***Abstract*—Remote Chess is a chessboard that allows users to play across the internet through physical chess boards. It uses RFID technology to sense the type and movement of pieces. This information is sent to a phone’s android application via Bluetooth which then sends it through a server to be recorded and communicated to the opponent’s board. Using a custom-made board with LED indicators we will be able to display the corresponding movement on another user’s board. The intermediary android devices will also display the chessboard UI showing the potential for Board to App and Board to Board matches. The app will be able to pick up on the accidental removal of pieces from a board and the server will use the recorded moves to decide when the game reaches a conclusion.**

# INTRODUCTION

The game of chess has been enjoyed for over 1000 years with just chess pieces on a simple chessboard. However, as technology advances, so too do our forms of entertainment. Everything around us is becoming more digitized. Digitalization has made many things more connected and accessible than ever before but there is a cost. It has made many of our experiences less intimate, forcing most of our daily interactions to be behind computer monitors, touchpads and mouse clicks. The game of chess is no exception. Today, Chess.com boast over 15 million members [1].

Our goal with remote chess is to provide the best of both the physical and virtual reality, where we overlay a physical board game into a virtual space, providing an interconnected and holistic experience

. We have seen a few designs concerned with this problem such as SquareOff and Chesstronic Smart Chess. SquareOff is an internet-connected automated chess board which uses pressure sensors to detect a move and a magnet-based system to make your opponent’s pieces on your board. The problem we see with this design is that it has no capability to uniquely identify piece, so it assumes the piece above the square is correct. The utilization of pressure sensors makes the game feel unnatural as you have to press down the square before and after each move. Their implementation of magnets to move pieces has made their chess pieces unaesthetically small relative to the chess square. Chesstronic is an internet-connected LED based board with magnetic relay switches as its sensing system. Chesstronic also does not identify chess pieces uniquely and displays the opponent’s move on the board with LEDs which are in the corner of each chess square. Both projects have a combined total of over $200,000 raised on KickStarter, proving that there is a demand for such products.

Our project will be an internet-connected LED based chess board which utilizes RFID technology and LED diffusers.

System Specifications:



# Design

## Overview

Our project will be an internet-connected LED based chess board which utilizes RFID technology and LED diffusers. What makes RFIDs unique is that it will allow our board to identify individual chess pieces while providing a hands-off sensing system as opposed to pressure, photosensors and relay switches. The idea of “self-moving” pieces was out of the scope for this project, so we opted to use LED indicators to show the opponent’s move on the board. However, instead of Chesstronic’s implementation of LEDs at each corner of the chess square, we will be lighting each the chess squares with a LED-diffusion setup.

Block Diagram:

Fig 1. Block Diagram

## RFID Sensing

The chessboard will be able to fully sense both players moves using RFID readers and tags. Each chess square will have its own associated RFID reader, where the center of the chess square is aligned with the center of the RFID reader antenna. Also, inside the base of each chess piece, there will be an RFID tag that uniquely identifies it. Our main concerns with utilizing RFIDs was how to deal with RF interference and choosing a reader with the right read range for our application.

 Our specifications for choosing an RFID reader and tag was that the reader be able detect any chess piece when placed anywhere within a chess square and that the tag be small enough to fit into the base of our smallest chess piece(pawn). The World Chess Federation recommends chess squares have a length of 1.97”-2.36” to be considered tournament size [2]. We chose our chess square size and the associated chess pieces based on the square size actually played in chess tournaments [3]. Our board will have chess square lengths that measure 2.25” and our smallest chess piece, the pawns, have a diameter of 1.06” (roughly ¼ of a chess square length).

Based on the above RFID reader and tag specifications we went with the MFRC522(HF RFID reader) which also came with a tag ($≈1")$ that met our size requirement. Before we began testing the RFID reader/tag for its range and interference we used online resources familiarize ourselves with wiring the reader for SPI and using the MFRC522 coding library with the arduino uno used in the prototype. Once we were able to set up the reader with the arduino we set out to test its read range by simulating a chess square with a reader underneath it and a tagged chess piece placed on its surface. We did that by drawing a 2.25”x2.25”in square on a piece of cardboard that is roughly 3mm thick, taped the reader underneath the drawn square with the reader’s antenna aligned with the center of the square and taped an RFID tag to the bottom of a pawn (edge case for reader/tag). We tested the read range on the tag by moving the pawn to various locations within the chess square. Our initial range results showed that worst-case scenario, the reader could only read the pawn when its >2mm from the edge of the chess square. We resolved this issue by setting the power of the reader to max which allowed the reader to detect the pawn at any location within the chess square.

 Our next concern was to test for RF interference between readers. When multiple readers are within each other’s interrogation(read) range, their range is decreased [4]. We first wanted to see what kind of interference we were dealing with, so we repeated our range test but with 2 adjacent chess squares. Our results showed that in the worst-case scenario, the pawn can only be read when >3mm from the square’s edge. Our current solution (what we did for MDR) was to route the RFID readers under the black squares to one control line and the RFID readers under the white squares to another. When reading a white square, the readers in the adjacent black squares can be turned off and vice versa. This will allow us to minimize the number of parallel control lines and the amount of RF interference. However, our current solution yields a read time of about 10ms which means that it will take about 640ms to cycle through all of the squares on the chessboard.

This read delay has potential to cause our response time specification(2s<), defined as the time from when player moves a chess piece to a new square and when the LED on the opponent’s board lights up, to not be met. Possible solutions that we are looking into to reduce read delay are:

1. Besides powering off adjacent opposite color squares to reduce RF interference, make them low powered instead.
2. Become well-versed with the MFRC-522 coding library in order to optimize the code to our project.

## LED indicators

 To reflect the opponent’s move on the board we plan to light up the square with opponents piece and the square to where the opponent moved the piece. Each chess square would be associated with one addressable LED, that diffuse their light into pieces of either plastic or acrylic. The LEDs will be on a single serial protocol to minimize the number of etches on our PCB. We are still currently looking into the type of diffusing material needed and how to incorporate it into the current RFID setup

## Android Application

The android application, programmed in Java, will serve as an intermediary between the board and the server. The application will display a visual chessboard, a notification system when a move has been made and also a missing piece notification to alert the user when a piece has been removed from the board. Currently the android can communicate with a Bluetooth module and is able to display the correct square and piece ID.

## Server

We will use a server/client model where we will host a server through the Google Cloud that will connect to the android applications allowing for successful communication between the two boards. It will also allow the user to find a new chess match and it will also keep a record of the moves made. Currently the server can only record the moves made on the board but by CDR will have the full functionality.

# Project Management



Fig. 2. MDR Deliverables

We have shown a working model incorporating a 3x3 chessboard which communicates the location of pieces to an app. What we still need to do is create the full board by designing the PCB, choosing a chassis design and finding LED’s that would work with our design. Currently our team is collaborating through our weekly meetings and discussing what each of has accomplished so far. I would not say there is really expertise that differentiates us but what we prefer to work on. Currently David prefers to work on the RFID readers, Batsooj works on the LEDs and chassis planning and Shashwat works on the server and app implementation. Each of us also tends to float to other priority tasks when needed.



Fig 3. Gantt Chart

# Conclusion

Currently we have built a 3x3 chess board to prove that these RFID readers can work robustly near one another. We have also built an Android App that interfaces with our 3x3 board and correctly displays the square and piece ID. We have also seen success in uploading these moves to a server that will be able to transfer these move to a different unit whether another board or another app. We expect to face difficulties in power management of 64 LEDs and RFID readers. We may also face difficulties in the design of the PCB with so many off-board and on-board components and hundreds of etches to route between them. Although “self-moving” pieces won’t become a reality for this project, it is something that should be considered in the future.

1. Acknowledgment

We would like to thank our advisor, Professor McLaughlin, for his expertise and guidance throughout our project’s development and our evaluators, Professors Soules and Taneja, for their helpful feedback at our design reviews.

*References*

[1] https://www.chess.com/

[2] https://www.fide.com/

 [3] https://www.chessusa.com/tournament-chess-size.hml

 [4] D. Y. Kim, H. G. Yoon, B. J. Jang and J. G. Yook, "Effects of Reader-to-Reader Interference on the UHF RFID Interrogation Range," in IEEE Transactions on Industrial Electronics, vol. 56, no. 7, pp. 2337-2346, July 2009. doi: 10.1109/TIE.2009.2012451