



## **Honors Capstone**

Sponsor: **Learning A-Z**

Project: **Letter Chase**

By: **Nathan Zika**

Special Thanks:

Sponsor Mentor: **Alexandra Brown**

Faculty Mentor: **Amy Hortop**

Team Members:

**Abigail Seguin**

**Ahree Hong**

**Bhargav Malladi**

**Dylan Rodriguez**

**Suraj Harjani**

December 10, 2021

## **Section 1: Project Overview**

### **1.1 Background**

Students of all ages are beginning to experience online lesson plans and assignments integrated into their curriculum. Learning A-Z offers students of all ages an online platform to either practice their own skills or have assignments that align with their schoolwork. An interactive spelling game for K-5 students will give teachers a way to assign a list of words for the students to spell or students will be able to play the game on their own time with randomly generated words at the student's reading level.

It is important to design games for K-5 students that have a healthy medium between challenging and entertaining. The overarching goal of the game is to provide a platform for students to feel comfortable in learning how to spell.

### **1.2 Expected Deliverable**

Learning A-Z has requested a web-based game that supports gameplay with the following components: a playable student character, a student reward for every word spelled correctly, a timed aspect in the form of a monster chasing the same student reward, a shuffled 2D array of letters, a variety of words with the length being their difficulty, and audio support for words.

This game must follow the general framework given by the Learning A-Z sponsor: the game must support the above components, users must navigate with the UDLR keys, and must use the same technology stack as Learning A-Z. Our team has the authority to design most of the gameplay, including specific behaviors of both the monster and the student character and the overall physical design of the game. Success will be evaluated based on how well our game meets the described criteria.

This tool should be compatible with Learning A-Z's standards and should support both generating the game's word list based on our database or from an entered vocabulary list by the student/teacher. Finally, our deliverable should also extend the UI to support very new readers and should have taken into account user feedback. The minimum viable product (MVP) for our project is a web-based working prototype of the game that allows students to hear a word and use the provided array of letters to attempt to spell the word correctly. The game must generate a new board for every stage of the game and must include UI elements that complies with the Learning A-Z brand standards. Stretch goals include extending support for mobile formats and outputting useful information for teachers.

The objectives of this product are as follows: explore using AngularJS technology for future Learning A-Z products and develop a new game for students to use to learn how to spell. With the development of a successful project, Learning A-Z will be able to use our work to launch a new Vocabulary A-Z game for students to aid their online learning, and will have an example to see how AngularJS may work for their other vocabulary and spelling games.

### 1.3 Objectives/Scope

The main objective of our project is to create a web-based prototype of a spelling game for young, K-5 students. The game should include the ability for the student to control a player character that navigates a grid of randomly generated letters to select the correct letters for a given word. The word will be conveyed to the student through some combination of playable pronunciation audio, an image representing the word, and the word's definition, all of which will be provided when available. The game should also have a timed component in the form of a monster that moves across the screen.

Development of this game involves creating all necessary front-end components (i.e. all components of the game on screen) and developing back end functionality for retrieving words and definitions. The scope of the project does not include connecting to any of Learning A-Z's databases, due to authentication issues. Instead, Learning A-Z has provided us with database copies for us to use during development, and a handful of word audio and image files to test the audio playback and image display features.

Other objectives in scope include features for student accessibility outlined in Learning A-Z accessibility documentation, thorough testing of all components, and creation of necessary algorithms or logic for all the components described above.

### 1.4 Stakeholders

There are three main stakeholders for the game: Learning A-Z, K-5 Students, and Teachers/Caretakers

*Learning A-Z:* Wants an end-to-end prototype for an interactive spelling game that serves as a proof of concept for AngularJS as a viable front-end framework.

*K-5 Students:* Wants a fun and engaging spelling game that challenges their spelling ability.

*Teachers/Caretaker:* Wants a way to extend their spelling and reading curriculum into an online format and wants a way to monitor students' performance through custom word lists.

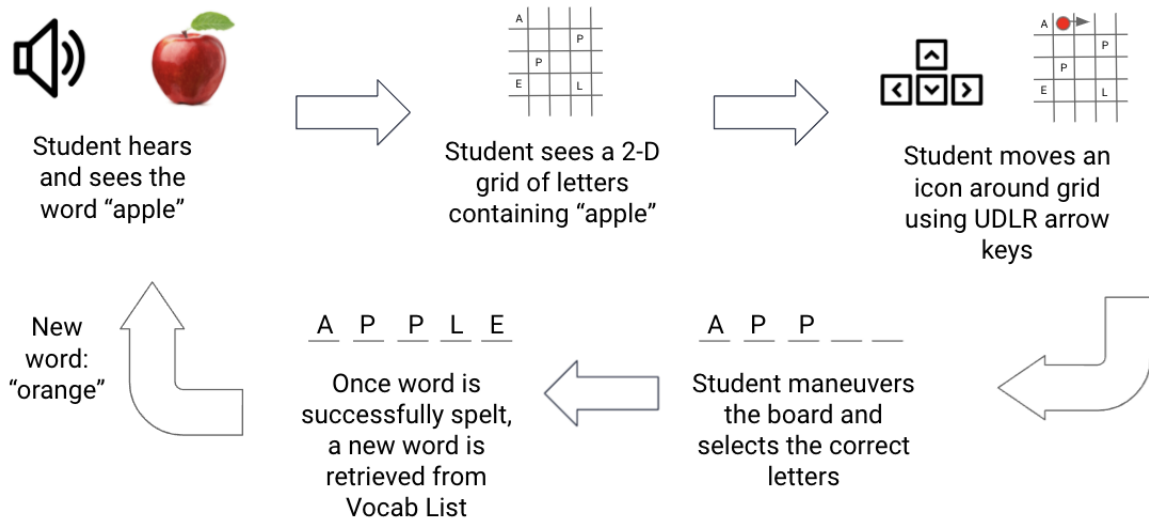
### 1.5 Value to the Sponsor

Through our project, our sponsor hopes to receive an engaging spelling game that they can seamlessly integrate into their Vocabulary A-Z product line, which they are currently revamping. Furthermore, the game will serve as a proof of concept for the viability of AngularJS (the framework we are using for development) in creating similar products for customers.

## Section 2: Solution Strategy

### 2.1 Basic Game Flow

Successful games for young audiences often rely on a repetitive game flow and easy to understand mechanics. Letter Chase offers simple controls and an intuitive design to help the student learn the letters of a given word.



**Figure 1:** One iteration of Letter Chase with the first word being “apple”

As shown in **Figure 1**, students will receive audio and visual clues of the word before they begin navigating a grid of letters that contains the letters of the word they are trying to spell. One iteration of the game proceeds as followed:

1. The student will hear a pronunciation of the word and an image of the word will appear if it exists
2. A 2-dimensional, 5x5 grid of random letters is generated, the letters of the word are contained within the grid
3. The student will navigate an icon above the letters of the grid with either their AWS D keys, ULDR keys, or a D-Pad controller
4. When the student navigates above the next correct letter of the word they are trying to spell, they will click the enter key
5. The letter of the word will be removed and sent to a word stub that is displayed on top of the screen to show the progress through the word
6. Once all the letters of the word are selected, the next word is retrieved, and the process begins again at step 1.

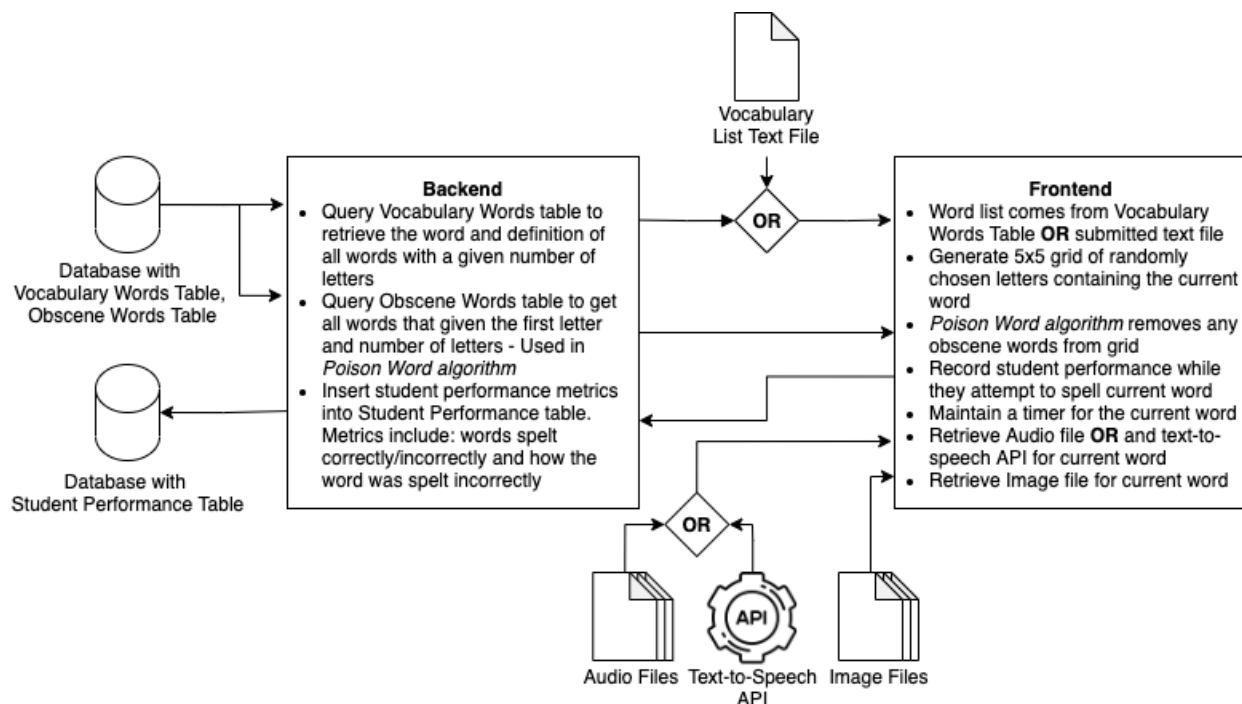
The game flow outline above is repetitive which is consistent with the existing spelling games provided by Learning A-Z and finding the letters of the word within a grid is unique and will complement the existing games.

## 2.2 Pirate Theme

We had freedom to choose the theme of the game. After considering a multitude of game designs including zombie, farm animals, and knights, we decided to go with a pirate themed game. We believed that it worked best with the components of the game such as a ship navigating the sea to find the letters of the word, a kraken swimming towards a treasure of gold to act as the timing mechanism and rewarding the student with gold whenever they spell a word correctly. The pirate theme allowed for an appropriate combination of silly and functional to keep the student engaged. Additionally, the absence of characters allowed the game to be gender inclusive. Furthermore, we were able to create graphics that had suitable color-contrast levels that would support color blind users.

## 2.3 Basic Architecture

The backend of Letter Chase will be written in PHP, the scripting language used by Learning A-Z. The goal of the backend is to create an API that can make SQL queries to Learning A-Z's Vocabulary and Obscene words databases as well as a Student Performance database. The frontend will be written in AngularJS, a Learning A-Z constraint. According to Learning A-Z's practice of separation of concerns, each key feature of the game will be its own component that depends on data maintained in a provider or service, while remaining independent of other components. There are 5 main components of Letter Chase: Grid, Word Stub, Monster/Timer, Audio, and. Each component depends on the current word being spelt.



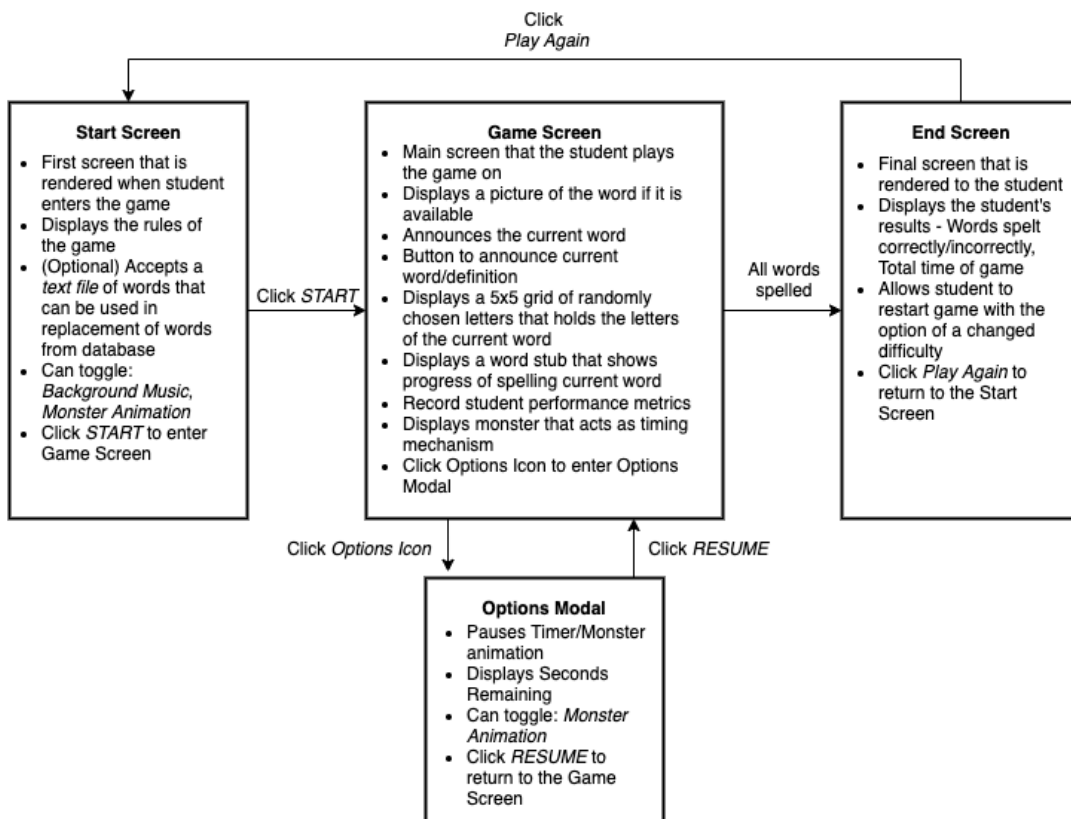
**Figure 2:** Basic architecture of Letter Chase. Explains the key features of both the Backend and Frontend, as well as their dependencies on respective Databases, files, and APIs.

For every iteration of the game, it will flow as follows:

1. Retrieve the next word from Vocabulary Words DB or a text file submitted by a teacher
2. Generate a 5x5 grid of random letters, then select random indices to put the letters of the current word, then cleanse the grid from any inappropriate words using the *Poison Word* algorithm.
3. Play word audio, and display the image file for the word if it exists
4. Trigger the monster and internal timer after the word audio finishes playing
5. Student maneuvers grid to select correct letters, which are then displayed in the Word Stub
6. Once all the letters of the word have been selected or time runs out, go back to step 1

## 2.4 Front End Design

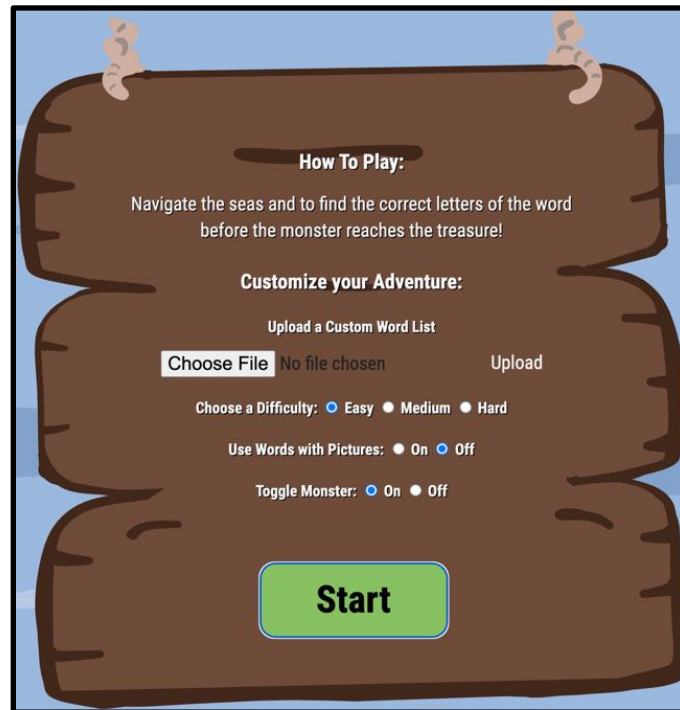
The game will consist of three separate screens: Start Screen, Game Screen and End Screen. When the student enters the game, they will be shown the Start Screen with basic onboarding instructions and options to toggle various game features. The Game Screen will be responsible for displaying the grid of letters and all the additional features that will support the student in spelling the word correctly. Lastly the End Screen will display the student's performance of the words they just attempted to spell.



**Figure 2:** High level overview of the three main screens of Letter Chase. Explains the main features of each screen and how to navigate between them.

The game flow pictured in **Figure 2** is consistent with the games on the existing Learning A-Z website. Each game allows the student to customize certain features before they begin playing the game. Additionally, each game has a pause button that ultimately stops any timing mechanism that might exist in the game as well as allow the student to toggle any features. Lastly, each game has an “end” which displays to the student their performance on the words they were required to spell. We did our best to emulate the game flow of their existing games.

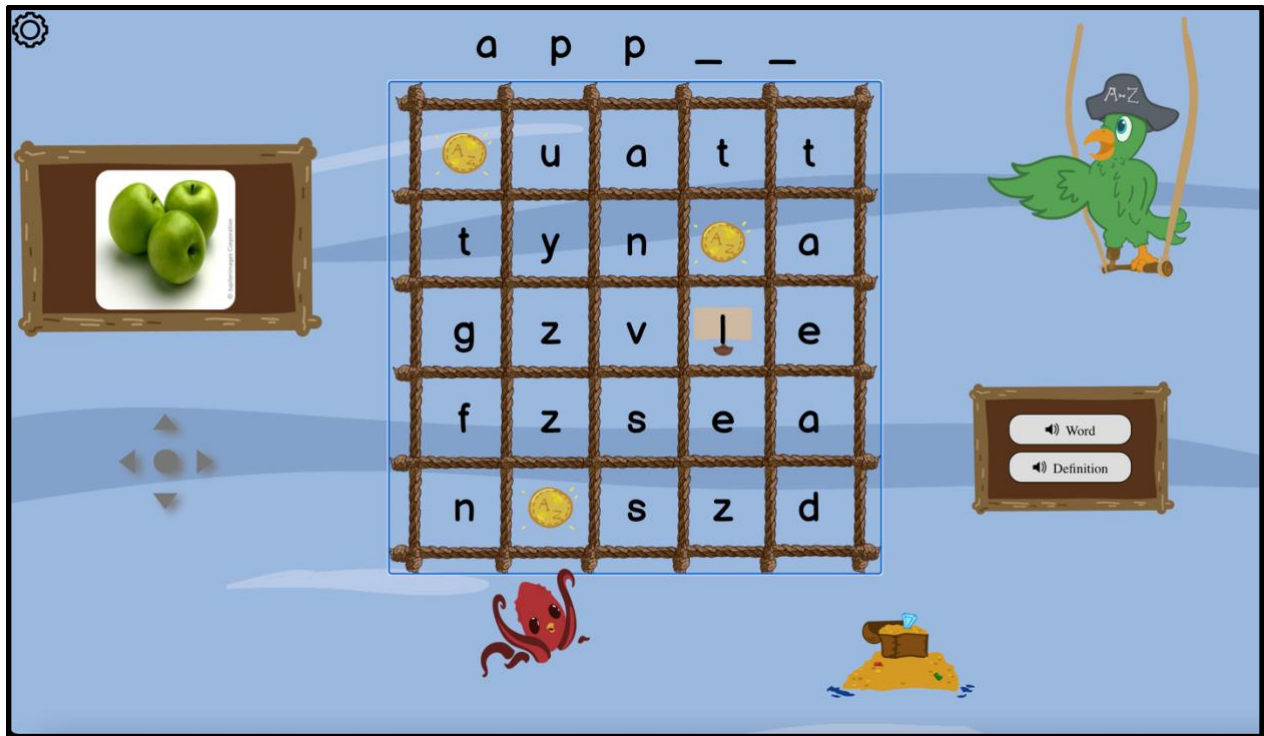
## 2.5 Start Screen



**Figure 3:** The start screen modal which allows the student to customize the game.

When the game is first rendered, the start screen modal is the first thing that is displayed. The start screen provides a brief, pirate themed, introduction to the game. The introduction informs the student that they must navigate the grid in search of the correct letters of the word. Secondly, the student can customize certain features about the game. As a proof of concept that the game can accept a custom word list, the student and/or teacher can upload a custom text file full of words. If no text file is uploaded, then the words used within the game will come directly from Learning A-Z’s database. Students can also select the difficulty of the game. Difficulty for Letter Chase is defined as the number of seconds per letter of the word that the student has before time runs out with Easy, Medium, and Hard being 15 seconds, 10 seconds, and 6 seconds, respectively. Next, students can choose if they only want to play the game with words that have an associated image. Lastly, to make the game accessible to all users we have added the option to enable or disable the monster animation. We believed that there could be an audience who might get discouraged or feel intimidated by the monster, thus we wanted to provide an option to disable it. Once the student has selected all their settings, they can click Start to advance to the Game Screen.

## 2.6 Game Screen



**Figure 4:** A screen shot of the main game screen. The current word is “apple”, and the student has previously selected the letters, ‘a’, ‘p’, and ‘p’ and is about to select ‘l’.

The Game Screen is the main screen of the game. It can be divided into three components that meet the requirements of the game: the navigable grid of letters, the audio/image clues, and a monster. Between each iteration of the game, a new grid is generated for the new word, the image and audio files are updated, and the monster is reset.

### 2.6.1 Grid Initialization

A new grid is generated for each word. Each index of the 5x5 grid gets assigned a random lowercase letter from the English alphabet. Then, the letters of the word are assigned to random indices within the grid. The indices cannot repeat. Once the grid is assigned all the letters, we verify that there are no obscene grids by using the *Poison Word* algorithm. The *Poison Word* algorithm is something we developed to ensure that there are no horizontal or vertical substrings of lengths 2 through 5 that could be classified as an explicit or obscene word. Learning A-Z provided us with a database table that holds every obscene word that they have identified throughout the years. When Letter Chase begins, we make an API request to retrieve all Obscene words of lengths 2 through 5. This list persists throughout the duration of the game. The *Poison Word* algorithm verifies that all possible horizontal and vertical sequence of characters within the grid do not result in a word that exists in the list. If an obscene word does exist, a random letter of that obscene word is swapped with a random letter that exists elsewhere in the grid and the algorithm is run again.



### 2.6.2 Grid Navigation

The students control a boat that can navigate the letters of the grid. The students can control the boat using the AWS D keys, Up Down Left Right keys, or the D-Pad that exists on the screen. When the student is under the next correct letter of the word, they can either click the Enter key or the circle in the middle of the D-Pad controller to select the letter. If the student selects an incorrect letter, the game announces a friendly audio file which says, “Keep Practicing!”. If the student selects a correct letter, the letter is removed from the grid and sent to the word stub that exists at the top of the screen. The index of the letter they just selected is replaced with a gold coin. In previous iterations of the game design, the letter would be replaced with a red ‘X’ to fit the pirate theme, however as suggested by the sponsor, the ‘X’ typically has a negative connotation.

### 2.6.3 Audio Clues

For the student to know which word to spell, they will audibly hear a pronunciation of the word. Learning A-Z has a database dedicated storage for audio files for every word in their database, however we were not allowed access to these files, thus we had to use a Text-to-Speech API. After the previous word is completed, the Text-to-Speech API announces the next word to the student. As seen in **Figure 4** there exists two buttons in the bottom right part of the screen labeled “Word” and “Definition.” The student can click on these buttons to listen to them. The definition comes from the Learning A-Z vocabulary database. Additionally, there exists audio clues whenever a student selects a letter. If the student selects an incorrect letter, the game announces a friendly audio file which says, “Keep Practicing!”. If the student selects a correct letter, a subtle chime can be heard by the student. The goal was to have persistent positive reinforcement whether the student got the letter correct or not.

### 2.6.4 Image Clues

In addition to audio clues, Letter Chase also offers image clues for students. If an image for a word exists, the image is displayed in a picture frame to the left of the grid. Given that not all words have an affiliated image, if no image exists, neither an image nor the picture frame will be displayed on the screen. In the starting screen modal as seen in **Figure 3**, there is an option to only choose words with an image. If the student enables this setting, then only words that have a non-null image file path in the database will be retrieved. Like the audio clues, we did not have access to Learning A-Z’s image storage system, thus we used hard coded image files for the purposes of demonstration.

### 2.6.5 Monster/Timer

To encourage the students to spell the word as quickly as possible, Letter Chase also includes a timing mechanism in the form of a monster swimming towards an island of gold. The amount of time a student receives per word is dependent on the difficulty they chose in the starting screen modal as seen in **Figure 3** with Easy giving 15 seconds per letter of the word, Medium giving 10 seconds, and Hard giving 6 seconds. If the student does not find all the letters of the word in the allotted time, an intermediary modal will be displayed which will show the correct spelling of the word as well as have buttons to hear the word and definition of the word. A monster swimming towards an island of gold was chosen because it gives the student an incentive to spell quickly, but also not scare the student. We considered having a monster directly chase the boat within the bounds of the grid, but we determined that this might discourage the student.

## 2.7 End Screen



**Figure 5:** The End Screen Modal is displayed after a certain number of words is completed. It shows the number of words spelled correctly.

The End Screen is displayed once after a given number of words is attempted. If the word list is randomly generated from the request to the vocabulary words database, then the default number of words is 10. If the word list comes from a custom text file, then the number of words is the number of words within the text file. The End Screen displays the number of words the student spelled in time over the number of words attempted. The End Screen also provides an option to play again which redirects the user back to the Start Screen. Lastly, if the word list came from a custom text file, then the student's performance is logged in a student performance database table. Each row of the table consists of a word id that is associated with the word id in the vocabulary words table, a unique student id, the date attempted, the name of the text file that the word list came from, and Boolean which indicates whether the student completed the word in time.

## 2.8 Back End Design

The backend for Letter Chase is primarily responsible for creating an API that queries a copy of Learning A-Z's databases. Considering limited storage space of the servers that were responsible for running Letter Chase and the sensitive data being kept within these databases, we did not have access to Learning A-Z's actual databases, but rather a snapshot of them as of January 2021. The backend was limited to using PHP to set up the API endpoints and MySQL to query each of the databases.

### 3 Work Remaining / Next Steps

Letter Chase is currently based suited as a computer game. We did not have the time nor the experience to cater the game to fit mobile screens. Thus, before Learning A-Z can integrate the game, they should revisit the user interface design to fit mobile devices. Additionally, we were unable to implement a background noise feature. At the beginning of the project, we identified background noise as a stretch goal. We believe that hearing water flowing in the background would go a long way in making the game feel more alive to the students. Letter Chase also implements a proof of concept of student logging metrics. We did not have access to Learning A-Z's performance logging APIs so we created our own, however switching to Learning A-Z's would not take much effort. Lastly, the game utilizes a default 5x5 grid that the students can navigate, and the difficulty determines the number of seconds per letter of the word; it might be worth looking in to having a scalable grid based on the number of letters of the word to make the game more challenging. Overall, due to the games modular coding stack, these changes should not involve too much additional work.

### 4 Conclusion

Letter Chase meets the expected final deliverable. The spelling game will a value asset for Learning A-Z as it fits in well with the existing spelling and reading games. The foundation of the game is complete and should have seamless integration in to Learning A-Z's coding stack. Students will be able to enjoy a new environment to learn how to spell.

Working on this project has given me an opportunity to strengthen my leadership and my technical communication skills. I had always viewed myself as a strong leader, however before this project I had not had many experiences to demonstrate my abilities. Given that this was a two-semester long project, motivation and determination ebbed and flowed. I took it upon myself to both lead by example as well as respectfully delegate tasks. After the end of the first semester, I noticed that we were not doing a good job at managing our weekly sprints, thus, to begin this semester I planned out the remaining work for the project and put it all in a Jira Board. The Jira Board allowed each team member to volunteer to add a feature to the game. The transition to better project management practices allowed the team to work better asynchronously and guarantee that each member is doing an equal share. Additionally, I learned the best ways to run meetings amongst my team members as well as with our sponsor. I found that having an agenda and slides prepared allowed for a more efficient meetings. Lastly, as a requirement by the Multidisciplinary Design Project course, we were required to create Executive Summaries and present Design Reviews. I was able to gain a deeper understanding of how to efficiently transfer knowledge in industry by means of written paper or by presentation. My confidence in both my writing skills and being able to speak in front of an audience dramatically improved because of this project.

Overall, working on the Learning A-Z MDP team was an amazing opportunity. The lessons I learned, both technical and interpersonal, will prove to be invaluable as I advance in my career as an engineer. Furthermore, creating a game that could potentially influence the trajectory of K-5 students will always be something I remember.