Functional Specification

Year: 2021 Semester: Fall Team: 8

Project: Sink or be Sunk

Creation Date: August 30, 2021 Last Modified: September 3, 2021

**Member 1:** Mitchell Arndt **Email**: [arndt20@purdue.edu](mailto:arndt20@purdue.edu)

**Member 2:** Garrett Brillhart **Email:** [gbrillha@purdue.edu](mailto:gbrillha@purdue.edu)

**Member 3:** Joe Mislansky **Email**: [jmislans@purdue.edu](mailto:jmislans@purdue.edu)

**Member 4:** Molly Arito **Email**: [marito@purdue.edu](mailto:marito@purdue.edu)

Assignment Evaluation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Score (0-5)** | **Weight** | **Points** | **Notes** |
| **Assignment-Specific Items** | | | | |
| **Functional Description** |  | x3 |  |  |
| **Theory of Operation** |  | x3 |  |  |
| **Expected Usage Case** |  | x3 |  |  |
| **Design Constraints** |  | x3 |  |  |
| **Writing-Specific Items** | | | | |
| **Spelling and Grammar** |  | x2 |  |  |
| **Formatting and Citations** |  | x1 |  |  |
| **Figures and Graphs** |  | x2 |  |  |
| **Technical Writing Style** |  | x3 |  |  |
| **Total Score** |  | | |  |

5: Excellent 4: Good 3: Acceptable 2: Poor 1: Very Poor 0: Not attempted

General Comments:

1.0 Functional Description

The Sink or be Sunk game is a battery powered and electronic-enriched board game that allows for a unique user experience in both single and multiplayer modes. It uses an LED array to display attacks, hits, and misses from yourself and the other player as well as giving audio cues from an onboard speaker. Boats are automatically detected once inserted using a resistor network specific for each type of boat to allow for ease of play. Two boards communicate with each other over WiFi, easily connected using a phone or laptop, with LEDs giving indication to connection to WiFi and to another board. Simply search for the Sink or be Sunk WiFi network, connect to it, and then follow the on-screen instructions that pop-up from the webpage.

2.0 Theory of Operation

The Sink or be Sunk game system relies on a WiFi connection to communicate with the main server (from now on referred to as the Game Server). This connection needs to continuously update both the player and the Game Server about various game events such as when a player makes a move, the opposing player makes a move, the game starts/ends. All these events must be coordinated between all parties: the player, the opponent, and the Game Server. To accomplish these synchronous actions the Game Server will deploy web sockets between all connections. These sockets are open connections between client and server that allow for bidirectional communication. Unlike a REST API which requires continuous polling for updates, this socket remains open for the duration of the player's game.

The Game Server is also responsible for most of the game code. This means that the server will handle the logic involved with recognizing valid player moves, keeping track of the current player turn, and determining when boats are hit or missed during gameplay. The server is implemented as an express server in the NodeJS framework. To maintain strong typing for ease of development and debugging, the language of choice will be typescript. Additionally, a MongoDB database will be implemented on the back end to hold player information such as their linked game console id, player history, and other profile information. To integrate into the server code, an object data modeling (ODM) library called mongoose is used.

Before the signals can be sent to the server, they need to be registered by the local microcontroller of the project (ESP32). The main user inputs include the indication of an attack move, the indication to start/end a game, the positioning of ships. The first two cases will be handled via a 16 key button matrix (buttons A-H for the rows and 1-8 for the columns). These buttons will be arranged into a grid where the row and columns will be scanned in a typical fashion to accept a single keypress at a time.

The more interesting inputs come from the positioning of the ships. Each ship will have two “pegs” that will insert into the board. Each “peg” will consist of four conductive legs. These legs are connected in pairs to form a resistor divider once placed into the board. One pair of legs is one resistor. When the legs are inserted into the board, two of the four holes are connected to Vdd and GND respectively forming the divider. The other two holes will be shorted together and represent the middle node of the divider where an analog voltage will be read. Each ship will have a unique set of resistors leading to a unique voltage created from the divider. Each of the 64 (8x8 grid) “pegs” will be multiplexed down to a single channel to be read in from the ESP32’s ADC. The microcontroller will then have logic to decipher the ship “peg” type and location or additionally if there is no ship there at all. The detection of no ship will relate to a voltage of GND due to an additional pulldown resistor at the input of the ADC to account for floating inputs of empty peg holes. (See Appendix 2)

The final piece of this project is to create an enjoyable user experience through audio, visual, and physical stimuli. This will be accomplished through sound effects from a built-in speaker, lighting from tri-colored LEDs, and shake from a rumble motor. The audio will be driven from the DAC of the microcontroller passed through an amplifier. The amplifier will have adjustable gain from a potentiometer knob on the game console. The LEDs will be addressable such that they can all be controlled via a single wire protocol that is daisy-chained throughout all LEDs on the board. The rumble motor will be controlled via a MOSFET driver controller via the microcontroller’s GPIO output.

3.0 Expected Usage Case

Sink or be Sunk is expected to be used wherever a WiFi connection is possible. Its portability allows the game to be played from a phone hotspot, so options are close to limitless. If the electronics and the device can be dry and in temperature, players should be all set to go. Players can play against an AI, or against another player across the world. Players are expected to have basic technical literacy, as setting up the WiFi connection requires some knowledge of networks, although the User-Guide will make the process easy. The age range can wildly vary, as young kids would love to play against their friends and older adults would love the nostalgia of the classic game.

4.0 Design Constraints

4.1 Computational Constraints

The primary computational functions of the Sink or be Sunk board game are focused on transferring small amounts of data between boards over WiFi, detecting boats, controlling game flow on the AWS server, and managing the LED matrix and sound for the user. The 128 LEDs will have to be blinked fast enough to give the illusion of persistent light to the human eye, nothing lower than 100 Hz. The detection of boats and attacks can be relatively slow, but not less than 1 second as it is doubtful the user will be able to plug more than one piece in a second. The microcontroller will have to take some time to do this detection, as it has 128 coordinates to investigate the I/O of, but 1 second should be more than sufficient. The information sent over WiFi will be sent immediately, to avoid unnecessary lag between users. The ESP32 can more than handle the WiFi aspect, as the payloads will be small and infrequent. The bulk of the computation to be handled by the ESP32 revolves around the maintenance of the LEDs PWM and detection of attacks/boats. Utilization of a shift register to write in the LED data would reduce the load of the microcontroller to only write the data only when the data needs to be changed, removing the necessity to continually monitor the LED array. Since the game flow will be handled on the server, the heavy computations and algorithms of game flow will be irrelevant to the microcontroller. The “Game-Flow” includes managing turns between players, start/end games, searching for matches in a lobby, connecting boards together to play, determining valid moves and hits/misses, and being the AI for single player. Memory-wise, the microcontroller needs to maintain the states of the 128 LED array, audio files, and other user experience I/O. Flash memory should suffice for the LEDs and inputs, although audio files may need additional memory.

4.2 Electronics Constraints

The Sink or be Sunk game console will use the following subsystems: LCD display, rumble motor, LED strips, analog ship detection multiplexers, speaker, WiFi connectivity, push buttons. The LCD display will be driven via a standard SPI protocol [4]. The rumble motor will be driven by a MOSFET controlled via a GPIO digital output. The LED strips can be controlled via a single wire via a standard LED protocol [5]. Multiplexers will require a certain number of select lines based on the reduction ratio the team chooses controlled via GPIO. This approach is shown in the diagram in Appendix 2. Additionally, the multiplexers will terminate into analog pins of the microcontroller which requires at least one internal ADC. The ADC will nominally be pulled low via a large impedance resistor to prevent the input from floating. The speaker will require the addition of a DAC feeding an amplifier on the console. The microcontroller will also need a WiFi module for wirelessly connecting to the internet. Finally, there are various push buttons in the design for user inputs. These buttons can be arranged in a matrix and polled via GPIO inputs using a standard approach [6].

4.3 Thermal/Power Constraints

The main constraint deals with the power consumption of the batteries. We target a usage of several hours (3-5) before needing fresh batteries. As the device needs to power LEDs, microcontroller, speaker, and ESP32 this seems reasonable for disposable batteries. We expect to utilize two AA batteries of 1.2 - 1.5 V each with 2 Ah. Thermal constraints are low, as our computational workload is low, and the LEDs should not produce much heat. However, the temperature at the component level should not exceed 80o C to protect components and temperature outside the packaging should be cool enough to be comfortable on the user’s lap.

4.4 Mechanical Constraints

The mechanical constraints all revolve around the device being portable. The device should be no larger than a 1.5 x 1.5 x 0.5 ft box when closed and weigh no more than 10 lbs. The device also needs to be able to hinge closed and open, for playing and storage. The closing of the lid also has the benefit of protecting the electronics and reducing size. The package needs to be durable, as the device needs to be able to survive the basic shocks and bumps of travel. Durability must include the ability to be tossed into a backpack and worn around for a day without any damage as this is a typical usage case. This device is intended to be played in favorable weather conditions, so weatherproofing is not necessary. The package must also be RF permeable, so that WiFi communication can occur.

4.5 Economic Constraints

There currently exists an upgraded version of Hasbro’s Battleship called Electronic Battleship. This version still requires two users to be in close quarters to play the game. This also still requires putting in pegs that are red or white, based upon if your opponent tells you that it is a hit or miss. The listing price for this game on Amazon [1] is $31.49. However, our version of the game allows users to play alone against a computer or with someone anywhere else on the internet. Other board games that connect to the internet through WiFi cost anywhere from $20 to $90 [2]. If this product was being mass-produced, it would be targeted to cost around $100 since this is an electronic game for children. This cost includes the price of two boards so that two people can play. Due to this being a singularly produced final prototype, the target is to spend around $250.

4.6 Other Constraints

Sink or be Sunk has optional single-player and multiplayer modes, however both require an internet connection. Since the game code is located on an online server, players are required to have WiFi access to enjoy gameplay.

Sink or be Sunk is designed to cater to all ages; it is critical that the player interfacing is engaging while maintaining the low-level complexity. The players should be able to navigate through WiFi and game setup, boat placement, and coordinate selection without hassle. Additionally, the rules of gameplay must be straight forward and consistent for first-time players.

Adding an electrical component to the typical board game can be enticing to players, however it is critical to ensure any additional LEDs are not causing harm to players’ eyes. As previously mentioned, Sink or be Sunk is aimed at people of all ages, demographics, and backgrounds; bright, blinking lights can pose health hazards. According to the Assil Eye Institute of Los Angeles, “The ANSES study warned that children and teenagers, whose crystalline lenses aren’t fully formed, have eyes that do not fully filter blue light, making them particularly susceptible to its harmful effects” [3]. Thus, it is essential to construct the boards in such a way that the LEDs are visible, without being overly bright.

5.0 Sources Cited:

# [1] Amazon. “Electronic Battleship Game.” Amazon.com. https://www.amazon.com/Hasbro-A3846-Electronic-Battleship-Game/dp/B00CPGZ9E4/r (accessed Sept 1, 2021).

[2] N. Summers. “The best board games with an app-based twist." Endgadget. https://www.engadget.com/2019-11-29-best-board-games-companion-app-guide.html (accessed Sept 1, 2021).

[3] A. S. Gaur, “Led lights can harm your eyes,” *Assil Eye Institute Blog,*

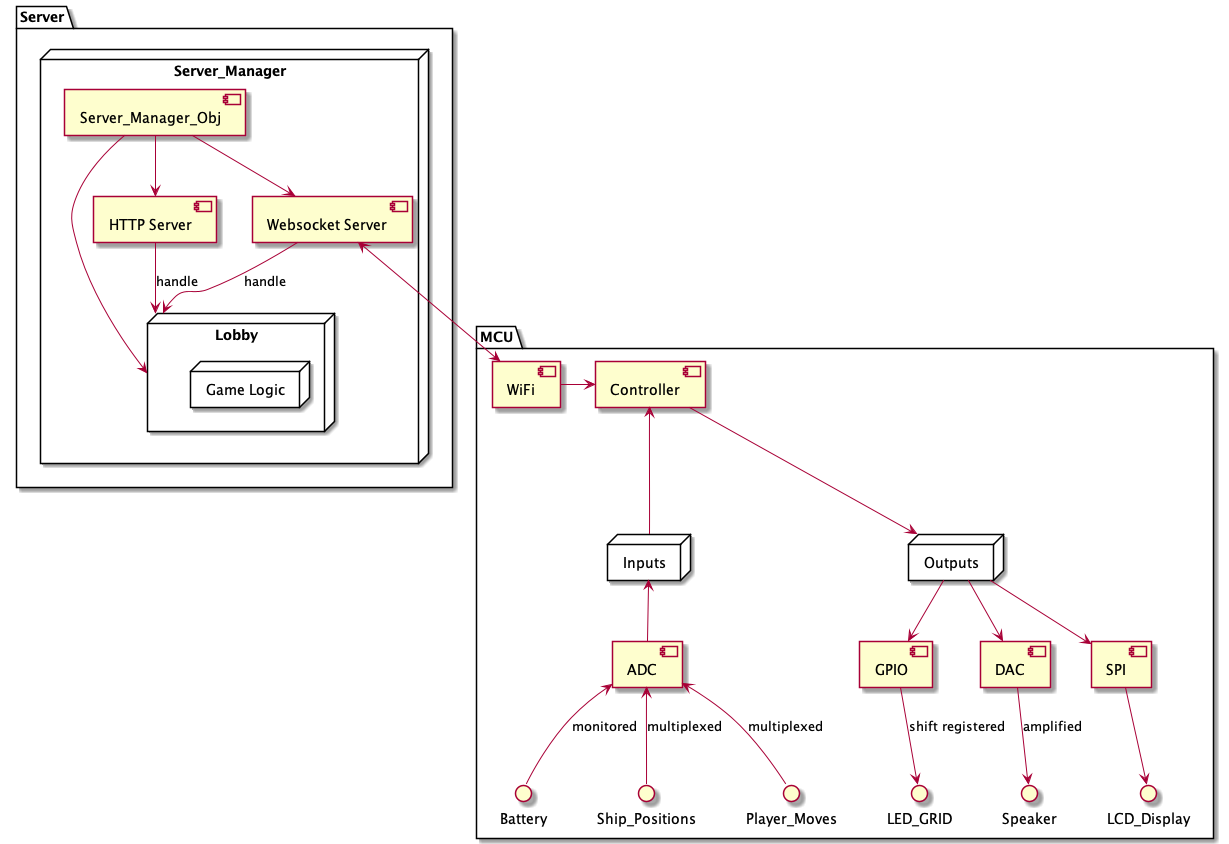
26-March-2021. [Online]. Available: [https://assileye.com/blog/led-lights-can-harm-your-](https://assileye.com/blog/led-lights-can-harm-your-eyes/) eyes/. [Accessed: 02-Sep-2021].

[4] SparkFun Electronics. (n.d.). *SparkFun 16x2 SERLCD - RGB backlight (Qwiic)*. LCD-16396 - SparkFun Electronics. <https://www.sparkfun.com/products/16396>.

[5] Cook, J. S. S. (2020, March 6). *WS2812 tutorial: Protocol for the WS2812B programmable led*. Arrow.com. <https://www.arrow.com/en/research-and-events/articles/protocol-for-the-ws2812b-programmable-led>.

[6] *Introduction to microcontrollers - button matrix & auto repeating*. EmbeddedRelated.com | Embedded Systems. (n.d.). <https://www.embeddedrelated.com/showarticle/519.php>.

**Appendix 1: Functional Block Diagram**



**Appendix 2: Boat Detection Using Voltage Dividers**

