WordPal

Group 36

Members

Hilda Deveaux
Computer Engineering

Daniel Ortiz-Gratacos
Computer Engineering

Nala Rivera
Computer Engineering

Review Committee

Dr. Sonali Das, ECE Department
Dr. Mark Maddox, ECE Department
Dr. John Aedo, CS Department

Advisor

Sreeram Sundaresh

1.1 Project Background and Motivation

Wordle is a game played daily by millions of people in all walks of life. It is simple enough to be completed in under 5 minutes, yet complex enough that it never gets boring. The game of Wordle is heavily inspired by the game of Mastermind (or “Bulls and Cows”), where the objective is to guess a sequence of 4 colored pegs in a limited number of opportunities. The key to the game is the feedback system; with each guess, the player is told the number of pegs they have in the correct position (communicated with smaller red pegs) and the number of pegs that are the same color as one present in the correct sequence, but are in the incorrect position (smaller white pegs). Wordle expands on this concept by replacing the sequence of colors with real 5-letter English words. As opposed to Mastermind, where the player is not told which specific pegs are correct or present (only the count), Wordle tells the player which letters are in the right or wrong position by displaying a green or yellow background behind them respectively (absent letters are marked with dark gray).

Our project idea aims to bring Wordle to life in a unique way by turning the keyboard keys themselves into the feedback system. The native Wordle application uses its own on-screen keyboard (notably doing so regardless of whether you have a different physical or on-screen keyboard handy), which it uses as an auxiliary way of communicating feedback to the player, by changing the background of each key to reflect the status of each letter. In our project, we plan to implement a physical keyboard, a la the Blackberry or other mid-late 2000s mobile phone keyboards. However, we will make our product stand out by using semi-translucent keys to color in the background of each letter analogously to the Wordle app as presented by The New York Times and Josh Wardle.

Additionally, a major aspect of Wordle is that it is only playable once per day; at midnight local time, the Wordle resets and everyone in the world is given the same new word to guess. We plan to implement a system that will randomly pick a 5-letter English word so that our device is usable more than one time per day, but we will also take advantage of the ESP-32's Wi-Fi capability to fetch the daily word from The New York Times to maintain the critical social element of the game.

1.2 Goals and Objectives

Basic Goals

- Generation/selection of 5-letter words on device
- LCD display to show game progress and interface with user
- User input via on-device physical QWERTY keyboard
- Feedback via RGB LEDs diffused behind keyboard
- Powered by a lithium-ion battery
- Battery charged via USB-C port
- Wi-Fi capability via ESP32 (built in), for obtaining daily word

Advanced Goals

- Implement a buzzer speaker for auditory feedback (and personality?)

- Allow exporting game results as serial text data via USB-C

Stretch Goals

- Create wired local multiplayer through USB-C
- Touch screen
- Additional word games
- Multiple word lengths

Basic Objectives

- Create a Network API to obtain the daily word if necessary
- Create an algorithm that pseudo randomly generates words
- Create an algorithm that keeps track of the rounds
- Create code that allows the player to guess and for the guess to be checked for correctness
- Display colors that give hints to the player on the LCD
- Display the game playing interface on the LCD
- Light up relevant LEDs with the relevant colors that also give hints to the player (implement an LED multiplexer)
- Implement a keyboard of 26 letters in the QWERTY format, a backspace, and an enter key
- Create code that allows a button to interrupt and do a new game
- Implement a 5V regulator circuit
- Implement a 5W charger
- Implement a USB-C port/breakout
- Implement a power switch
- Create Wi-Fi configuration code

Advanced Objectives

- Create the relevant code which allows exporting (potentially take advantage of UART)

Stretch Objectives

- Implement LEDs that light up relative to who wins or loses, and this be accomplished through the USB-C cable
- Create code that can have smaller or bigger word sizes for the word that needs to be guessed, and rounds adjusted accordingly
- Implement capacitive touchscreen

1.3 Description of Features and Functionalities

The WordPal is a rectangular device with a keyboard of 26 letters arranged in the QWERTY format as well as a backspace key and enter key. It also has a button on right side on the top side that starts a new game whenever it is pressed. The WordPal can play a word guessing game (a clone of Wordle or Wordle inspired). In the game, the word is either pseudo randomly generated or the obtained daily word, and the player has to guess it, while

WordPal gives out relevant hints, in a limited amount (6) of rounds. The device has an LCD on the front side above the keyboard. The hint system of WordPal includes color changing LEDs which can be green for the correct input in the correct place, yellow for the correct input in a misplaced place, and red for an incorrect input. The LEDs are located behind each of the keys. There is also a hint system on the LCD of WordPal in relation to the player's guesses. The same colors of the LEDs apply for background of the letters of the guesses. The enter key allows the player to submit guesses, and the backspace allows the player to delete letters. A battery component will be a rechargeable lithium-ion battery, and it can be charged via USB-C (its USB-C port is on the top as well as a power switch next to it on the right). WordPal also has the capability of exporting game results as serial text data via USB-C. It will have Wi-Fi capability where it can access the daily word from New York Times; it will also have a buzzer speaker for auditory feedback.

For the hardware of WordPal, its MCU is the ESP32. Data flows from it to other components like the Keyboard, LED Multiplexer, LCD Display, and USB-C/Breakout, and data flows back from the Keyboard. The other places of data flow are from the power switch to the lithium-ion battery and from the LED multiplexer to the RGB Key LEDs. For the power, it flows like this: from the 5W Charger it goes the USB-C/Breakout, from there it goes to the lithium-ion battery, from there it goes to the 5V regulator circuit, and from there it goes to multiple components which are the LCD Display, the Keyboard, the MCU, the LED Multiplexer, and the RGB Key LEDs. The buzzer's incorporation is something that may not be fully decided upon yet.

For the software of Wordpal, it might go a bit like this. We first start off with the power on/initial state. From there it checks to see if the daily word was solved. Then it goes to different places depending on whether the answer is yes or no. If yes, it can either send the results over USB or it can generate a word. After the results are sent, it also goes to generate a word. Before we continue from there, let's go back to what happens if the answer is no for if the daily word was solved. It will go to check if the Wi-Fi is configured. If it is then it will get the daily word. If it is not, then it will go the Wi-Fi configuration and after that has been done successfully would it to get the daily word. Once the daily word has been obtained, it will go to the same place as where it goes after the word has been generated which is maybe where the game happens and where the player guesses the word. It will check for a correct guess. If it does then it will send the results over USB, and if not then it will give feedback via LEDs and then go back where the player can guess a word. There will also be an interrupt that can come from a button on the right side where it will go generate a word and go from there. There will also be a round checker which ends the game once six rounds have completed. In relation to the sending the results over USB-C, there may be some figuring out to do but UART could potentially be taken advantage of. For the programming language that might be used with WordPal, it might not be fully decided, but it could be C.

In relation to the stretch goals/objectives of WordPal, it could have wired local multiplayer through USB-C implemented, perhaps reminiscent of the wired multiplayer of the Game Boys via the Game Link Cable. Through a USB-C to USB-C cable that connects between them, it could light up the relevant LEDs between the two of them (either a red one for the loser and a green one for the winner), and it could also be implemented where the same word they are trying to guess is shared between them (though perhaps the words could also be different). For the touchscreen, it will be capacitive maybe also capable of multi-touch. There can also be additional word games. How they look like is maybe not decided currently. There is also the option that could be included of Wordle of multiple word lengths. Instead of 5 letters, it could be less or more, and the number of rounds could be changed relative to the number of letters. It might be that the round number will be one greater than the number of letters that will be guessed.

1.4 Existing Product/Past Product/Prior Related Work

We found two projects that stood out when looking at other handheld versions of Wordle. The first was a build by Michael Lacock, who made a Wordle clone on the Adafruit CLUE board using CircuitPython [2]. The CLUE already has a small color screen and two buttons, so it was an easy way to get the game working on hardware. Lacock made two versions of his build, one that relied on the A and B buttons to scroll through letters and confirm guesses, and another that connected to an external keyboard for a quicker input. Both versions displayed results on the screen just like the online game. These projects showed us that Wordle can be run on a small portable device, but it also showed the limits of using a dev board that already has everything built in. Since the CLUE handled the display, buttons, and power on its own, there wasn't much custom designing or wiring involved. Overall, the device worked as a demo but skipped the hardware challenges that we would encounter in this course.

The second project we looked at was Richard Falcema's Wordle handheld device, called "Wordle Term" [3]. This device was more about looks and design than electronics. It had a 25 square display grid and five wheels that players turned to pick letters, with the game giving feedback in orange, gray, and green. What stood out was the style of the device, the circuit board was exposed, the battery was visible, and it had bright colors and even a lanyard. However, this project didn't go into how the electronics work, so it came across more like a design concept.

Seeing both examples gave a clearer sense about what has already been done and where our project fits. Lacock's handheld device showed that Wordle logic can run on hardware, while Falcema's design gave an example of how a handheld device could look. For our project, it will expand on these two ideas. We are planning to design our own PCB and add buttons and LED indicators and connect an LCD display to run the game. Instead of just showing that Wordle can run on hardware or focusing only on how the device looks, our goal is to put the whole system together and build a working prototype.

These prior projects highlighted how important feedback and usability are in a handheld game. The Adafruit CLUE project worked, but the input method was limited to just two buttons on one of the versions, which slowed down gameplay. Our inclusion of a keypad in contrast that players can quickly select letters (which is also applicable to the other version that connects to an external keyboard), as well as LEDs to provide immediate visual feedback. Adding both features will make our design closer to the way the game is normally played and create a smoother user experience overall. Also, having multiple feedback channels, like an LCD for displaying guesses and LEDs for color cues makes the system easier to follow. To meet course expectations, we cannot rely solely on an existing dev board. Our project is going to focus on making the game easier to play while also tackling the real hardware challenges. The keypad and LEDs give us better input and feedback.

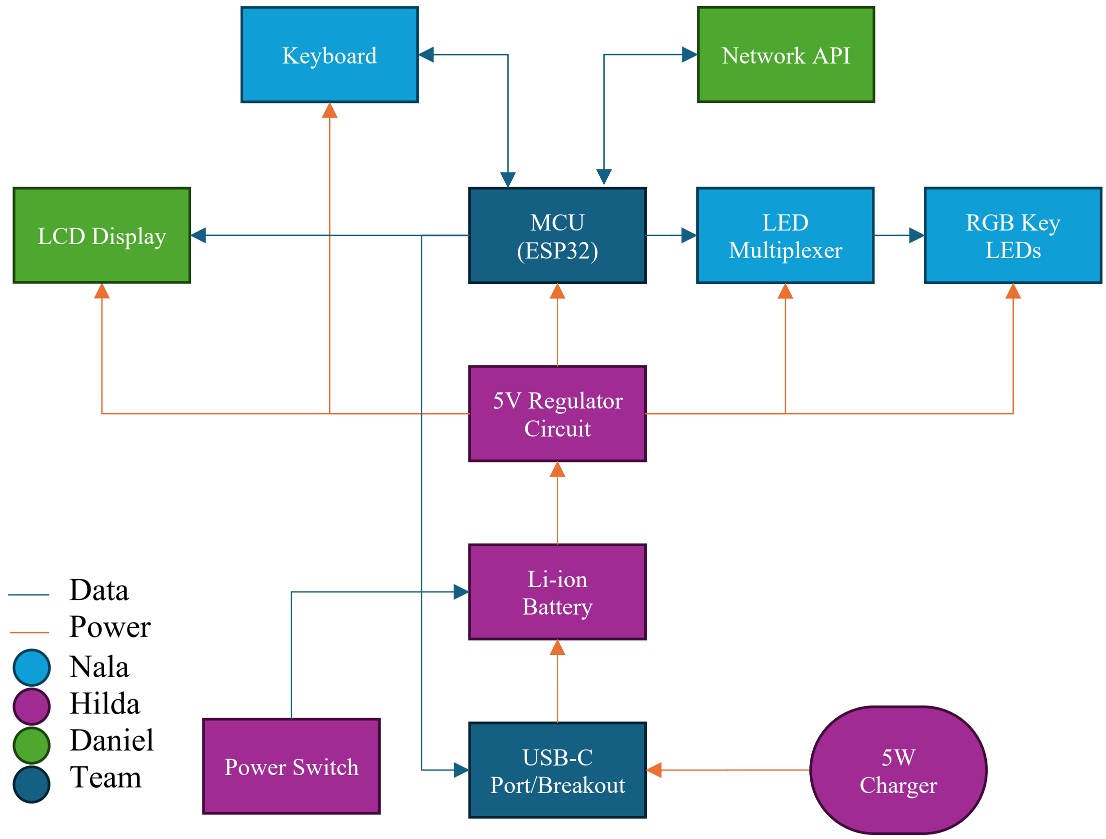
2.1 Engineering Specifications Table

Table 2.1: Engineering Specifications

General	
Dimensions	< 8 in x 5 in, < 1 in thick
Battery Capacity	> 1000 mAh
Battery Life	> 2 hours
Cost of Materials	< $350
Charging	>= 5V1A, USB-C
Weight	< 250g
Keyboard	
Response Time	Less than 100 ms
RGB LEDs	Addressable, <= 5V, >2 bpc
Display	
Response Time	Less than 100 ms
Brightness	> 250 nits

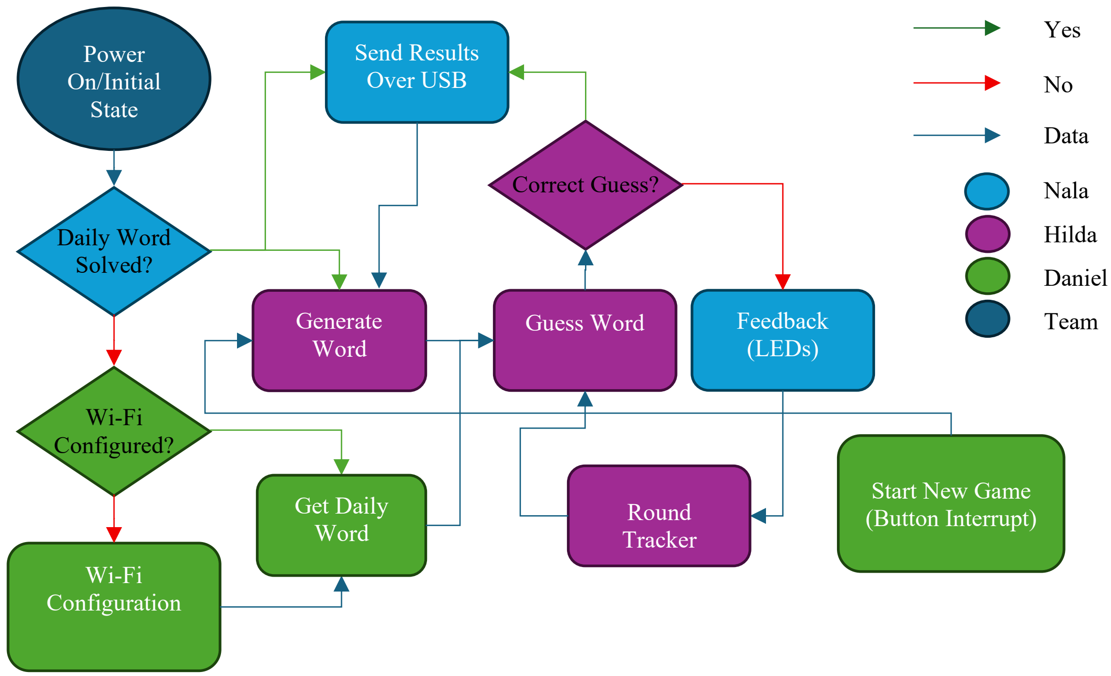
2.2 Hardware Diagram

For the following diagram, the status of all of the blocks may be currently to be acquired.



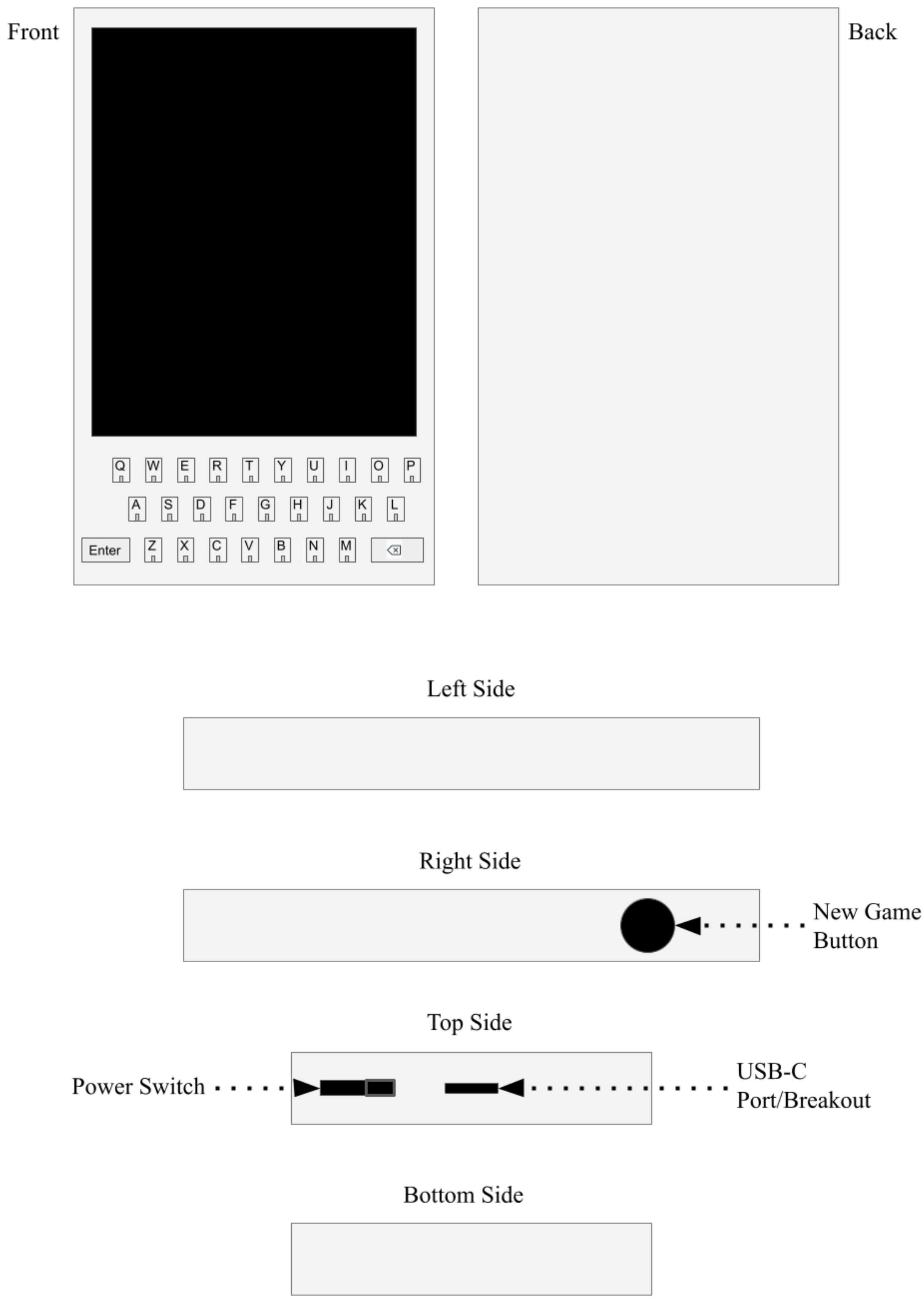
2.3 Software Flowchart

For the following diagram, the status of all of the blocks may be currently to be acquired.



2.4 Prototype Illustration/Blueprint

The prototype design might still be a work in progress. The design might be similar to the BlackBerry Keyone where the screen was quite a bit bigger than the keyboard. The ratios of the design is in line with an 8 in. length by 5 in. width and a 1 in. thick type ratio though the actual device might be smaller than that. The prototype’s outer casing might be built with a 3D printer, and the keys are perhaps semi-translucent with the LEDs being able to shine through behind the keys.


Front
Back
Q W E R T Y U I O P
A S D F G H J K L
Enter Z X C V B N M
Left Side
Right Side
New Game Button
Top Side
Power Switch
USB-C Port/Breakout
Bottom Side

2.5 House of Quality

Table 2.5: House of Quality

	Response Time	Display Brightness	Battery Capacity	Weight/Dimensions	Cost
Response Time			↓		↓
Display Brightness			↓		↓
Battery Capacity				↓	↓
Weight/Dimensions					↑

Customer Requirements		Design Requirements	Response Time	Display Brightness	Battery Capacity	Weight/Dimensions	Cost
			-	+	+	-	-
Customer Requirements	Ease of Use	+	↑	↑	↓	↓	↑
	Portability	+			↓	↑	
	Battery Life	+	↓	↓↓	↑↑	↑	↓↓
	Charge Time	-			↓		↓↓
	Cost	-	↓	↓	↓		↑↑
Targets			<100 ms	> 250 nits	>1000 mAh	<= 8x5x1in, <=250g	<$350 excluding prototyping

2.6 Budget and Financing

Because our design is still being developed, the numbers in this budget should be seen as estimates rather than final values. The costs shown are based on online prices for the kinds of parts we plan to use, with room for adjustment once we choose exact components. The higher costs are expected from the PCB fabrication, the microcontroller, and the display. Other items such as the keypad, LEDs, power circuitry, and the enclosure are smaller line items but still important. Since prototyping often requires ordering extra boards and backup batteries, the quantities are set slightly higher in some cases. Taking everything into account, the current estimate is around $335, but the final amount will depend on part selection and any changes we make as the project moves forward.

Note: This is a self-funded project, so we there is not a strictly constrained budget at this stage. The values shown below are general estimates, and the budget can be refined once more specific details are finalized.

Table 2.6: Budget and Financing

Component	**Quantity**	**Price (Est.)**	**Total Cost (Est.)**
PCB Board	3	$35	$105
Microcontroller Unit	1	$20	$20
Rechargeable Battery (Li-ion)	2	$15	$30
LCD Display	1	$20	$20
Keypad / Buttons	1	$15	$15
RGB LEDs	1	$25	$25
USB-C Port	2	$10	$20
Voltage Regulator	2	$30	$60
Power Switch	1	$10	$10
Enclosure	1	$30	$30

-	-	-	$335

3.1 Milestones

Table 3.1: Senior Design I - Milestones

Senior Design I – Fall 2025

Week	**Project Task**
1	Course intro, team formation, brainstorm project ideas
2	Finalize our group and decide on the project idea
3	Write and turn in the Divide & Conquer report
4	Meet with advisor about D&C and start drafting requirements
5	Create list of main components (LCD, LEDs, keypad, battery, etc.)
6	Order sample parts, draft block diagrams, initial test plans
7	Test basic LCD functionality
8	Test basic LED functionality
9	Test basic keypad functionality
10	Begin integrating simple game logic with components
11	Submit Midterm report
12	Midterm review meeting
13	Work on schematic design and component testing
14	Upload final report draft and prepare demo video
15	Submit final report and demo video
16	Reviewer evaluation period

Table 3.2: Senior Design II - Milestones

Senior Design II – Spring 2026

Week	**Project Task**
1	Review progress from Senior Design I
2-3	Begin PCB assembly
4-5	Test major subsystems (power, display, LEDs, etc.) to confirm basic functionality
6-7	Troubleshoot any issues and adjust code and wiring as needed
8-9	Integrate all subsystems together and refine game performance
10-11	Prepare final presentation materials
12-13	Final debugging and adjustments
14	Conduct full run throughs of the demo, report, and finalize slides
15	Submit final report and demonstrate completed project

Appendices

Appendix A – References

[1] "University of Central Florida Logo," 1000logos.net. [Online]. Available: https://1000logos.net/university-of-central-florida-logo/ . [Accessed: Sept. 4, 2025].

[2] M. Lacock, "A handheld version of Wordle using a CLUE and CircuitPython," Adafruit Blog, Feb. 11, 2022. [Online]. Available: https://blog.adafruit.com/2022/02/11/a-handheld-version-of-wordle-using-a-clue-and-circuitpython-clue-circuitpython-michael_lacock/ . [Accessed: Sept. 4, 2025].

[3] R. Falcema, "Quirky Wordle handheld game device feels heavily inspired by Game Boy and Teenage Engineering," Yanko Design, Aug. 30, 2024. [Online]. Available: https://www.yankodesign.com/2024/08/30/quirky-wordle-handheld-game-device-feels-heavily-inspired-by-game-boy-and-teenage-engineering/ . [Accessed: Sept. 4, 2025].