The S.T.G Device: Smart Tabletop Gaming

Richard Almario, Hubert Barrantes, Jose Castano, and Coraima Orochena

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The Smart Tabletop Gaming (STG) Device is a hybrid application of an XY-plotter CNC device and a web application that bridges players from offsite locations onto a physical tabletop with other players. This system is a server-client system, in which multiple clients are able to interface with the physical device and able to communicate their actions effectively via the web app. This web app can translate points of interest from a player's perspective, communicate movement of game pieces from web app to the tabletop, and keep track of basic data. This data comprises of game pieces statistics, a chat window for connected clients, a user account system.**

I. Introduction

When it comes to Tabletop games, one thing that keeps people from trying out this genre of gaming is they want to play with a group they are familiar with. Another would be that one or more people would like to join into an ongoing session, but they are in faraway locations. Also, most people would want to have a tangible feeling to their game and do not like to use a virtual tabletop. This project could provide a solution to a variety of people with different needs.

The Smart Tabletop Gaming (STG) device utilizes a web application that communicates to the main tabletop board device. It comprises of a server-client system in which the server contains all the persistent data that is occurring on the board and updates all the client applications that are connected to the server. The main board is the central point where all events occur, so any changes that happen on the board overrides any relevant data that is stored on the server and database.

The physical board is comprised of a 10" x 13" grid. This limitation is in place due to the x-y plotter area constraints. This plotter device is controlled by the AVR Atmega2560 chip, utilizing the Arduino environment. This chip is embedded into a custom 2-layer PCB, which contains components to sense changes to the board and output various light effects through Adafruit WS2812B individually addressable LEDs to emulate player actions and decisions. This board utilizes the ESP8266-12E WIFI module, to connect to a local area network and transmits/receives data from our web server.

There will also be a sensor system in the project to detect where a game piece is located and when it's moved. Sensing changes on the board and moving the game objects without external actions is an important feature of our project. For both to work, the tabletop device utilizes magnets on the bottom of the game pieces. These game pieces are detected by our Reed switch array, located under the main board. To move the game pieces, the xy-plotter has a mounted electromagnet on the translating tip of the device, and orientates itself under the game piece of interest, and activates to firmly attach itself to the movable object. To path this object without colliding with game pieces, we utilize the A* pathing algorithm to create a proper path from start to finish.

This system of components all works together to create a dynamic tabletop board that can be controlled by exogenous actions and via the web application, while receiving feedback on live changes to the board.

II. System Overview

The Overall system comprises of different components working together. The current state of the tabletop is continuously being checked to assure that the web application is synchronized with the state of the board. The state block diagram is shown in figure 1, which shows the different states the tabletop application will go through.

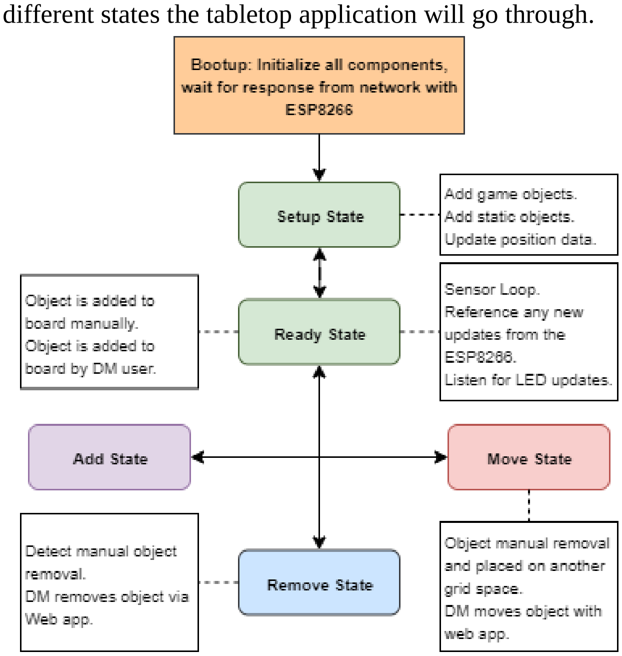

Fig. 1. Game State diagram

As shown in the diagram, our main controller boots up, and waits for a proper connection with a wifi network, once this is successful, it goes into the setup state, in which game objects will be added to the board and grid spaces can be modified into static objects. These static objects can represent walls or objects that are considered impassible. The different game states are set depending on the event that is occurring on the tabletop or web app.

III. Hardware

In this following section, the hardware implementation is using a fully constructed XY plotter and the following devices which includes the Arduino MEGA 2560, ESP8266 and the A4988 stepper driver controllers to make the stepper motors move the game piece in a Full-Step motion. By doing so, the game piece would move in X and Y motion to the designated location when receiving commands via web API application by phone. The person or player playing in this tabletop gaming will select a point on the phone touch screen which then will be transmitted through a connection from ESP8266-12E to the ATMEGA 2560 chip and finally to the A4988 stepper driver controllers. These stepper driver controllers will then activate the stepper motors for movement.

These NEMA 17 stepper motors are already assembled onto the XY plotter. The holder will have an electromagnet with enough force to attract a game piece with a neodymium magnet located underneath a game piece. The electromagnet can be activated when it is appropriate to do so. These stepper motors will move the holder just underneath the game piece which then after fulfilling this task, the electromagnet will activate to attract the magnet piece and smoothly drag to the designated location.

During the movement of the game piece, there is a sensor matrix beneath the board made up of reed switches and diodes. Shift registers will be used for the sensor matrix to increase the number of outputs. There will also be LEDs underneath each block for color and visual special effects.

A. IDRAW Machine XY plotter

Most XY plotters are commonly used as printers to able to receive data from the user and able to draw a figure as the output. The XY plotters consists two stepper motors for X and Y motion respectively. For these stepper motors, they can be configured to different steps of motion. These stepper motors must be configured in Full-Step motion for the game piece to move smoothly.

Since this type of XY plotter is already assembled with the NEMA stepper motors and is a viable candidate to satisfy the working area, it is best to create a PCB and replace the embedded circuit board and use the connections of the stepper motors.

Fig. 2. IDRAW Machine XY plotter Physical presentation

B. A4988 stepper driver controllers

In order to maintain the right amount of current being delivered to the stepper motors with efficiency, the A4988 stepper driver controller with the adjustable potentiometer will embedded onto the PCB design. This can help adjust the right amount of current to the NEMA 17 stepper motors. During the testing and prototyping, the digital multimeter was used to measure the current and this helps determine the resistance of potentiometer and adjust the potentiometer accordingly.

The number of steps and direction of the stepper motors was then determined by the web libraries compatible with the Arduino MEGA 2560. From there, the code was sent to A4988 stepper driver controllers to their respective DIR pin and STEP pin.

C. Electromagnet

The electromagnet is essential during hardware configuration. It is connected to a transistor called TIP-120 which acts like a switch and can withstand a high current during normal operations. The Arduino MEGA 2560 will indicate to the electromagnet when and where to deactivate its magnetism to the magnetic piece underneath the game piece.

Once the user sends instructions to move their game piece to another location on the 10”x13” grid, the electromagnet holder will move to where the game piece is located and activate its electromagnet by the Arduino

MEGA 2560. From there, the game piece will be attracted to the electromagnet. Then, the game piece will be dragged across the pixel glass gameboard to its designated location. After fulfilling the task, the electromagnet will deactivate its magnetism and remain still until the user instruct to move his or her game piece.

D. Magnets

The magnets that were chosen to be put under each game piece must be very strong to make sure the electromagnet is able to detect them and move them easily.

Fig. 3. A rounded neodymium magnet (18mm x 3mm) to place under the game piece

For this purpose, circular flat neodymium magnets were chosen. Multiple, both neodymium and ceramic magnets were bought and tested with the electromagnets and reed switches to see which one would be most compatible and work best for movement and distancing between the electromagnet, reed switch matrix, and game piece with the neodymium magnet underneath it. The one that was chosen for the project is the neodymium magnet that is 18mm x 3mm as shown in the picture above.

E. LEDs

The LEDs being used are WS2812B individually addressable LEDs that can be programmed with the MEGA2560. These LEDs are a perfect to be placed under each grid. The purpose of these LED's is to add visual effects during game play. The grid will light up when the game piece is on that same grid and the grid will light up the path where a character needs to travel. The LEDs will be placed right under the senor matrix. All LED connections are all in one strip since it is easier to implement.

F. Reed Switch Matrix

For the sensor matrix, reed switches were chosen, they are magnetic sensors that get turned on when a magnet is placed over them, when the magnet is removed, they turn off. This is applicable to the project since the game piece will have a magnet underneath it which is what's needed to trigger the reed switches. The matrix will be placed right below the game board.

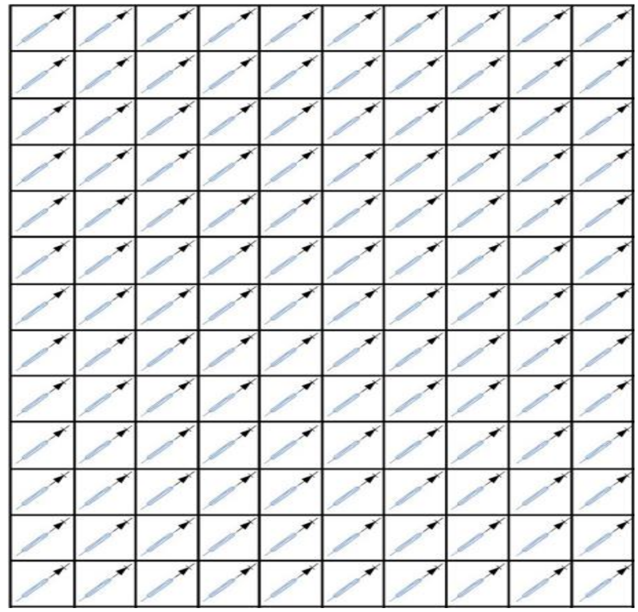

Fig. 4. Reed-Switch matrix conceptual grid 13 in. by 10 in. presentation.

G. Shift Registers

The shift registers that are being used are 74HC595 and 74HC165. Two of each will be used and they are connected in a daisy chain which means that these two devices are connected in series. The reason of this implementation is because there are not enough I/O pins in the MEGA2560 to handle 13 outputs that are coming from the sensor matrix.

H. PCB Design

The PCB schematic consists of using the important components such as ATMEGA 2560 surface mount electronic chip, ESP8266-12E Wifi module, A4988 stepper drivers which would be the connected to NEMA 17 stepper and the shift registers two 74HC165's and two 74HC595's and the connection with the TIP120 transistor with the electromagnet. Most of the passive elements such as ceramic capacitors 0.1uF and 47uF and resistor arrays valued of 10Kohm and 2Kohm are maintained onto the schematic.

To avoid overheating, the heatsink would be placed onto the power regulator all electronic operations are going to take place. An CPU fan can also be included to regulate the operating temperature of the PCB board. Two power jacks are placed on the corner of the PCB board. The first power jack is connected to the ATMEGA 2560 chip, the Wifi module, and the shift registers with 5V. The second one is to power 12 volts to the A4988 stepper drivers connected to the stepper motors.

IV. SOFTWARE

The overall software design is sectioned between the web application, database and microcontroller application. The web application is synchronized with the current state of the database and two controls within the system which are the web app and controller app. Each is in charge of loading the database and since there is multiple users performing different actions at various time, a non-blocking application will be implemented which is key to reduce possible errors between the applications.

The initial design will make the application responsive and be able to receive multiple actions with minimal error margin. These actions can happen at any time and the only action set to be synchronous is the game piece movement.

In most tabletop games, in order to move the pieces around the board it expects a defined two-part action of lifting a game piece and placing it down at a different location. Any movement action within both applications will be expecting a complete movement action. Other game actions such as adding a new game piece, pinging the map can be done asynchronously.

A. Wifi Communication

In order to communicate between the microcontroller and the devices on the board, the Wifi communication is needed. The Wifi communication will be established via school server in order to initiate the project. It is vital to establish this connection an hour before the final presentation to troubleshoot any web violations. The wifi connection will commence using the ESP8266-12E which has wifi capabilities and WebSocket, a computer communication protocol, that provides full-duplex communication channels over a single TCP connection.

This creates a persistent connection between a client and a server that both parties can use to start sending data at any time. The client establishes a WebSocket connection through a process known as the WebSocket shake, then the client sends a regular HTTP request to the server. An upgrade header is included in this request that informs the server that the client wishes to establish a WebSocket connection.

B. Web Server and Database

NodeJS is the chosen webserver to host our API for our web application. The decision was made after analyzing the functionality of using an apache-based web server and its capability in solving the issues involved in this project. Its back-end capability can support a dynamic and modular web application using also libraries and API's. For the Database, MongoDB will be used. It is an object-oriented, simple dynamic and scalable NoSQL database. This Database was chosen due to the benefit of using JavaScript as the programming language for the backend and, for the most part, the frontend as well.

C. User Interface

The Web application user interface will allow all users to interact with the board and receive feedback. It displays the functionality in such a way that it can be easily understood and performed. It uses a hub to receive all changes that occur on the main tabletop application.

V. SYSTEM DESIGN

A. Full-Step Motion Configuration

In using the A4988 stepper driver controllers, the potentiometer configures the types of motion that each stepper motor can do. By twisting the potentiometer, it allows the driver to allow some current into the stepper motors which can affect the motion. The operating current for the stepper motors is 0.5A in order for the motors to perform a full step motion. The constant voltage operating is 12V and by adjusting the potentiometer to increase or decrease the resistance to have an operating current to be 0.5A.

By using Arduino web libraries, the code was already set to put the number of the steps desired during a time interval. The working area is constructed into a fixed number of grids consisting roughly 1 in. for length and width. The number of steps implemented in the code will the determine that desired length and width of the grid which resulted to be 120 steps. This reference is used to avoid any conflicts with the other game pieces during normal operations. The equal spacing prevents the game piece to be magnetize to another grid. The game piece should be magnetized to one grid at a time after the reed-switch is activated during a normal operation.

B. Electromagnet Configuration

The electromagnet is connected to the Arduino MEGA 2560 with the connection of the TIP-120 transistor. The TIP-120 can withstand high current including the operating current 1A which prevents overheating. The coding script consists of setting up the pins of the Arduino 2560 to "output" pins in order to configure to turn on and turn off its magnetism.

Also, the Arduino MEGA 2560 can set up a time interval of when to turn on and off its magnetism. The electromagnet will be mounted onto the holder of the XY plotter. This will become the center piece for the movement of the game pieces.

C.ESP8266-12E Configuration

The ESP8266-12E is a wifi module that is responsible for online communication between the user and the board of the XY plotter. Since the operating voltage for the Arduino Mega 2560 is 5V and the module requires 3V, a bi-directional logic level converter is needed. This component is required because the module will be receiving and sending data to the Arduino Mega 2560.

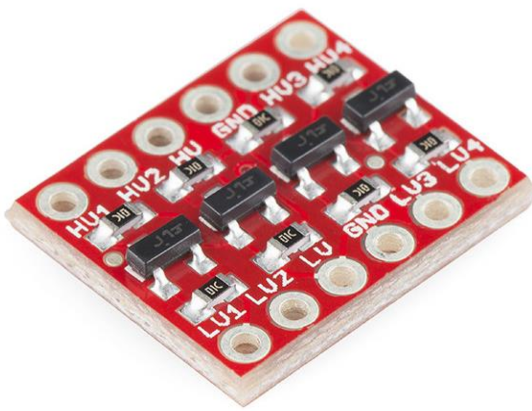

Fig. 5. A 5V to 3V Bi-Directional Logic Level Converter

The previous method was implementing a voltage divider for cost-effective purposes, however, sending data from the ESP8266/12E to Arduino Mega 2560 during normal operations since sending the data in 3V to 5V is not possible. The HV pins will be connected to Arduino Mega 2560 respective pins along with a 5V pin. For LV pins, the ESP8266/12E will be connected to its respective pins with a 3V pin.

D. Reed Switch Matrix configuration

For the construction of the sensor matrix 130 KSK-1A66-2030 reed switches and 1N4148 diodes were soldered on to each other to have them connected in series. These type of reed switches were used because they are very small reed switches that are measured to be 14mm x 2.2mm, very small reed switches are necessary because they have to be under each grid and each grid is only an inch in size. Since the matrix is 10x13 and there are many reed switches all next to each other in every direction the 1N4148 diodes are needed to prevent "ghosting" from happening.

"Ghosting" happens when the MEGA 2560 scans the matrix and it mistakes some of the reed switches for being activated when they actually aren't. An example is a game piece (with a magnet under it) on a specific grid activating the reed switch beneath it, the diodes make sure that the reed switches around the game piece don't get activated by not letting the current flow back to any of the reed switches around it.

As shown in the picture below the way the reed switches are set up in a standing position in a 45 degree angle. They are set up like this because for the neodymium magnet under the game piece to activate the reed switch the reed switches have to be in that position, that was figured out from the various ways that it was tested before the matrix was assembled.

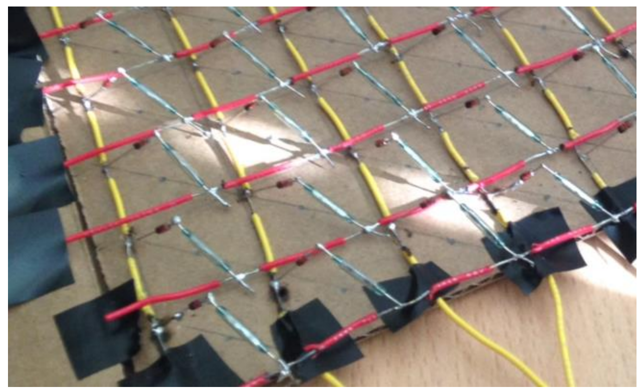

Fig. 6. Reed-switch matrix during construction

E. Shift Register Configuration

After finishing the Reed-Switch matrix configuration, the next step to connect the shift registers in daisy chain. For this type of connection to happen, the 74HC595 will be connected in series with another 74HC595 shift register. The 'DATA' pin or 'Serial In' Pin from the first shift register will be connected to the 'MOSI' pin from the ATMEGA 2560 chip.

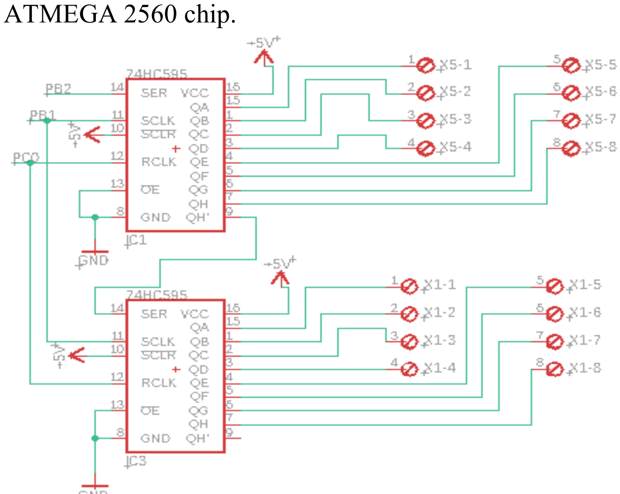

Fig. 7. Two 74HC595 shift registers connected in series

From the first shift register, the QH' pin will be connected to the 'DATA' pin of the second 74HC595 shift register. The 'SCLK' and the 'RCLK' pins will be connected to each other. The output pins from those two shift registers will be connected to male headers, so that the connections will be secured when placing the wires from the reed-switch matrix.

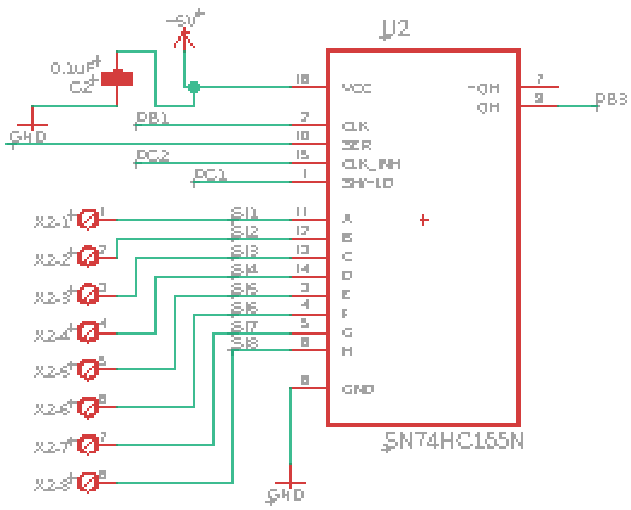

Fig. 8. Two 74HC165 shift registers connected in series

The connection of the 74HC165 shift registers are also being considered. Two 74HC165 shift registers are required to be connected in series as well. In this case, the 'DATA' pin of the first shift register connects to QH' pin of the second shift register hence there is a daisy-chain connection. All the 'CLOCK' and 'LATCH' pin connects to the respective pins of the ATMEGA 2560 chip.

F. LED Configuration

The LEDs were the simplest parts to set up out of all the other aspects of this project. As mentioned before, they are WS2812B LEDs that were chosen because they are individually programmable and compatible with the MEGA 2560. Their purpose is to add visual and special effects to the game board. To achieve this, they will be programmed using Adafruit Neopixel libraries.

The way they are physically set up is for them to be placed under each grid of the board right under the sensor matrix. A total of 176 LEDs is needed just like the reed switches for the sensor matrix, so there is one under each grid. To make sure they are exactly under each grid, LEDs that are a bit more than an inch apart were bought and folded to be able to space out each LED as desired.

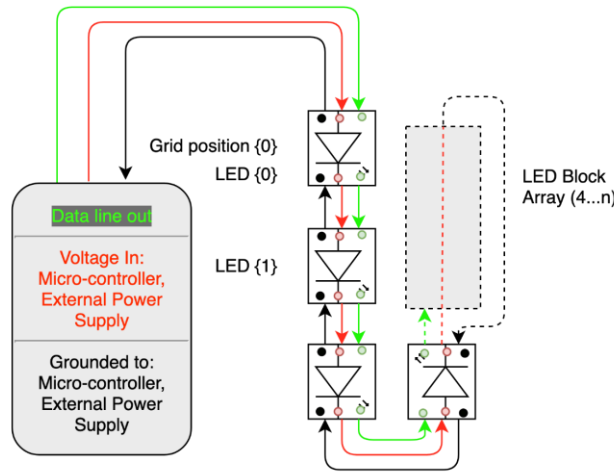

Fig. 9. LED configuration diagram located underneath the gaming platform

To power the LED's independently from the MEGA 2560 to not cause overheating LED drivers were bought, which is just a power supply for the LEDs. As shown in the diagram below this is how it's connected to the MEGA 2560 and to the LED driver and shows how it functions. It is connected to the ground pin of the MEGA 2560 and the data pin also, then to power it, it is connected to the external power supply which is the LED driver.

VI. Software Details

The operation of our system components is driven by well-defined architectures and open source libraries. Our front-end design on the web application is built on the latest version of Angular, which communicates with our Node JS webserver. This web server is the main hub between all connected web clients, including our tabletop device. Communication is done via Wifi on a local area network, and the Esp8266 interfaces with the webserver via this network. From this point, bi-direction communication is integrated between the Esp8266 and the Atmega2560 via the SPI protocol, which data is transferred in parallel. The Atmega2560 is the Master device, which also communicates to the sensors via SPI as well.

A. Front-End Angular Design

Angular is a Component and module-based system, with Typescript being the designated programming language. The design of the front-end is separated into modules on major feature implementation, while individual components build up the various subsystems of the module.

Helper classes were made to do small tasks, while services drive the data manipulation and communication protocol.

To improve the formatting and visual design of the user interface, we included Angular Materials and Bootstrap libraries to improve our visual design, maintain User friendliness, and keep the responsiveness of the web pages across different devices. The module and component design are displayed within figure 8.

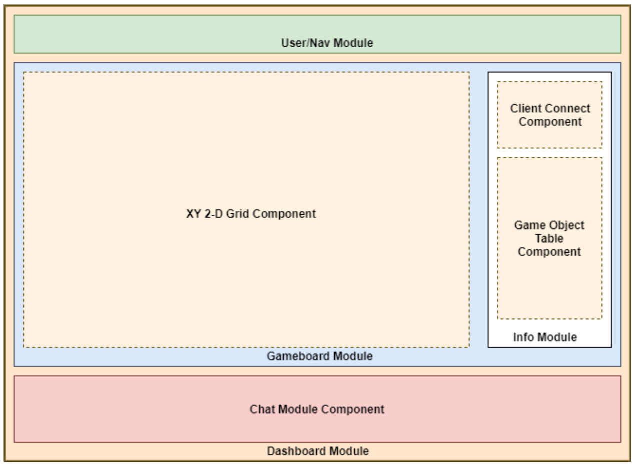

Fig. 10. Front end Dashboard page

Figure 8 is the main dashboard in which a user will perform most of the game actions. When the user first arrives on the web application, they are directed to the landing page component, which gives the user the option to log in.

Users are designated at two separate types, as a player or as a Dungeon Master (DM). The player is only able to control game objects that they are designated to control, while a DM can control all objects that are displayed on the board. Other actions the DM is able to include is to create impassible grid spots to represent walls or doors.

B. Back-End Node JS Server Design

NodeJS is an application runtime environment that allows you to write server-side applications also written in JavaScript. The setup of NodeJS is done by writing event handlers that get called when certain Node events happen. To initialize the application, the node package manager, NPM, was installed in order to create a json package which contains all the libraries relevant to this project.

To authenticate all users who create and account and login into our server, we included passport.js which is a authentication middleware especially for NodeJS. As our front-end is being utilized client side, any relevant data flows through the Node JS backend to synchronize all the connected clients to the server.

C. Front-End and Back-End Communication

In order to communicate Angular frontend and NodeJS backend, we used HTTP requests to our server-rendered app. This is done by having the browser send HTTP request and the server replies with a HTML page.

Once the user clicks in any feature of our tabletop web application, the Front-tend makes a call (GET or POST) request in response to an event. Then the backend receives this call an takes that information to route the request appropriately.

D. Software Test Plan

The purpose of a software test plan in our project is to make sure the web application and tabletop software development procedure is done efficiently as well as to save time when debugging. A functionality test was used in order to check the product implemented work as intended, this was done by feeding input to functions and then inspecting the output. Small pieces of code were first selected in order to check the output of each before merging. Once each worked as intended, an integration test was done in order to expose faults in the interaction between integrated units.

C. Network Communication

In order to allow the applications to communicate with each other, a local are network connection will be implemented via a Wifi hot spot and implemented before the demonstration begins. An assigned laptop will be brought and used as to create this wireless hotspot to connect our device.

The device communicates using socket io libraries with the device acting as a client. There is not NODEJS server which pushes to the connected web application clients. The DM client oversees all data from the server and has privileges to add challenges and objects to the board such as creating walls and pushing this object data to the server and syncing all clients including the tabletop device.

D. Tabletop Application Game Design

The tabletop device is the master device controller between all the connected components. The SPI protocol is utilized to send data between the microcontroller and the shift registers in our project. Separate clock and data lines along with a select line are used to choose the device that we wish to talk to.

When the player chooses a location to move to, the microcontroller created will send programming algorithms for sensing communication which then communicates to the XY plotter to choose a game piece and move from a location to another location. The microcontroller will also display a visible interface which indicates health and armor level of the character created. Once this is completed, the interface will store and modify data to upload certain statistics after a battle has been finished or some certain action during gameplay.

VI. Conclusion

During the gameplay of tabletop gaming, the players are not going to use their hands to move the game pieces. Instead, they will communicate via phone application using the Wifi-module ESP8266-12E that connects ATMEGA microcontroller chip through the logic level converter. The movement will take place on the XY plotter grid by implementing communication with the A4988 stepper drivers to the stepper motors. The LEDs will be added for visual effects and will be located underneath the gameboard.

The front-end design on the web application is built on the latest version of Angular, which communicates with our NodeJS webserver. To integrate both, HTTP requests were used. Users are designated at two separate types, as a player or as a Dungeon Master. The player is only able to control game objects that they are designated to control, while a DM can control all objects that are displayed on the board. Other actions that the Dungeon Master is able to do is to include is to create impassible grid spots to represent walls or doors.

Once a complete integration of the hardware and software is finished. The player will be able to enjoy a tabletop game based on the Dungeon & Dragons game that will allow the player to either play in person with other users or from long distance via online.

Biography

Richard Almario is a 27-year old graduating Computer Engineering student. He is an intern with the software company, Ace applications, LLC, and is working towards creating solutions on various website and mobile applications.

Coraima Orochena is a 23-year old Electrical Engineering student. She is currently an intern at Athena-Tek, a networks operation firm, and was previously an intern at NFI Industries, a supply chain solutions company.

Jose Castano is a 24-year old Electrical Engineering student that is passionate in studying further in electronics. His main goal is to obtain a master's degree in the field of Electronics.

Hubert Barrantes is 27-year old Computer Engineering student who is self-employed. His interest is to obtain a master's degree in the field of embedded systems, coding, and management.

References

[1] Ziccardi, Chad. "Magnetic Sensing Chessboard." *mtifall10*, 11 Dec. 2010, mtifall10.wordpress.com/2010/12/10/magnetic-sensing-chessboard/.

[2] "In-Depth: How 74HC595 Shift Register Works & Interface with Arduino." *Last Minute Engineers*, Last Minute Engineers, 7 Dec. 2018, lastminuteengineers.com/74hc595-shift-register-arduino-tutorial/.

[3] "How to Build a Register Circuit with a 74HC165 Chip." How to Build a Register Circuit with a 74HC165 Chip, www.learningaboutelectronics.com/Articles/74HC165-register-circuit.php.

[4] Nedelkovski, Dejan. "How To Control WS2812B Individually Addressable LEDs Using Arduino." HowToMechatronics, 4 Sept. 2019, howtomechatronics.com/tutorials/arduino/how-to-control-ws2812b-individually-addressable-leds-using-arduino/.

[5] "High-Power Control: Arduino + N-Channel MOSFET." Bildr, 5 Mar. 2012, bildr.org/2012/03/rfp30n06le-arduino