105.5 mm. In this way, we had to split the 3D model into 3 smaller parts.

The last main challenge we faced was that the ESP 8266 WiFi module was unable to connect to the 5.0GHz channel of the University's network. We were able to solve that problem by connecting to another network on the 2.4GHz one.

The main improvements we found to be relevant to our system are: the implementation of a rotating basis with a built in motor to deal cards, an adjustable knob to control the rpm of the motors on the physical device, and a redesign of the compartment the cards go in because the weight of the remaining cards is always changing, so we noticed a trend that the cards would jam as the cards would leave the compartment.

Conclusion

Overall, our project was successful in achieving our fundamental objective of exploring randomness and fairness in card games. We managed to build a functioning shuffling simulator as well as a 3D printed card shuffler. This project really made use of many cross-functional skills both technical and non-technical. The main technical skills in use for the card shuffler were algorithm implementation, 3D printing, software embedding and hardware controllers. The main non-technical skills were communication with relevant faculty members, learning how to fail better and learn from past iterations and lastly to condense our information into a digestible form to be presented to an audience with differing technical backgrounds.