Kimbee: A Speech Therapy Application For Children

Ryan Drapeau
University of Washington

Nick Huynh
University of Washington

Aaron Nech
University of Washington

{drapeau, huynick, necha}@cs.washington.edu

ABSTRACT

We have developed an application named *Kimbee* that aims to help improve the process of speech therapy in young children. Speech therapies are often expensive for schools and parents to maintain, especially with younger children who require more time than others. Our application does not replace the speech therapist, rather it supplements the therapy. Instead of being limited to short sessions, therapists can use *Kimbee* to have their students work at home as well as in the classroom. The therapist can then monitor the child's progress through our application and use this newly gained information to further tailor the therapy to individual students. This will save both time and money for school districts and parents who are paying for their child's therapy.

1. INTRODUCTION

The purpose of this paper and project is to make speech therapy more effective in English speaking children from age 5 to age 12. Children in this age group tend to average over a year involved in weekly speech therapy meetings to overcome a speech disorder [4]. It is also common for therapists to spend most of their time with students during these sessions or in the classroom because it is often hard to work with children remotely. A session usually consists of having the child repeat words back to the therapist in order to target a specific sound or phoneme. Common speech impediments in children are often caused by functional speech disorders, or trouble learning to make specific speech sounds [1].

We chose to focus on 'R' sounds as they are one of the more common mispronunciations found in our target age group [2]. Most 'R' disorders will commonly pronounce words as if the 'R' was a 'W'. An example would be pronouncing the word *"rabbit"* as *"wabbit"*. We use similar techniques as therapists use in their sessions in our online application. By having the child or user repeat a target several times until he or she is able to pronounce it correctly, we are emulating the in person setting that they are used to. Each time a word is spoken, an audio recording of the speech is documented and saved for the therapist to analyze at a later time. This process is then wrapped in game setting to make the process of saying and repeating words more enjoyable for children. This helps supplement the therapy, rather than aiming to replace it. We hope that this will cause a decrease in the amount of time that children will need to spend in therapy, which will help save the school district or parents more money.

We will also show how our application is general enough that it can be expanded to beyond 'R' sounds making it applicable to any of the 36 different types of speech impediments. This will make our application a useful tool to use in all therapies that are being used with children's speech.

2. RELATED WORK

With the widespread use of the tablet and increasing availability of the Internet, several applications have been developed that are similar to *Kimbee*. Researchers at the University of Zaragoza have created a standalone tablet application that aims to remove the therapist from the treatment [10]. The researchers created an application for children to speak words and receive feedback on whether the word they said matches the correct pronunciation of what was on the screen. The model they used also requires the student to train his or her voice in order for it to be recognized and processed correctly. Because of this, the amount of time it would take a new student to get started is significant. Their application currently only supports the Spanish language, which would not be of much help to English speaking students. Our project's scope is almost much narrower. Our focus is to help the therapist spend higher quality time with his or her students by giving them additional information from the student's work outside the sessions.

There has also been work designing an elaborate collaboration between students and therapists. The European Department for Speech, Music, and Hearing published an article describing a new system and plan for speech therapy in children [7]. This paper addresses the fact that having a visual feedback system is essential for children to be able to learn and improve their speech. However, this paper fails to provide an implementation or describe one further than a few diagrams. The real-time feedback system and long-term statistics described are very similar to what we have implemented in *Kimbee*.

In addition to the work previously listed, there has also been a lot of development for speech therapy applications on Apple's iTunes Store and Android's Google Play Store [9]. A quick search for speech therapy mobile applications will reveal many results. However, most of these applications fail to incorporate a therapist into the learning model, which we believe is essential for helping the child overcome his or her disorder.

3. APPLICATION DESIGN

We will now introduce our implementation and solution to the problem posed for children's speech therapy. Before we started designing *Kimbee*, we met with speech pathologist Richard Kreider to discuss what features would be needed for a system like this to work [4]. Kreider has over 25 years of experience working with children to overcome their speech disorders. Kreider talked about how this application would need to have a way for children to practice at home as additional work as well as a way for him to track the progress of his students over time. As these were the two most important features Kreider discussed, we designed Kimbee around them as core features.

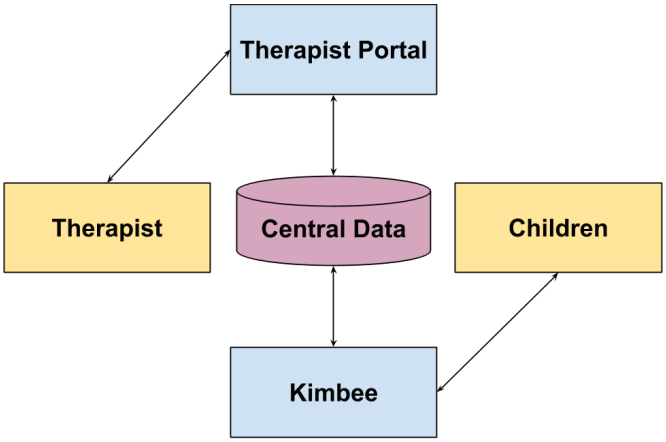

Figure 1: Left: The game, *Kimbee*, where children play the role of a bee trying to collect honey as a resource. Right: The flow diagram showing how therapists and children exchange data through the game and the portal.

3.1 User Flow

The flow is split into two parts, one to describe the game the child or user would play and another to describe what the therapist would see after the child has played the game. This will follow the example that the therapist has assigned *Kimbee* as homework for the child to do at home outside of the session. When the game is launched for the first time, a unique identification string is generated as a token for the child. This token should then be sent to the therapist allowing them to track the progress of the student. It is suggested that this first launch and setup take place in an in-person session with the therapist to alleviate any technical problems that may arise.

3.1.1 Gameplay

The game has a simple interface that is easy for a child to navigate, see Figure 1. Children simply tap the screen to make their character, a bee, move along the y-axis of the screen. Collecting honey causes their score to increase and colliding with an enemy wasp causes the game to end. Interleaved with the enemy wasps and honey pots is a special honey pot that will grant a significant amount of honey if a challenge is completed. This challenge consists of having the child speak into the microphone of his or her device and say a designated word or phrase that appears on the screen.

The difficulty of the text increases over time to simulate the treatment that the child would receive during a therapist session [8]. If the system succeeds in verifying that the child spoke the text correctly, then the currency is added to the child's score. If the child fails to correctly pronounce the phrase or if the system fails to recognize the audio, then the child is asked to try again in order to continue. To avoid discouraging the child, *Kimbee* will say a positive statement asking him or her to try again to continue playing the game.

After every challenge, the audio spoken by the child is encoded and sent to the central data server, see Figure 1 for a diagram of this. These data can be examined and reviewed in the therapist portal.

3.1.2 Therapist Review

The therapist is responsible for tailoring lesson plans for each session with his or her students. Currently, this involves taking detailed notes during every session to document and track the progress of each student over time [4]. The goal of the Therapist Portal is to improve and simplify this process to increase the quality of treatment given to the student. As described in section 3.1.1, every recording that each child produces will be sent to the central server. These recordings are then sent to Therapist Portal and listed for every student under the therapist. Additional data is sent down with each recording to help the therapist focus on words that the student had trouble speaking. For example, whether the child said the word or phrase correctly is included with the recording allowing the therapist to filter the recordings.

After listening to all of the recordings that the child produced for homework, the therapist will have a good idea for where the student currently is in terms of progress. With this new information, the therapist will be able to start of the next in-person session with a more focused lesson plan.

3.2 Currency

As mentioned in section 2, many other applications failed to create a captivating game for children. We wanted to have an application that children would see less as homework and more of a game. This was the underlying motivation for making *Kimbee* a game. In order for a game to be immersive, a sense of progression must be present [3]. We were able to create this using the honey collected as a currency. We also added the dynamic scoring of the honey, with the collection of the pots and completion of the challenges.

Many therapists currently use operant conditioning as a reward system [4]. For example, Kreider rewards children with small toys after they do well on a set of phrases. A reward system could be implemented where the currency is the honey from *Kimbee*. This would encourage students to play the game outside of the therapy sessions and motivate them to improve over time. As this was only a quarter long project, this is left as future work.

4. IMPLEMENTATION DETAILS

We will now discuss the implementation details of each of the systems that make up *Kimbee*. This project is made up of many components. In particular we will discuss the three main systems: the Server, the Game, and the Therapist Portal. From a high level, the Game and the Therapist Portal are thinly interconnected to the Server via HyperText Transfer Protocol (HTTP).

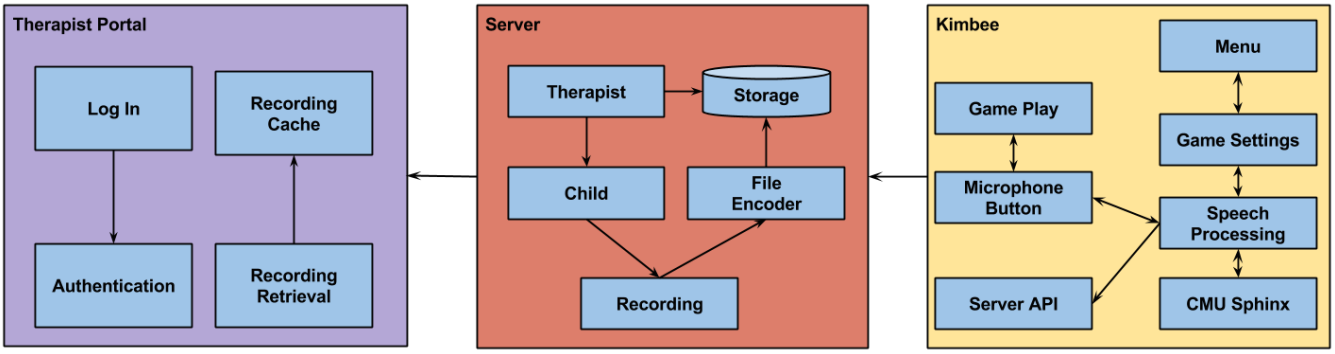

Figure 2: Left: The Therapist Portal, the user system of the therapist. Middle: The Central Server, hosts the recordings and facilitates interactions. Right: The Kimbee Game, interactive game that allows practice of speech for children.

4.1 Game (*Kimbee*)

The *Game*, see Figure 2, is the system that the children interact with to practice speech. It is written in TypeScript and compiled to optimized JavaScript. Since this system also performs speech processing to determine speech correctness, there are many interacting components. From a high level, the major sub systems are: the *Phaser Game Implementation*, the *Speech Processing Component*, and the *Server Application Programming Interface (API)*. We target this project toward web technologies to allow wide distribution and adoption. From a design standpoint, we strived to make each system loosely connected and non-disruptive to gameplay. This required us to explore various new web technologies including HTML5 Web Workers for multithreading and the UserMedia HTML5 API.

4.1.1 Platform choice

Since we want this project to work on as many devices as possible, we target emerging web technologies that work seamlessly across all devices with a web browser component. The web technology stack is advanced enough to handle our needs: there is a JavaScript port of the popular speech processing library CMU Sphinx, and we now have access to the user's microphone via UserMedia HTML5 API. After creating our project in the browser, we then can translate it to native applications via Apache Cordova, a web distribution technology based on embeddeding a web component inside a native application container on mobile devices.

4.1.2 Phaser Game Implementation

The *Phaser Game Implementation* of the *Game* encompasses the view and control of the application: The main menu, the game play, and rendering. This is what the children interact with to practice speech. The game is implemented with the Phaser Game Framework which represents screens of the game into states. States are linked together in meaningful ways. For example, the Main Menu State links to the Option Menu State via a Phaser Framework Button. It also provides image animation, tweening, and collision support. To create our microphone button, we extend the Phaser Framework Button and implement connections to the speech processing component while preserving button functionality through inheritance.

4.1.3 Speech Processing

The *Speech Processing* of the *Game* handles all audio processing and speech related requests from the *Phaser Game Implementation*. Requests are made through an API Singleton that is constructed on game load. These include requests such as getting the next practice word for a particular speech problem, setting which speech problem the child is working on, and harvesting microphone data. Harvesting and processing microphone levels is a multithreaded process implemented with HTML5 Web Workers. When the application is started and the API Singleton is constructed, audio stream *consumers* are created to process future audio streams in seperate threads. Once a microphone "Start" request is made from the *Phaser Game Implementation*, the following steps are executed:

1. Web Workers are initalized to start processing the audio stream in parallel.
2. Each stream buffer is processed by each Web Worker in parallel as the microphone streams raw Left-Right buffers.
3. Once a microphone "Stop" request is made from the *Phaser Game Implementation*, the consumers are queried for final results of their processing.

Currently there are two *consumers*: an *AudioStorageConsumer*, and a *RecognizerConsumer*. The *AudioStorageConsumer* simply stores the raw microphone levels, while the *RecognizerConsumer* communicates with the *CMU Sphinx* to iterate a best guess for the correctness of the child's speech.

4.1.4 CMU Sphinx

CMU Sphinx is a speech processing library [6]. There is a JavaScript port that allows audio recognition in the browser. Our *Speech Processing* component utilizes this JavaScript port to do audio recognition in the browser without server assistance. To deduce correct child speech we utilize this library creating target word and invalid word pairs. For example, when targeting 'R' sounds, we can use CMU Sphinx to recognize the target word 'Rabbit' or a invalid word 'Wabbit.' With a large dictionary of words this has been effective in distinguishing replacements caused by speech disorders.

4.1.5 Server API

The *Server API* is what connects the *Game* to our central recording server. Once audio is processed by the *Speech Processing*, it is sent off via POST request in raw microphone form to the central server if a internet connection is present. A unique identifier is computed and stored by the Kimbee System that will uniquely identify recordings in

the central server as belonging to a particular child's device. This unique identifier is accessible via the Options Screen in the *Phaser Game Implementation*.

4.2 Central Server

The *Central Server*, see Figure 2, faciliates communication between all the *Game* instances and the *Therapist Portal*. It is written in Node JS and stores data in a MongoDB instance. We also do audio encoding to create playable audio files from child microphone data obtained from *Game* instances.

4.2.1 Data Models

The server has three data models: recordings, children, and therapists. Recordings are the microphone encodings and recognition data from the *Game* instances. They are tied to a particular game instance via a unique token generated by the *Game*. Children are models created by therapists to effectively "hook into" recordings. They specify a name for the child and their token obtained from their *Game* instance. Future recordings are then relayed to the therapist from this particular *Game* instance. Finally, therapists are the authenticated user records that contain a user email and password and provides access into the therapist portal.

4.2.2 Audio Encoding & Recording Storage

When a audio file is recieved by a *Game* instance, we encode it into a WAV file by interleaving the left and right buffer channels from the microphone. This file is then compressed and stored into the database as a recording. The recognition data (whether the *Game* instance thinks the recording was correct) and the target word are also stored with the recording model.

4.2.3 Therapist Portal API

The server exposes a set of APIs for the *Therapist Portal* to utilize. This includes functionality to retrieve recordings for a particular child owned by the therapist record, retrieve the authenticated therapist record, authenticate therapists, and register therapists. This functionality is used via a HTTP interface with the *Central Server*.

4.3 Therapist Portal

The *Therapist Portal*, see Figure 2, is the application the therapist will use to track his or her student's progress and listen to the student's audio recordings. It is written in Javascript using the jQuery library with Bootstrap for the design elements and look of the page.

4.3.1 Connection to Child Audio

When the portal is launched, the data for all of the children under the therapist is downloaded through a HTTP GET request to the central server. The data is returned as a JSON object a short amount of time after the page is loaded. The WAV files for each recording are not downloaded at this time because of the significant size of each file. Instead, each recording is lazy instantiated and only requested when the user clicks play. A unique ID represents each audio recording and if clicked, a request will be made to the central server for the WAV file of the recording. When this request returns with the audio, the data is injected into the DOM through a HTML audio tag and then played.

4.3.2 Caching System

Before any request for a recording is made, a cache is first checked to see if the recording is already on the client. If a cache miss occurs, then the request will be made. If a cache hit occurs, then the audio will be immediately injected into the DOM and played. This system allows the therapist to replay the same audio recording multiple times without having to make a significant number of requests. A system for pre-fetching was discussed but never implemented; we leave this as future work.

5. RESULTS

Upon the completion of *Kimbee*, we met with Kreider in order to assess how a professional speech therapist would respond to the application and to see if he had any suggestions and improvements [5]. His reaction was very positive. He found no real flaws with the application and expressed that it would be very useful for children to work with. He wants us to continue consulting and working with him in the future as he thought that *Kimbee* was very promising. He did have suggestions for the future that are listed below under section 6. Unfortunately we did not have time to test our application with actual children and track the results but we are confident that *Kimbee* can be used in a real world setting.

6. FUTURE WORK

There are a few ways that we can expand on *Kimbee*. We plan on deploying the project for use in the Arlington Public School District to see how children respond and learn from *Kimbee* along with how experienced speech therapists use *Kimbee* to aid the therapy process. Along with this, we can also expand on the application itself by targeting speech sounds past the 'R' and 'W' sound replacement. We can also enhance the child's experience by adding more minigames for the child to play and a token system for children to spend the honey they collect on aesthetics for the bee character. The token and currency system described at the end of section 3.2 is also left as future work.

7. CONCLUSION

Kimbee is an application which uses speech recognition to help children and speech therapists better remedy speech impediments. While it currently only focuses on 'R' to 'W' sounds, it can be further expanded to supplement therapy by giving a therapist the recordings of a child in a remote setting and by giving a child the opportunity to practice more than the short time period allotted weekly. By giving children an enjoyable speech therapy tool we hope to save the money of both schools and parents while simultaneously giving children a better education by helping them overcome their speech impediment as quickly as possible.

8. ACKNOWLEDGMENTS

We would like to thank Richard Kreider of the Arlington Public School District for meeting with us to discuss the viability and details of our project when it was in its developing stages and then again to discuss improvements when Kimbee was almost finished [4, 5]. We would also like to thank Bruce Hemingway and Hanchuan Li for helping us throughout the project with guidance and feedback.

9. REFERENCES

[1] C. Bowen. *Children's speech sound disorders*. John Wiley & Sons, 2009.

[2] B. Dodd. *Differential diagnosis and treatment of children with speech disorder*. John Wiley & Sons, 2013.

[3] L. Ermi and F. Mäyrä. Fundamental components of the gameplay experience: Analysing immersion. *Worlds in play: International perspectives on digital games research*, 37, 2005.

[4] R. Kreider. Personal interview, January 2015.

[5] R. Kreider. Personal interview, March 2015.

[6] P. Lamere, P. Kwok, E. Gouvea, B. Raj, R. Singh, W. Walker, M. Warmuth, and P. Wolf. The cmu sphinx-4 speech recognition system. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong*, volume 1, pages 2–5. Citeseer, 2003.

[7] A.-M. Öster, D. House, A. Protopapas, and A. Hatzis. Presentation of a new eu project for speech therapy: Olp (ortho-logo-paedia). In *Proceedings of the XV Swedish Phonetics Conference (Fonetik 2002)*, pages 29–31, 2002.

[8] J. C. Rosenbek, M. L. Lemme, M. B. Ahern, E. H. Harris, and R. T. Wertz. A treatment for apraxia of speech in adults. *Journal of Speech and Hearing Disorders*, 38(4):462–472, 1973.

[9] J. Solari. ipad apps to use in speech-language therapy sessions. `http://consonantlyspeaking.com/`, March 2015.

[10] C. Vaquero, O. Saz, E. Lleida, J. Marcos, C. Canalís, and C. P. de Educación. Vocaliza: An application for computer-aided speech therapy in spanish language. *Proc. of IV Jornadas en Tecnología del Habla, Zaragoza, Spain*, pages 321–326, 2006.