UNIVERSITY OF CALIFORNIA
SANTA CRUZ

DEVELOPMENT AND EVALUATION OF SOFTWARE TOOLS FOR SPEECH THERAPY

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

by

Zachary Rubin

March 2017

The Dissertation of Zachary Rubin
is approved:

Sri Kurniawan, Chair

Travis Tollefson

Mircea Teodorescu

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Development and Evaluation of Software Tools for Speech Therapy

by

Zachary Rubin

Speech-language pathologists (SLPs) provide speech therapy services to children and adults, improving all characteristics related to speech and oral functioning. SLPs can correct the most severe of impairments in less than three years, assuming the patient is motivated. Unfortunately SLPs can only guarantee motivation during weekly meetings that last one-hour. Outside of the office patients increasingly depend on their friends and family for encouragement and support. As therapy slows due to factors outside the SLP's control, patients risk joining the two-thirds of people who drop out of speech therapy.

In this thesis I discuss the development and evaluation of a suite of tools intended to address the dearth of technology plaguing modern speech therapy. Applying a user-centered design approach I identify the primary stakeholders as the SLP and the patient, putting the support network of family and friends as secondary. Using this perspective I uncover the absent elements and develop tools to demonstrate both the value of each element along with the benefits of improved access. These tools introduce three novel technologies to speech therapy: accurate speech evaluation via automatic speech recognition (ASR), gamification of speech therapy exercises, and big data analysis for complex ASR output. When combined they tackle the primary bottleneck in speech therapy today: what occurs outside of the office.

In the first section, I collaborate with doctors at the University of California, Davis Medical Center. Together we identify the primary problem in speech therapy as progress outside of the office. We both agree that a properly calibrated ASR system would provide more consistent grading than support network members. Using this information I develop a speech recognition system on a mobile device that someone can tune to virtually any speech impairment. I tune the system for a lateral lisp and evaluate the system with an undergraduate student at the University of California, Santa Cruz. The system meets our gold standard by producing a 95% accuracy rate [32, 59].

In the second section I embed the aforementioned ASR system into a game engine with the goal of producing motivating speech exercises. Through investigations with therapists

and observations with children, I build a system that uses an interactive storybook game to automatically evaluate the speech characteristics of children with cleft palate. The system is successfully deployed in a clinical setting to children visiting a therapist for speech therapy evaluation. The system correctly evaluates 8 out of 9 children on par with a SLP, demonstrating ASR as a reliable speech evaluator. Interviews confirm that we have successfully created an experience that is as motivating as working with a live SLP in the short term [59, 27, 50, 49].

In the third and final section I add big data analysis to the existing toolset in order to provide therapists effective feedback. I design with the goal of maximizing data visualization while minimizing mental and temporal demands. This results in a statistics engine that affords the therapist quantitative metrics at a glance. The speech recognition engine is modified to process speech mid-sentence, enabling real-time interactions. As this permits us to increase the rate of speech feedback, I adjust the game design from a storybook style game into a cycling series of mini-games in order to maximize speech rates in game. The system is distributed to therapists at UC Davis, and evaluated with various adult communities in Santa Cruz. We receive positive feedback from therapists on all aspects; with the tool providing both increased motivation in-office and increased visibility of tasks outside of the office. On the patient side of the work I effectively motivate 60% of adult speech therapy dropouts through the game environment. One participant demonstrates steady speech improvement through usage of the game. The SLP confirms that the improvement transfers to real speech scenarios, thus validating the design.

To my parental units, for their boundless love and genetic material

To my older sister, Jamie, for reminding me to strive further

And to my failures, for their endless education.

Acknowledgments

First and Foremost I would like to thank my professor Sri Kurniawan whose intelligence, kindness, and diplomatic prowess made all of this possible.

Thank you to Travis Tollefson and Mircea Teodorescu for serving on my dissertation committee, and Patrick Mantey for serving on my advancement committee.

Throughout my graduate career this work has involved many students, reseachers, doctors, and artists. Thank you to undergraduate and graduate students Cédric Foucault, Taylor Gotfrid, Annie Pugliese, Michael Weber, Maya Bobrovitch, Dylan Gardner, and John Chambers. Thank you to all of the staff at University of California, Davis: Christy Roth, Sandra Sulprizio, Susan Goodrich, Jamie Funamura, Erin Hubbard. Without your help this wouldn't have been possible. A thank you to Alexandra Dunham for providing the initial art direction in the game.

A special thank you to all of my English teachers; turns out you write a lot as a researcher. Craig Phimister and Henry deMauriac provided the greatest influence on my capabilities as a writer, with Craig establishing how to become an effective writer, and Henry encouraging me to embrace creativity and expressiveness while keeping me on course.

Chapter 1

Introduction

Speech-language pathology focuses on correcting and improving characteristics related to speech production, though Speech-language pathologists (SLPs) also provide assistance in swallowing as well as broader communication aspects. While adults utilize SLP services - particularly those who experience a stroke, Alzheimer's, or Parkinson's - most SLP services are directed towards children. Children are provided access to SLP services in the public school systems of most developed nations including the United States, New Zealand, and the United Kingdom. Unfortunately, rural areas as well as developing nations suffer from a dearth of resources.

Cleft Lip and Palate is one of the most common congenital birth defects in the world, with about 20 infants born with an orofacial cleft on a given day. While surgery to correct clefts can begin as early as six or nine months, it is rarely performed this early and is often peppered with up to 20 years of additional surgery. Along the way patients may receive up to three years of speech therapy, but fight an uphill battle as 65% of children drop out of speech therapy [31, 25]. For the children that enroll and stay, they are subjected to decreasing amounts of one-on-one time with a professional SLP as they contribute to a 7% annual caseload increase [4]. With over 52% of regions in the United States reporting a shortage of trained SLPs, many do not even receive the opportunity for speech therapy.

Automatic speech recognition (ASR) has achieved near ubiquity in the past 20 years for medical dictation purposes. A study in 1998 replaced traditional dictation with computerized speech recognition for a 3 month period. The system was used in 87% of cases, leading to a 99% decrease in the amount of time until a report became available for other doctors to use

and dropping the overall cost of dictation. Since then most hospitals in the United States make use of an ASR tool for medical dictation [48]. Despite these achievements, the medical field remains absent of other tools that utilize speech recognition. In the realm of speech-language pathology, interactions between SLPs and patients have remained essentially static since the 1980s. Telepractice, or providing speech therapy remotely through internet-based teleconferencing tools, has enabled SLPs to work with clients and patients in rural areas lacking public school services as well as undeveloped nations. An estimated 74.9% of SLP professionals in the United States do not use telepractice, and computers are used primarily for clerical work like patient data entry and scheduling [4].

Researchers in 2012 reviewed computer-assisted speech therapy's benefits and drawbacks by investigating a variety of existing research systems. It found four critical language therapy components: intensity, active attention, feedback and reward. Additionally, computerized speech therapy programs are focused on providing the client exercises while the speech therapist evaluates pronunciation. A major disadvantage results of the inherent difficulty of determining the type and degree of language disorder and in the severe limitation of monitoring the therapy progress and automated readjustment of practice exercises upon the phonological feedback provided by the subject. In other words, the results of the whole therapy depend essentially on the therapist's direct expertise, including variables such as therapist's agenda, his or her professional experience, and effective intervention time for each client, therapy group size, session's frequency and so on. They noted that the software should be used as an assistive tool and not as a substitute for the SLP, but both groups should use it as a relief from repetitive tasks [42].

1.1 Cleft Palate

The first two years of a newborns life are a crucial period of time in the development of speech and language. Within the first year, a child is using his or her mouth to form a variety of phonetic structures and is saying their first words. The second year, toddlers are forming basic sentences [3]. Children born with craniofacial defects such as cleft palate are unable to form a variety of structures, using substitutions and compensations that result in speech problems [22]. These can take years to correct and can lag speech development.

Approximately 1 case of orofacial cleft occurs in every 500-550 births. In the United

States, 20 infants are born with an orofacial cleft on an average day, or 7500 every year. Surgery to repair the cleft can be performed within the first year of birth. However, speech therapy to correct impairments often continues until the child is five or six [31]. Children who have an orofacial cleft require several surgical procedures, complex medical treatments and long-term speech therapy; the estimated lifetime medical cost for each child with an orofacial cleft is $100,000, amounting to $750 million for all children with orofacial cleft born each year in the United States.

Figure 1.1: Example of a Cleft Lip

Orofacial clefts (i.e., cleft lip [CL], cleft lip and palate [CLP], and isolated cleft palate [CP], as well as the rare median, lateral [transversal], and oblique facial clefts) are among the most common congenital anomalies. Corrective surgery is commonly performed around 10-12 months of age with the goal of providing a more normal anatomical framework by the time children begin practicing speech [3]. The repaired palate continues, however, to be variably impaired by the less-than-normal muscle bulk typical of cleft palates and by the stiffness of normal

post-surgical scar tissue. Surgical repair impairs facial growth, so one year is the compromise age for repair used by most plastic surgeons. Over perhaps one year following surgical repair, palatal function spontaneously improves to the point that in the majority of children it is adequate to produce "velopharyngeal adequacy," i.e., it is able to selectively prevent nasal escape. The remaining children undergo additional reconstructive surgery, commonly around the age of 5 years. Speech therapy after surgery to correct CAD begins at the age of two years and often continues for many years [22].

With a cleft of the palate, one is unable to stop airflow through the nose using normal mechanisms. Therefore, cleft palate speech contains a number of words with air leaking out the nose when it should not, referred to as "nasal escape." In the world of trial-and-error that governs speech learning, the child uses the only tool he or she has available to keep air from escaping out the nose; he or she holds it back at the level of the glottis or the larynx. This mechanism is used by the normal voice to make a hard 'g' as in 'go.' The child uses this "glottal stop" as a substitute for a variety of sounds that cannot be normally created. Cleft palate speech thus becomes a collection of sounds characterized by glottal stops and inappropriate nasal escape, known as compensatory speech structure [31]. These anomalous articulation patterns are usually referred as compensatory articulation disorder (CAD). This disorder severely affects speech intelligibility.

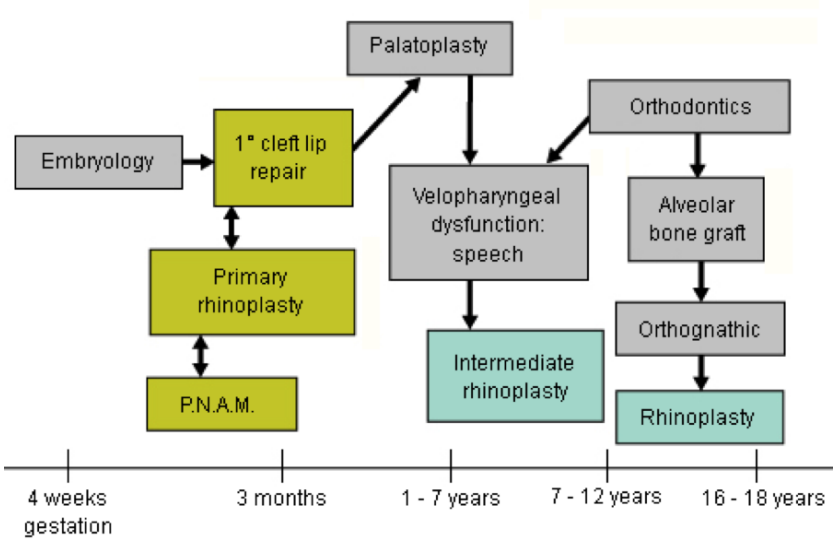

Figure 1.2: Cleft Palate Treatment Schedule

Speech therapy is usually initiated anywhere from 20 months to 2 years of age. At this age SLPs are directed to instruct parents to become 'therapists' at home. This is because

parents will naturally become the child's first language instructor. Young children between 3 and 5 years are very receptive to learning new language skills, but parents are still encouraged to become therapists at home. The prognosis for normal speech past 5 years of age is guarded as the critical period of speech development has passed; bad speech patterns have become ingrained habits [29] [26] [31].

Correcting cleft speech is crucial for a child's future ability to live independently and to participate fully in society. Hunt et al. assessed the psychological functioning of patients with CLP, comparing 160 children and young adults with an age- and gender-matched control group [21]. The researchers found that children and young adults with CLP reported greater behavioral problems and increased symptoms of depression and were more likely to experience bullying in social settings. Another study showed that those with CLP registered higher scores for social problems and deficits in social and academic competencies [52]. Finally, even with corrective surgery, they married later, displayed a delay in scholarship, had a lower income, and reported a significant delay in their independence process from their parents [12].

SLPs instruct parents to perform speech practice with their children for at least 5 minutes per day [56]. The precursors to speech learning are motivation, focused attention, and adequate training before entering the practice phase. The conditions of practice are motivation, goal setting, effective instructions, effective modelling (e.g. production characteristics), and setting. Parents are instructed to perform repetitive motor drills There are four types of practice schedules in therapy: massed, distributed, blocked, and random. The first parameter relates to date scheduling: massed practice involves fewer sessions, but longer sessions while distributed practice favors the same duration but shorter and more frequent sessions. The second parameter involves the order of presentation in a practice session: blocked practice involves focusing on one stimulus until progress is made, while random practice randomizes the stimuli. While SLPs intend for parents to choose the best practice schedule for their child, in the real world parents may not have much choice regarding practice distribution. Negative feedback has been historically and famously shown to produce disastrous results [60]. As a result, parents are exclusively instructed on how to provide positive feedback. Parents must learn how to deliver reinforcement, provide explicit modelling, and provide effective positive feedback. It is necessary to choose and develop appealing activities for the child, to provide input without insisting on a response, and to not have rewards take up too much time. Establishing beneficial behaviors and facilitators for the patient are key [6].

Despite the documented benefit of correcting cleft speech, it remains challenging for speech pathologists to guide children to remedy their compensatory speech and to train them in proper speech production. Earlier intervention correlates with a higher success rate, but decreasing age also correlates with less cooperation. This intereferes with all aspects of therapy: paying attention to the therapist, comprehending instructions, and compliance efforts. The speech pathologist must have an ear trained to hear the distortions of speech and also must effectively motivate their patients. Once the therapist successfully coaches the child to produce the desired sound in the therapist's office, they return to their home and are asked to perform unrewarding speech homework. Children typically practice under the guidance of parents who are not speech pathologists and therefore often unable to assess subtle progress (or lack of). As a result, many children have varying degrees of resistance to such work, and the cause for their lack of progress remains undiscovered until the next visit to the therapist's office.

1.2 Speech Recognition

Most modern speech recognition engines utilize two algorithms: Fourier transforms and Hidden Markov Models (HMM), though some now exploit neural networks to produce faster and more accurate results on smaller training sets. Speech samples are analyzed in over small time intervals called windows. When speech is analyzed in 10ms window individual speech sounds approach a stationary process, permitting feature extraction. This property means the Fourier analysis of a phoneme will be the same regardless of when the analysis begins along the waveform.

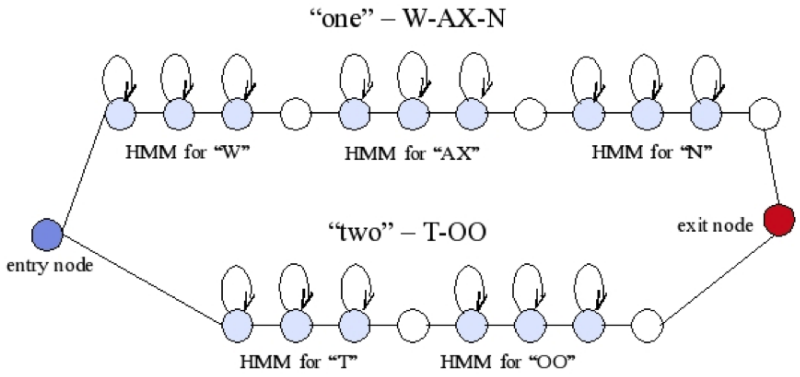

Figure 1.3: Example of an HMM Processing The Words One and Two

ASR is composed of three models: an acoustical model, a dictionary (word model), and a language model. The acoustical model contains statistical representations of the sounds that make up the phonemes in words. This file is not editable, and the only way to add additional data is to adapt the model. The dictionary contains translations from phoneme to word. This file is the easiest of the three to edit as its structure can immediately be understood. The language model is the last Markov model and turns words into groups of two and three words to check for coherency. One can also hand-edit this, but as the probability weights are included in the file and generated by a transcription file beforehand the words added in may appear too often or not often enough.

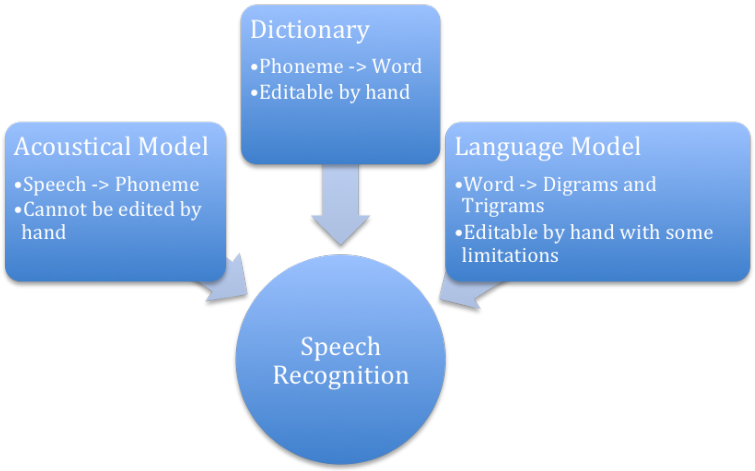

Figure 1.4: Models in an ASR system and their functions

Developing a new acoustical model requires wave files of speakers and the transcriptions of the files. Training requires one hour of recording for command and control with a single speaker before acceptable performance begins to develop. For many speakers up to 5 hours each from 200 speakers are needed, or 1000 hours of recording. Dictation is an order of magnitude more difficult, requiring 10 hours for one speaker or 10000 hours for multiple speakers (50 hours from 200 speakers) [9]. The accuracy and speed of speech recognition are functions of the three models. Large dictionaries or language models will increase lookup time and decrease accuracy, while large acoustical models increase accuracy and lookup time but on a much greater magnitude than dictionaries.

Dictionaries (or word models) take the output from the acoustic model and translate them into words. Dictionaries are created by feeding in a text file or corpus, which takes individual words and converts them into a phonetic equivalent known as ARPAbet. ARPAbet

is a phonetic transcription code in which every phoneme in General American English are represented by one or two capital letters. The code HH represents the sound in 'house' or the international phonetic alphabet (IPA) representation /h/.

The language model takes words composed from the dictionary and ensures that the words result in a reasonable sentence. The model is composed from the same corpus as the dictionary, but analyzed into Markov chains of Bigrams and Trigrams. The model also includes weights representing the probability of different word groups to occur.

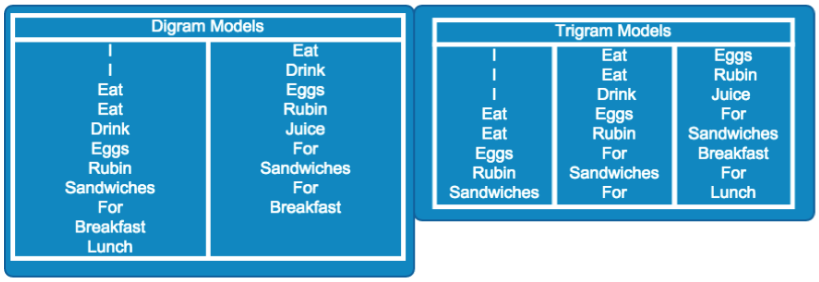

Figure 1.5: Examples of Bigram and Trigram Entries In a Language Model

A recent development in speech recognition is the ability to offload speech samples to an external server for significantly improved processing while maintaining a reasonable turnaround time. This allows mobile devices to generate much more accurate results that would not be feasible in a reasonable amount of time on the device's hardware. Personal assistants like Google Voice, Siri, and Amazon Alexa notably do this when a user asks a question that involves a web query or has too many out of vocabulary words.

Chapter 2

Proof of Concept: Mobile Mispronunciation Recognition System

The first tool investigated the capabilities of speech recognition on mobile devices. This came as the best direction to start based on interviews with SLPs and other craniofacial specialists, assessment of technological goals, and gauging development time. The discussions with doctors and professionals indicated that a way to remotely collect speech data from patients as the gateway component to a number of other important capabilities. Developing this would open the most doors into feasible user studies. Much of the initially developed system's direction depended on what I could quickly use and avoiding potential issues as the research progressed.

During the brainstorming and discussion sessions with doctors the ideas gravitated towards a system usable by groups in third-world countries, with the idea of a charity group such as Operation Smile performing surgeries and then dropping off devices for the patients to perform their speech therapy remotely. Further discussions in this area emphasized a need to enhance - but not replace - existing services provided by SLPs.

Though the other tools were designed for use both by the therapist as well as the young patient, I designed this tool as a demonstration for use by a therapist or a researcher. Interactions with young patients would not occur until later after approval from the Institutional Review Board. Since there was a large interest in having mobile games, the response time of the speech recognition system was a key factor in deciding the next iteration of tools.

2.1 Related Work

Initial work in the area of mispronunciation detection specifically for speech impairments such as cleft palate began in 2006, when researchers at the University of Erlangen-Nurnberg compared a highly customized automatic speech recognition system to expert SLP analysis for evaluation of cleft lip and palate. Researchers developed an acoustical model using data from the VERBMOBIL project which recorded the speech data of non-pathologic children voices across Germany [62]. They then integrated the acoustical model into a customized version of a commercial speech recognition engine and collected speech samples from 31 children totaling 120 minutes. Results from the automatic speech recognition's evaluation of the word accuracy closely matched a Likert scale evaluation of 3 SLP experts [13, 14, 15, 44]. Their research most closely relates to the tool developed. However, their system uses a number of parts unavailable to us. Their group acquired a commercial engine and customized it, while we investigated both open source and commercial solutions. There remains no acoustical model developed for English speaking children, though some commercial groups have begun work on this. Their work emphasized automated diagnosis of speech impairments and provided three different ways to analyze the resulting speech. It had the capability to check for hypernasality as well as phonetic issues. The drawback to this is the increased difficulty of implementing this as a real-time input method for games. Our system had real-time considerations, so I opted for speed and worked only on phonetics.

Mispronunciation detection is of interest in mobile assisted language learning. Google hosted a project for mispronunciation detection during their 2012 Summer of Code. Their system proposed using acoustical alignment scores; matching a given sentence against a gold standard [46]. A project determined to do pronunciation evaluation over the web with a game-based interface was also accepted into the 2012 Google Summer of Code. Their projects differ from this one in the fact that they could collect copious amounts of speech data, while we cannot use any collected speech data. The projects concluded with a web portal being put online for speech collection as well as the potential integration of Amazon's Mechanical Turk [47]. However, the work did not continue as no recent work has appeared.

A group of researchers in China developed an evaluator for cleft palate speech intelligibility. They decided to start with hypernasality, building custom ASR systems using Mel frequency cepstral coefficients and Gaussian mixture models. Testing their work on a database

of 240 spoken words, 60 words each in 4 different levels of hypernasality, they achieved a classification accuracy of 79% in their first test. In their second test they collaborated with the Department of Cleft Lip and Palate at Sichuan University, giving them access to a CP speech database containing recordings of 120 patients and a total of 10080 words. They achieve classification rates of 83% for hypernasality and 94% for consonant omission. Like the work done in Germany, this group's tool Their work occurred after this tool had been developed, and was not known until later. [18, 19, 20]

2.2 Requirements Gathering

The first step in requirement gathering was to interview several cleft speech pathologists working for the Cleft and Craniofacial Reconstruction team at the University of Caifornia, Davis Medical Center to gain an understanding of their requirements when assessing as well as prescribing in-home exercises for children with corrected cleft. While assessment practices can vary, in general, these children are administered the Sounds in Words subtest of the Goldman Fristoe Test of Articulation (GFTA). In addition, samples of spontaneous speech of these children communicating with their parents, siblings or speech pathologist are collected using high-fidelity audio recorder. In addition, a speech pathologist with trained ears will ask the child to speak a few words and sentences to detect their compensatory speech. Usually the child is asked to say the same words/sentences three times. The sentences are in general aimed at detecting the following symptoms:

Symptoms and Evaluation Sentences	
Hypernasality on vowels Mixed vowel	He had two rock lizards.
Hypernasality on vowels High Front /i, I/	Bill sees the sleepy kid.
Hypernasality on vowels High Back: /u, U/	Sue took the old blue shoes.
Hypernasality on vowels Low Back: /a, o/	Father got all four cards.
Nasal emission on pressure consonants: p	Pet the purple pup.
Nasal emission on pressure consonants: t	Take Ted to town.
Nasal emission on pressure consonants: k	Cut the cake!
Compensatory articulatory gestures, such a glottal stops	It's a pretty little kitten.
Compensatory articulatory gestures	What a lot of little bottles!

Table 2.1: Sentences used by SLPs for diagnosing CP-related speech impairments

The speech pathologists we interviewed also mentioned two main problems with the current practice of cleft speech therapy. The first is the fact that once the child has been coached

to produce the correct speech in the therapist's office, they return to their home and are asked to do boring speech homework, typically under the guidance of parents who are not speech pathologists and therefore often unable to assess subtle progress (or lack of). Many children have varying degrees of resistance to such work, and they progress slowly and ineffectively because their lack of progress is undetected by the inexperienced parents until the next visit to the therapist's office. Secondly, school and community based speech therapy is limited to one-or-two times per week for less than 30 minutes. Delivery of school speech services has further been reduced due to budget cut and limited to only group sessions due to high caseloads of speech therapist working for the schools. Individual therapy is performed infrequently and therefore individual needs of children are often unsatisfied.

Two major complications that arose were the need for compliance with the U.S. Food and Drug Administration (FDA) and the need for patient privacy. At the time of development, the FDA released new guidelines regarding apps and what they would consider apps that turned the device into a 'medical device'. This designation would hamper work as serious user testing would need FDA approval of the system [2]. Fortunately, these new guidelines did not impact the development or testing of any tools.

Addressing patient privacy issues became a major design requirement for this tool. After additional discussions with doctors and SLPs from the craniofacial specialist team, we agreed that speech recordings on the device was considered patient data. Consequently we added the requirement that the system would discard sound recordings after processing and analysis. We also decided that the development of an acoustical model from children's voices was too time consuming and too invasive. From this we determined the requirements for the ASR engine.

Requirement	Priority
Compensatory structure or hypernasality recognition rate of at least 95%	High
Recognition responds fast enough for use as gameplay input	High
System does not require an internet connection	High
Recognition does not require custom acoustical models	High
System must fit on a device small enough for a child to carry	High
System functions in noisy environments	Medium
System is automated and does not require a therapist	Medium
System stores ratings securely	Medium
Recognition of both compensatory structures and hypernasality	Low
Recognition scores and ratings archived on device	Low

Table 2.2: Proof of Concept requirements

2.3 Development

In the step of the development process we selected a suitable speech recognition engine to use as a base. We selected recognition engines from a variety of commercial and open-source locations. Commercial providers were contacted to determine their software's capability and their interest in providing the engine for modification and evaluation.

Software	Dictionary	Language Model	Acoustic Model	Has API	Works on Mobile
Dragon					
Rosetta Stone					
SIRI					✓
Kinect	✓		✓	✓	
PocketSPHINX	✓	✓	✓	✓	✓
Julius	✓	✓	✓	✓	✓

Table 2.3: Comparison of Speech Recognition Engines

A minimum of 20 hours of speech recordings from various children with cleft palate were needed to achieve any usable accuracy. Along with this, the acoustical model's size has the largest effect on recognition processing time [9]. Though development of an acoustic model was infeasible, we determined that acoustic model adaptation was an extremely useful capability. Acoustic model adaptation takes an existing acoustic model and integrates speech recordings and their transcription. This permits the system to adapt to a variety of factors including speaker, microphone, davice, and listening environment. As adaptation requires access to the acoustic model, this was considered a beneficial feature [8].

In finding a suitable recognition we first investigated commercial and proprietary options. Nuance, the makers of Dragon: Naturally Speaking did not have a suitable portable system developed. Rosetta Stone's language learning software has a system capable of evaluating pronunciation and accent, but was unsuited for speech therapy purposes. SIRI requires a mobile connection to upload most speech files and did not have a way of creating or editing dictionaries. Kinect provided an API, but the need for connection to a laptop or game system made the system nonportable.

We could not find a commercial solution that met our needs, so we turned to the open source community. There we found two tools: Julius and PocketSPHINX. Julius uses word 3-gram and context-dependent HMMs to achieve over 90% accuracy during real-time processing with a 20,000 word dictionary. The acoustic model, dictionary, and language model are all

editable, and base ones are provided for each. Their system works in Linux-based environments, and works on low-end hardware [35, 34]. Though Julius would work on Android-based devices, the lack of a mobile framework made it too time-consuming to develop.

PocketSPHINX is an automatic speech recognition system developed by researchers at Carnegie-Mellon University. Developed as an optimized port of the SPHINX-II engine [36, 41, 28]. The acoustic model, dictionary, and language model are all editable, and base ones are provided for each. Researchers tested the system on a 206MHz handheld device and achieved a 10% word-error rate for a 1,000 word dictionary while maintaining response in real-time [30, 61]. PocketSPHINX could quickly be deployed onto Android devices, but a framework already existed for iOS devices under the name OpenEars. OpenEars performs all speech recognition on the device, and does not need to offload any data to an external server. Additionally, it can swap out both acoustical models and dictionaries on the fly to improve accuracy without penalty. We selected OpenEars as our recognition framework.

We implemented impairment detection through the use of dicionary approximations, as they provided the highest degree of configurability for the least amount of effort. Development into accurate speech impairment dictionaries started by attempting to detect plosives, as it is a common characteristic of cleft speech that requires speech therapy after initial surgery. Plosives comprise the b,d,p,t, and other sounds that involve a small expulsion of air from the mouth to produce. Various dictionary ideas were attempted before one producing an acceptable accuracy was found. First an experiment was run with a dictionary containing a 1:1 transcription of the phonemes, but found that phoneme-for-phoneme accuracy of the recognition engine barely achieved 50% accuracy in the best of conditions. This poor accuracy rendered raw phoneme recognition unusable as a stand-alone solution. Individual word dictionaries were attempted next with phonetic approximations of the mispronounced words, but recognition remained far too low to determine if there were mispronunciations. Finally, sentences were constructed with mispronouncable words towards the end of the sentence. This provided enough accuracy to optimize and develop a proof of concept that could be used for testing.

2.4 Lateral Lisp Test

We embedded the ASR system in an iPhone app as a proof of concept. The system currently uses a 5-megabyte acoustical model trained on the Wall Street Journal to keep the

Sentence	Target Word	Arpabet
Place your bets	Bets	B EH T S
Place your bets	Bets (Lateral)	B EH T SH
Cats drink milk	Cats	K AE T S
Cats drink milk	Cats (Lateral)	K AE T SH
I eat oats	Oats	OW T S
I eat oats	Oats (Lateral)	OW T SH

Table 2.4: Dictionary to Detect a Lateral Lisp

system as quick as possible and real-enough time for older devices. Using the UI shown in 2.1, it displays the text of what the recognition system heard.

We first tested the system's recognition time on different devices and different speakers. We recruited a graduate students on the campus of the University of California, Santa Cruz and tested the system's response time and accuracy. We tested response time across three different devices: a second-generation iPod Touch, an iPhone 4s, and a first-generation iPad Mini. The iPhone 4s and iPad Mini were recently released, while the iPod Touch was two years older and had approximately half the computing capability of the other two devices. Accuracy for all devices were 90% for 10 sentences, though response times differed. We selected to use the iPod Touch instead of the newer devices in order to maximize our hardware availability and determine if its response time was deemed too slow.

Device	Average Response Time
iPod Touch	1.2s
iPhone 4s	0.5s
iPad Mini	0.4s

Table 2.5: Recognition time for devices investigated

A study was run with a student at the University of California, Santa Cruz with a lateral lisp as a participant. A /sh/ at the end of a hard /ts/ sound characterizes a lateral lisp. We developed a dictionary and language model containing one possible mispronunciation for each target word. 2.4 contains an example of the sentence, target words, and their APRAbet equivalent. We constructed a dictionary consistint of 10 sentences, and loaded them onto an iPod Touch to investigate accuracy as well as evaluate the speed. Initial tests with the dictionary were run with one speaker speaking both normally and imitating a lateral lisp. The system consistently discerned one speaker lisping and not lisping on 7 out of 10 sentences so the dictionary was pruned to a total of 51 words and 8 target words. Following this the participant

was invited to test the system.

In selecting a target recognition accuracy we consulted medical literature for accuracy rates clinicians considered 'high accuracy'. We determined that 95% accuracy is considered high accuracy [64], so that was selected for as the target recognition accuracy. A sentence was judged correct if the speaker, observer, and recognition system agreed that the target word was spoken with a lisp or not. A sentence was incorrect if both observer and speaker agreed, but the system disagreed on the target word.

Using a student with no speech impairments as the control, the two students spoke each of the sentences 20 times over a 1.5-hour period. Students alternated between speaking and observing after every run through of the sentences. The researchers hypothesized that the system would have consistent accuracy throughout, and that the repetition of the syllable in an untrained non-clinical setting would have no actual effect on speech quality for the speaker with the impairment. In other words, the system's accuracy should not change between the beginning and the end of the study, as no one is trained to give the student proper instruction on producing the correct sound. Normally, word error rate calculations take into account the number of omissions and insertions against the system but as the focus was on target words, non-target words did not affect the accuracy. The only exception would have been if the target word was omitted entirely, but this situation did not occur.

On the interface side the participant said that the speech recognition engine responded at an appropriate speed. The participant stated he desired a way for the system to provide verbal feedback, stating "If I could hear the system feed my speech back to me corrected I would be able to imitate those sounds." It is not known how to provide this feedback, as it would require an unusual modification of the speech recording. Accuracy did not vary during the test period, and the system did not alter the speech of the participant.

2.2 shows the accuracy results from the study. The system achieved the target accuracy for 2 of the 7 sentences. The remaining sentences were between 80-90% for non-lateral speech and 40-70% for lateral speech - too low for use with either group. Though these results seem discouraging the two sentences that did achieve a high accuracy, "I eat oats for breakfast" and "Place your bets", determined that the system could achieve a high accuracy in detecting a speech impairment in real-time using only a customized dictionary and language model. With this information we began developing the game section and the data section for a pilot test with children and therapists.

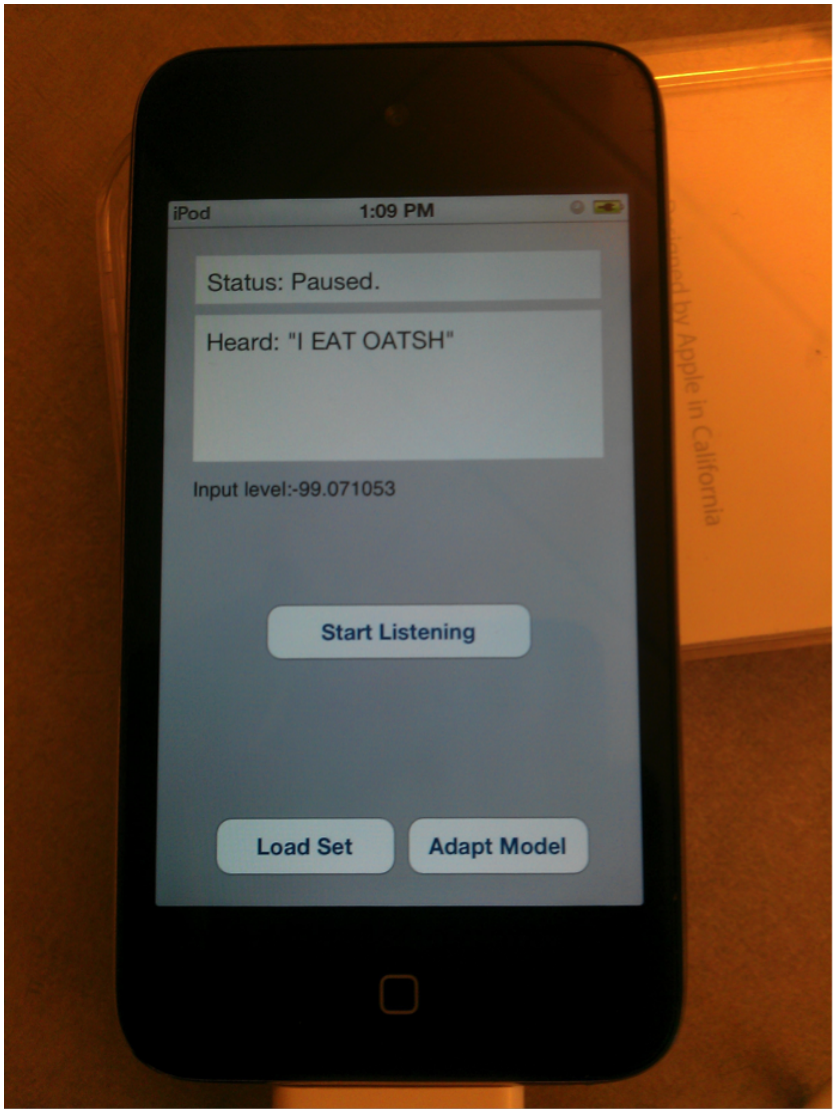

Figure 2.1: The proof of concept application on an iPod Touch

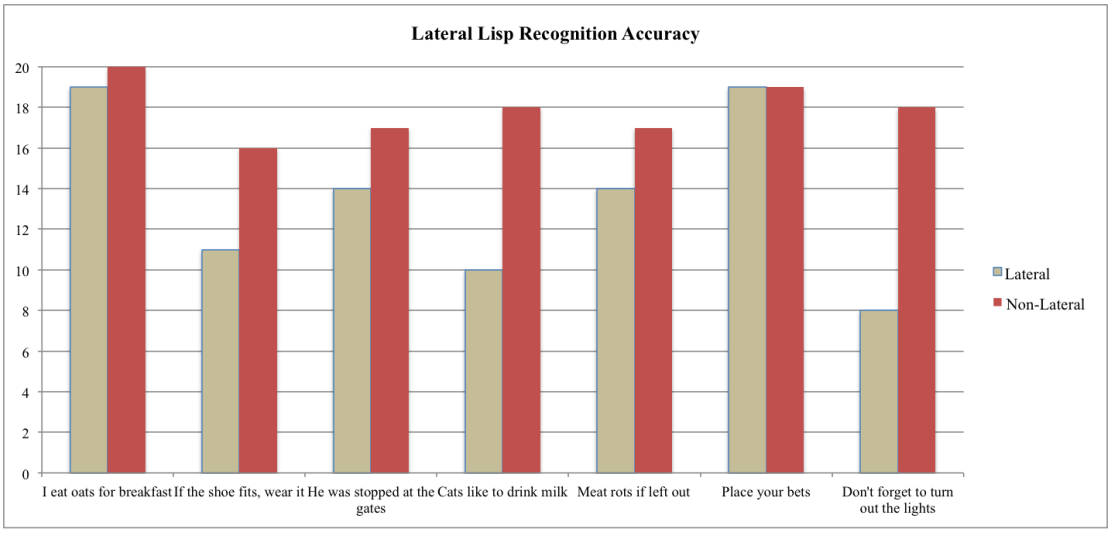

Figure 2.2: Lateral Lisp Results

Chapter 3

Pilot Test: Speech Adventure

We initiated the development of the other systems following the success of the speech recognition engine, this included the game engine and the statistics engine. I reconfigured the speech recognition engine to detect plosives and investigated different Arpabet transcriptions. Then we brought the system to the craniofacial specialist team where they tested the system with a few children. The results were promising and demonstrated that continued investigation would reap benefits.

Initially we were uncertain of the expected age that would play this game. In the ideal case surgery is performed at 6 to 9 months of age, with therapy beginning the following year. We observed young children in addition to observing young patients. At the conclusion we determined designing the game towards all ages was the best direction.

Determining the hardware we would use in testing served as a major factor in the development direction. Different acoustical models were investigated during this time to try and find a good speed/performance balance. Additionally, the tools were developed to scale to any screen size. Though this increased the difficulty of the programming, it allowed us to greatly expand our initial testing and evaluation process. It also allowed us to demonstrate the device live to various groups, generating interest in the project.

Deciding upon the test environment for the app served as another major challenge. Initially we designed the system for use in the waiting room of a doctor's office and collect all data anonymously. The game originally asked player's age, to determine if the player was a child or not, we ended up not using this data. All of this changed quickly when we gained approval to involve child participants in our study, allowing us to acquire real data. The results showed both

the functionality and the limitations of the systems. We would use this feedback to improve our systems in the final iteration.

Due to the speech recognition engine limiting our input speed to waiting for completed phrases, we selected a game style quickly. This benefitted us as we could start on game development early on, which permitted us to apply a large amount of polish towards the end. The user interface took a large amount of time in iterative development. Many different configurations of speech recognition analysis and resulting user interface behavior were tried before finding good combinations to evaluate.

As ethics and patient privacy were extreme concerns for our research, we ensured that researchers only received anonymized data stored on a secure server. However, receiving this data required an Internet connection. As the data file was uploaded immediately at the end of the play session and then deleted from the device, a loss of connection during that period meant potential loss of data. This happened with our first participant when we learned that the hospital did not have WiFi. We solved the problem by having the iPad tether to an iPhone device. This issue reinforced our requirement for always offline performance in future iterations.

3.1 Related Work

Project LISTEN is an automated reading tutor aimed at helping children learn pronunciation and proper speech when reading aloud. It does this by analyzing various aspects such as pitch, speed, and pauses. Researchers have tested the system in India for assisting children learning English as a second language, as well as in Canada for children looking to improve their speaking skills [17, 5]. Project LISTEN's system requires a desktop computer, but can work on lower-end systems. Primary schools in Ghana had no difficulties running Project LISTEN on 266 MHz systems with 64 MB of RAM. Specifications on modern mobile devices far exceed these computers. Mobile Assisted Language Learning (MALL) is a popular field of study. Project ALEX has proposed a very robust application for language learners. Project ALEX focuses on a large dictionary with text-to-speech functionality. Most importantly, Project ALEX included pronunciation practice and used speech recognition to check if the user says the work correctly using Microsoft SAPI [16].

Couse and Chen [8] investigated the usefulness of tablet PCs in early education. Children used drawing tools to make self-portraits, do freehand drawing, and practice writing. They

enlarged important icons on the screen and removed unnecessary icons and OS elements to simplify the system as much as possible for children. They found that children spent significantly more time on the task than conventional paper. Although children experienced some technical difficulties, none were discouraged enough to stop. Along with this, the majority of children preferred using the tablet to conventional pencil and paper.

A study put iPads in the hands of kindergarteners to look at the change in performance on standardized literacy tests such as the Rigby's Benchmark Assessment over a 9-week period [9]. The kindergarteners that used the iPads saw significantly greater increases in their scores, notably in the Hearing and Recording Sounds in Words subtest.

3.2 Requirements Gathering

User interviews consisted of three groups: extremely young children, craniofacial surgeons and SLPs, and child patients. The purpose of working with the extremely young children was to gain an understanding of their cognitive capabilities when interacting with tablets and mobile devices. In our interviews with therapists we investigated the instructions they give family members to assist in therapy, and with young patients we observed the toys they bring to the office.

This tool became the first one to directly involve children. A variety of early games that use speech recognition as their main form of input are investigated for common elements to draw inspiration from. The development split off with the therapists guiding the backend development while children guided the frontend. Both sides built upon the foundation of a speech recognition engine capable of detecting mispronunciations.

The investigation of video games takes a look at different video games. Speech recognition in commercial video games only became a reality in the 1990s, when both consumer electronics and speech recognition achieved enough performance to merit use as game input. Though the games generally received high ratings and reviews, most games did not gain mass appeal. Since this writing, it is very likely that there are many new video games that use speech recognition, and this list is by no means complete.

3.2.1 User Interviews

After receiving approval from the Institutional Review Boards, we directed user interface development towards the patient side of the application and focused the backend development on the therapists. We started with an observation of children participating in separate, but similar, research. Next we revisited our doctors and SLPs to investigate how their requirements may have changed since the completion of the recognition engine. Finally we passively observe children as doctors make their rounds to understand what motivates them and reduces the tedium of therapy.

Druin suggested a user-centered design method where children directly influence the development of software designed for them, called children as co-designer. This method has been shown to be successful in creating fun, motivating, and effective user interface for children, although no published manuscripts reported the use of this technique for therapy software for children. In this method, Druin proposed that children should be on equal ground with the developers and designers. In many software development projects, children themselves are rarely included in the design and creation of software for them due to the difficulty of effective communication [11].

While this method has experiences success in developing general software for children, our contribution is a deeper understanding of the use of this method for educational and therapeutic software. A significant barrier when developing such software to teach a concept to the target demographic is the tension between the software being entertaining and playful and the software being successful in its didactic and therapeutic purposes. We began involving children and their feedback directly into the research for this tool.

In our first involvement we observed, behind a one-way mirror, children without cleft used an iPad to play virtual games within a larger study conducted by a developmental psychologist that aims at understanding various executive functioning tasks and attention of young children when playing virtual games on an iPad. The study was conducted in a living lab that looks like a playground, and each child plays without any other child in view but with an adult (usually his/her mother) next to him/her. In this study, we observed five children playing both virtual as well as physical jigsaw puzzles. We observed that children of the target age range are capable of solving the virtual jigsaw puzzle and can use the iPad effectively within 2 minutes of introduction and practice. When provided with supervision from an adult, the children we observed were able to engage with the virtual games for the required minimum time (20 min-

utes). Notably, children solved the virtual puzzle more quickly than physical puzzle by rotating the tablet, giving them different perspectives of the puzzle while staying seated. This behavior was not observed during the physical puzzle. The children observed were much more excited upon the completion of the virtual puzzle rather than the physical puzzle, although it should be noted that the virtual puzzle gave an audible 'click' when the pieces connected correctly and the picture centered on the screen. Also, upon completion of the virtual puzzle, an animation plays on the screen. This animation provides a greater reward to the child than the physical puzzle did. From these observations we concluded that a child would both This observation verifies that children as young as two can use iPad to play virtual games (actually, we found that the children were more excited about the virtual game than its physical version) and when the games are interesting, it is possible to sustain their attention for the duration required. The next step in our requirement gathering was to interact directly with young patients.

During the proof of concept we collaborated with surgeons and therapist to produce the requirements list in 2.2. Once we developed a speech recognition system meeting at minimum the high priority requirements, we revisited our collaborators at the UC Davis Medical Center to investigate how children and therapists interact during their meetings. We interviewed the doctors there about how they instruct parents to successfully coach their child and compared it to current literature.

Doctors explained to us that they instruct the parent to turn the exercises into games. In order to promote speech production SLPs use a variety of toys and devices to motivate the child, and SLPs recommend parents get creative using the SLPs office as a start point. They suggested to parents that keeping track of practice in a visual way for the child, such as putting a calendar on a cupboard or refrigerator and adding stickers when the child completed practice that day, keeps accounting accurate and gives the child an easy way to see their own performance. Incentivization such as pieces of candy for completion of practice or larger gifts when certain milestones were met are extremely effective, but are limited by the amount of time a parent or family can commit to being a part of therapy.

They explained that the at-home section of speech therapy was effectively impossible without a parent to both motivate and interpret the child's speech. They also expressed that one single parent usually does not have the time to manage all of the aspects related to the at-home component of therapy, and recommends the involvement of as many family members as possible. As a result a number of complications arise. The first is the lack of a trained ear. While family

members can hear speaking difficulties, they are not likely to know how to effectively coach their child towards correct speech production. Incorrect coaching and correction often leads to the reinforcement of compensatory structures, which are shortcuts to producing the correct sound but with a muted or slightly incorrect sound. Unlearning these structures increases therapy time, which increases the likelihood that the child will not complete therapy.

The second problem therapists deals with the inaccuracy of most family member's record keeping and performance analysis. The instructions provided to parents consist of a list of vocal exercises and the need for a minimum of 10 minutes per day of focused practice. However, other than the 10 minute time requirement, family members have no objective measures of the child's performance. The at-home component of therapy is often misdirected as a result with parents spending unnecessary time on completed sounds, while spending insufficient time on impairments or considering them finished prematurely. Consequently when the patient returns to the SLP's office, family members are often disappointed to learn that the child is not practicing enough at home.

While parents and family members remain a core part of a patient's support network, too much expectation is placed on this group. Doctors explained that having more objective measures of performance outside of the office would create a more direct line between the child and the SLP. This would reduce the number of tasks required of the family members and allow them to focus on more positive aspects of therapy such as motivation and reward.

We observed nine children being interviewed by the Cleft team. The setup of this particular Cleft and Craniofacial Reconstruction center requires the child to remain in the room for 3 hours (accompanied by their family). The therapy team, which consists of speech pathologists, plastic surgeons, psychotherapists, audiologists, and dentists rotate to see them. One common theme with those children were, although they were cooperative in doing tasks requested by the team (such as counting to twenty or repeating some sentences), most children were visibly bored to be in the therapy room. We noted that most of these children bring with them either an iPad to play games or a book for the times when no therapist is in their room. Most books are on Disney theme or other cartoons, providing us with an idea of the type of story to design. The games vary from coloring games to fast moving games (e.g., Mario Bros).

These exercises led us to a set of basic ideas about the system we should develop, which is an iPad based game that has an underlying speech recognition algorithm that can detect the required sentences. Using these observations we produced a requirements list for the tool.

Requirement	Priority
Plosive Impairment Recognition Rate of at Least 95%	High
Response rate of 2 seconds or faster	High
Interface usable by participants as young as 2 years of age	High
Motivating graphics and sounds for children	High
Games take 2-3 minutes to play	High
Grading and feedback is understandable by parents	Medium
System adapts to speech improvements	Medium
Recognition of both compensatory structures and hypernasality	Low
Recognition scores and ratings archived on device	Low
Grading and feedback is understandable by children	Low

Table 3.1: Speech Game Requirements

3.2.2 Video Game Analysis

In order to create an effective user interface, we first investigated commercial games that use speech recognition. Speech recognition has become more commonplace, easier to develop, and less resource-intensive since its inception; developers have successfully produced speech recognition-based video games. The games found were grouped into two categories: virtual pets and karaoke/singing games.

Hey You, Pikachu!, released in 1998, used a voice recognition unit that plugged into the back of the controller to allow players to control the game. This game had many firsts: the first commercially produced game to use speech recognition as the primary form of interaction, the first game to use speech recognition on a console rather than a powerful desktop or laptop processor, and the first game to tune the system to the higher pitch of a child's voice. Hey You Pikachu! uses a phrase-based interface: it does not process partial sentence, instead waiting for the user to complete a sentence before processing. The player holds down a button on the game controller to start recording. This causes the bubble in the lower right of the screen to grow and shrink according to the loudness of the player's speech. When the player finishes speaking he or she stops holding the button to allow processing to begin. On-screen the speech bubble moves from the player's silhouette in the lower right to Pikachu (the yellow animal) in the center of the screen. If the speech engine understands the phrase, Pikachu will perform the corresponding action. The in-game narrator helps the player towards the correct phrase by providing text hints with key words that the animal understands highlighted in red. Players alternate through different mini-games including hide-and-seek, fishing, and gardening. The two distinct forms of conversation are phrases, such as 'stay at my house' and 'wake up', and single words such as

on-screen objects or names of PokÃĭmon (other animals). If the user fails after a certain number of times, Pikachu will go through available choices affording the player the ability to interact with yes or no answers.

Other forms of interaction include walking around using the directional pad, selecting objects using a pointing hand and a button, and selecting items from a personal inventory. All of these supplement the speech interface, but do not provide speechless alternate courses of action. As a result, the game is unplayable without the microphone. This design forces the user into the speech interface.

While the game was praised for its novel form of interaction and appeal to young children, it was heavily criticized for its speech recognition engine. Most games are limited to two-or-three choices. The speech recognition engine suffered from severe accuracy issues even in quiet environments. Providing a binary interface using a spoken yes/no was implemented to maintain immersion in the event of serious recognition difficulty [37, 43].

Figure 3.1: Hey, you! Pikachu! Game Interface.

Seaman is a virtual aquarium game. Using a microphone, players talk to Seaman, an anthropomorphic fish, over a one-month period. Owing to the increased capacity of discs over cartridges Seaman was the earliest commercial console game found to respond to voice input with real voice input. This provided a new level of immersion as players could maintain a verbal conversation with the character.

Like Hey You, Pikachu!, Seaman uses a phrase-based speech recognition interface where players use a button to control the microphone. Players are introduced to a virtual aquarium and learn via trial and error how to properly care for a virtual fish named Seaman. Players must maintain a daily schedule of feeding and caring for Seaman. If they miss a day, Seaman will die.

Seaman is intended as a puzzle game for adults, and as such provides as little instruction as possible. The game does not provide in game hints, advice, or tutorial. It does not provide on-screen visual listening or loudness indicators. The only feedback provided for speech input is Seaman's verbal responses. If Seaman understands the phrase he will repeat what he interpreted and ask for confirmation. Players respond with a yes or no, after which Seaman responds. If Seaman has to restart his conversational loop too many times, he will become annoyed with the player and refuse to respond for a short while.

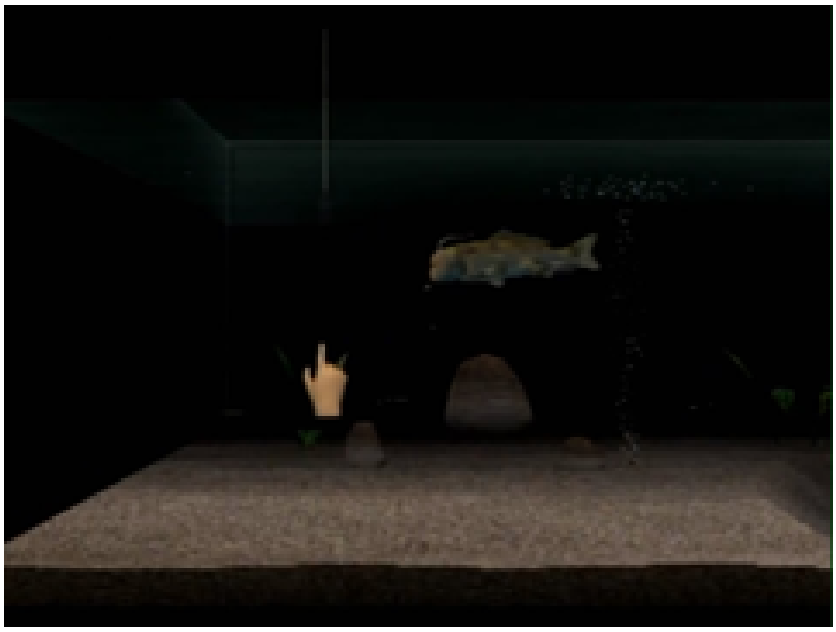

Figure 3.2: Seaman Game Interface.

As the game progresses, Seaman goes through various stages of evolution, moving from a parasite to a fish to eventually a frog. During the first few days players say phrases such as 'Talk' and 'I Love You' and receive gibberish or one word responses from Seaman, mimicking conversation with a baby. Once Seaman's language skills progress to a certain degree he will start asking questions when the player instructs him to talk. Questions begin with simple questions such as "How old are you?", progresses into personal questions like "Do you have a

lover? Do you live together?", and concludes with abstract or philosophical questions such as "Do you believe in God?" and "Do you know yourself?". Additional inputs include changing the camera view, moving a pointing finger icon around to interact with objects, and switching between the aquarium where Seaman lives and the terrarium where players may grow food for the later stages of Seaman's life. The game ends when Seaman grows large enough to leave the tank.

SpeakAZoo is a mobile app for iOS devices intended for children developed by ToyTalk. Players assume the role of a zookeeper tasked with talking to animals including pandas, porcupines, and warthogs. They have a short conversation, about 2 or 3 minutes, and then move onto the next animal. SpeakAZoo follows a conversational model similar to the later stages of Seaman. Players interact with the virtual pet by answering questions. Questions include open-ended questions such as "I'm hungry, what did you eat today?" and yes or no questions such as "Can I have a hug?".

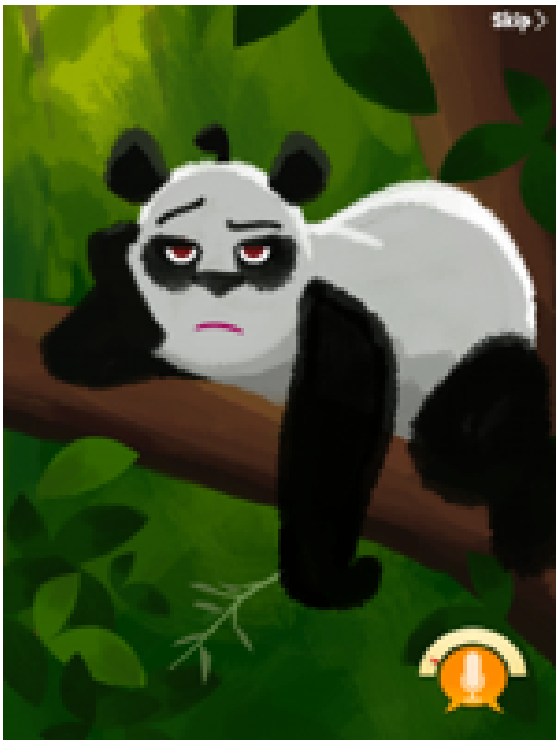

Figure 3.3: SpeakAZoo Interface.

SpeakAZoo animals never repeat a question nor do they ask players to repeat something in order to keep the player interested. If the system cannot determine what the child

said on the yes-or-no questions it gives an interesting response and moves onto the next question. SpeakAZoo's interface only has two buttons: the skip button on the upper right and the microphone button on the lower right. Like Pikachu and Seaman, users must push and hold down the microphone button in order to talk to the creature. The button is greyed out and unclickable when the animal is speaking. Slightly above the button is a loudness meter, with the level indicated by a red line. If players do not respond for 10 seconds a speech file is played reminding the player to down the speech button in order to talk to the animal.

SpeakAZoo has a screen explicitly telling children they must have their parent's permission in order to play this game. SpeakAZoo uploads recordings of the child's answers to questions to ToyTalks servers. Parents may log in to hear what their child was talking about with the virtual animal. Though parents are free to delete them, this provides ToyTalk with enough voice samples of children speaking to feasibly generate acoustical models for children.

Music and karaoke games require extremely low input latency and points are awarded for pushing buttons within very small timing windows. Speech recognition systems that respond as you speak, known as partial hypothesis, still wait for a user to finish a word or a predefined loop rate before returning a result. This is too slow for these games, as they need detection mid-word. Frequency detection involves a Fourier Transform to determine the strongest frequency, while phonetic detection uses Hidden Markov Models. While not directly speech recognition, it uses the same components.

Rock Band allows players to sing to their favorite songs while friends play controller versions of musical instruments. Rock Band allows players to play their favorite commercial songs in a video game and play different instruments with their friends. For the voice section of the game, input is provided via a USB microphone. The microphone is always live, so players do not need to hold a button down. Rock Band uses a combination of phoneme recognition as well as harmonic detection instead of speech recognition [1].

SingStar is a multiplayer karaoke game. It uses speech recognition during rap and hip-hop tracks, and frequency and phonetic detection for pop and rock tracks. SingStar's user interface is designed for multiplayer. Bars scroll across the screen in positions relative to how high or low the singer's tone should be. When a player sings the bar reacts to let him or her know if the singing is on key or not. At the top and bottom edges of the screen the lyrics for the song are shown and slowly fill up with color to indicate how fast or slow a certain word should be spoken.

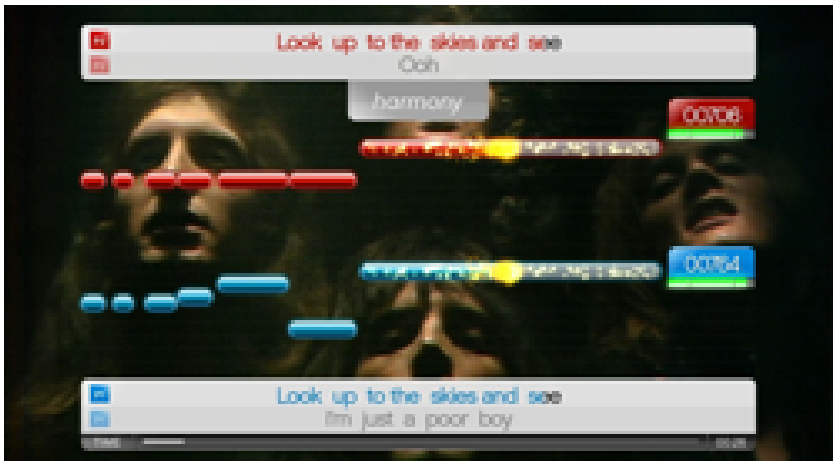

Figure 3.4: Singstar Interface.

3.3 Development

As we concluded our requirements gathering we created a flowchart showing the interworking tools and their resulting outputs. We learned about an enhancement to OpenEars called RapidEars that could analyze and record simultaneously. Though we developed the systems for the phrase-based system provided by OpenEars, we listed RapidEars as our ultimate design goal. The development breaks off into three sections: the speech recognition engine, the game engine, and the statistics engine. The game and statistics engines feed directly off of the input provided by the speech recognition engine.

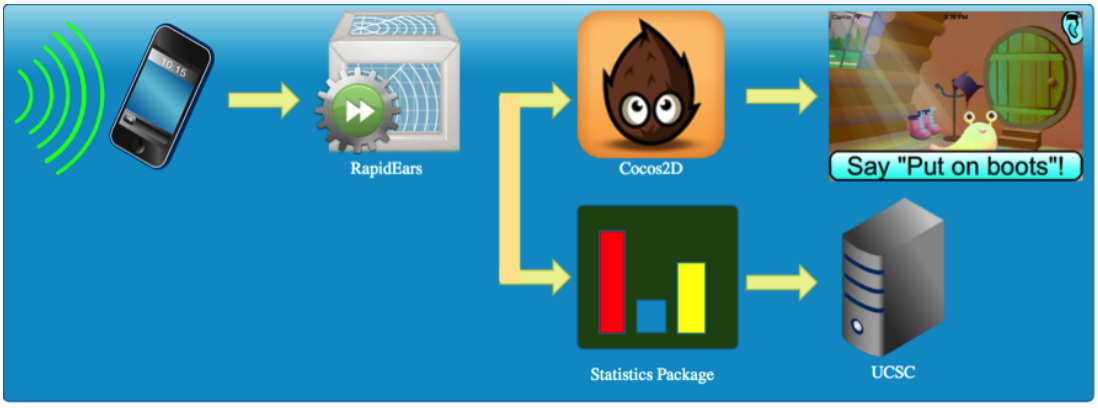

Figure 3.5: The Tool Technology Flowchart

The uncertainty of the hardware influenced the design of the games. Without knowing the device we could not guarantee a screen size or game performance. Thus the system was

designed with programming to scale the game images, sprites, and movement to whatever screen. This meant that templates or visual layouts were not usable during programming. As a result all levels are written programmatically, as are all sprite placements. This took a large amount of calibration time. The game's user interface (UI) has a lot of elements and as a result the artwork is less visible. Code was added to hide the UI elements for periods of time so players can see all of the art.

The speech recognition engine undergoes a few modifications. Most notably we are finally designing it towards recognizing the speech impairments we ultimately want to target. As the hardware remained uncertain we developed conservatively. When the chosen test device became and iPad Mini, we quickly provided the system with a larger acoustical model. Everything used was publically available, and we made no modifications to the acoustic model.

Initially the goal with statistics was to store the analyzed data on the device for therapists to later look at. Once we determined that the system would be tested in the office, we decided to only provide the therapist feedback on if the ASR determined if the speech was correct or not. Metrics were not available to the SLP and were uploaded to a secure location on the researcher's network. This turned into a positive move as we could correlate the SLP's feedback to what the metrics reported, and let us determine what tool was causing the problem.

3.3.1 Video Game

Video game-wise we decided that developing our own engine would take too much development time. Making individual engines for each level would increase the individual development time of each level, so we investigated off-the-shelf and open source options. Investigation quickly led us to Cocos2D as a suitable system. Cocos2D is an open source 2D game engine that provides a physics engine as well as the ability to respond to sensor input. With this we found it would be possible to develop a game in this engine that used speech recognition as the input.

We did not have access to RapidEars so we decided to develop for OpenEars, limiting us to phrase recognition. In phrase recognition the system waits for the speaker to complete a sentence before beginning processing. This provides more accurate recognition overall, but response is slower as it must wait until the end of the utterance before processing. This limitation restricted us the most game-wise, as initial investigations indicated that faster-paced games would not be possible.

We chose an interactive storybook as our game wrapper based on our observation and engine limitations. Our observations also revealed that these children understand simple instructions such as "Say the words on the screen". As a fallback option for children who are unable to read, a button at the bottom of the screen will read the instruction out loud. Additionally, in iTunes store found a plethora of popular interactive storybooks from companies such as Disney, which we can easily adapt to work with our speech engine.

As we used iOS-based devices our development platform consisted of XCode running on a 2011 15-inch Macbook Pro. This also meant that we used Objective-C as our langauge. While this would make any cross-platform attempts more difficult, it did unify our development process overall.

All games inherit from an abstract class known as a SpeechAdventureLevel. SpeechAdventureLevel combines three core game components: the audio manager which handles everything related to audio input and output (including speech recognition), the statistics engine which handled post-gameplay analysis, and the heads-up display which provides feedback to the user about their performance as well as the engine's behavior.

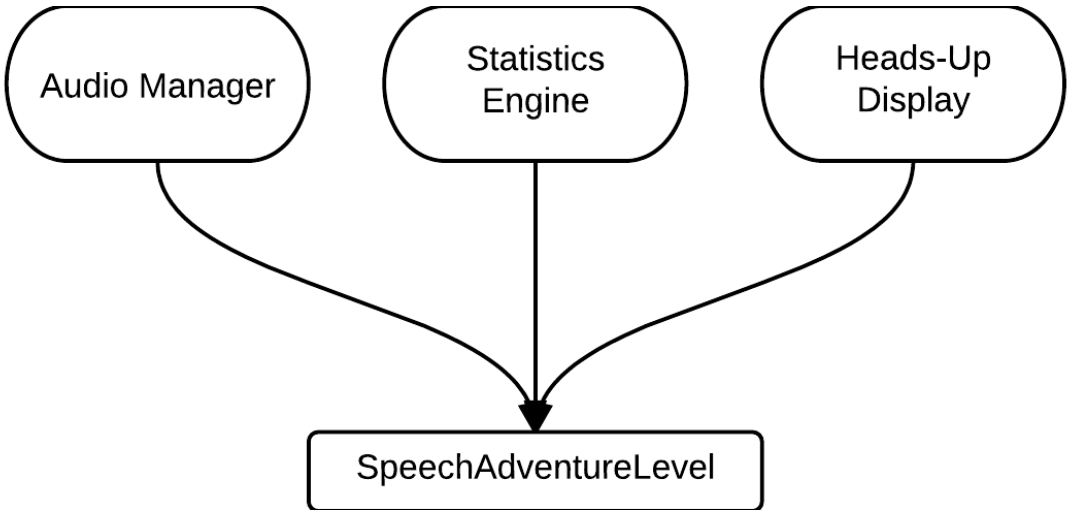

Figure 3.6: Speech Adventure Component Breakdown

The best way to have the game engine receive speech recognition input was determined to be delegates. In this setup, a universal manager controls OpenEars and another controls the statistics system. Levels connect to the manager and request to receive events fired whenever the speech recognition engine finishes analysis of the speed or whenever it changes settings. It also allows the game to directly start and stop the speech recognition engine.

3.7 shows the low-fidelity prototype of the game running in the iOS Simulator. The players must say the correct phrase in order to get the on-screen character to move. In this scene the phrase is "Go Right". We presented this to graduate researchers in human computer interaction field and interviewed them after about UI elements they would like. We combined this with our data collected in 3.2 to determine the elements to use.

Figure 3.7: Speech Adventure Low-Fidelity Prototype

Game	Listening Indicator	Loudness Indicator	Phrase or Real Time	On-Screen Text
Speech Adventure	✓	✓	Phrase	✓
Hey, You Pikachu	✓	✓	Phrase	✓
Seaman			Phrase	
SpeakAZoo	✓	✓	Phrase	
Rock Band	✓		Real-Time	✓
SingStar	✓		Real Time	✓

Table 3.2: Speech Game UI Elements Compared to Our Chosen UI Elements

The biggest feedback received from the graduate researchers was the need for interactive graphics and animations. We contracted a professional graphic artist to add art assets to our game. Based on our time and budget constraints we worked to create two levels. Taking

inspiration from the University we made the main character Sam into a banana slug. We also found players felt stuck if they could not say a sentence correctly. If they still could not after multiple tries, they would give up on the game. We added code that would automatically advance to the next target sentence after three tries.

We developed two levels: A home level and a river level. In the home level players respond to on-screen prompts as they interact with objects in Sam's home. In the river level they must say a phrase multiple times. The second level evaluates differences in ASR performance on a repeated phrase.

A narrator is provided a voice that kids could attempt to match. A female graduate student at the University of California, Santa Cruz was chosen as the voice of the narrator. When the narrator speaks, the speech recognition engine stops so that it does not pick up the narrator as a valid speech attempt. The narrator introduces the scene and congratulates the child when they say a phrase correctly.

Figure 3.8: Speech Adventure Level One High-Fidelity

In transitioning to high-fidelity we also designed a heads-up display in order to convey information to the player about their performance as well as activity of the speech recognition

Figure 3.9: Speech Adventure Level Two High-Fidelity

engine and the microphone. We iterated weekly on the interface, meeting with human-computer interaction researchers and gathering their thoughts afterwards. We frequently recruited new users who had never interacted with the system before to gain an understanding of the intuitiveness of the system.

Similar to SingStar, we chose to include on-screen text to instruct the child what to say. This prompt box received the greatest number of iterations. We added highlighting into text in the prompt box to let users know both when the speech recognition engine has completed listening and processing as well as what words they said correctly in the phrase. Initially the system only changed when all the words were correctly spoken. On feedback we changed it to highlight words that were correctly spoken. An interesting development out of the highlighting was that new users would continue the phrase where the highlighting left off. In the case of 3.9 users would say "A Balloon" repeatedly seeing that they had said "Pop" correctly. We altered the speech recognition engine to allow users to finish a phrase in this manner. We placed the box at the bottom, and hide it when it is not in use so that players can enjoy the full graphics.

Originally, touching the prompt box would cause the game to repeat the narrator's

recording of that target phrase. However during initial testing many users repeatedly touched the prompt box. This sometimes caused a glitch with the speech recognition engine that would cause it to either accidentally process the narrator's speech, or stop responding to input entirely. The only way to fix this would be to force quit the application. As a result we removed this functionality.

An ear in the upper right serves as the listening indicator. When the game starts the recognition engine is calibrated to the ambient noise and only begins the recognition loop when a threshold is reached, which turns the ear from grey to green. As the system would be tested in a quiet doctors office, we decided that a loudness indicator would not provide additional useful feedback. When the player pauses listening, the ear turns blue.

Due to the initial uncertainty in the testing environment, we added in a back button in the upper left corner of the screen initially that took the shape of an arrow pointing to the left. This would allow a player to return to the main menu, and would erase all data collected from the current play session. We originally did this as an easy way to find playthroughs from a single player. However as our testing environment changed to a controlled one-on-one engagement we removed the back button in the final version.

3.3.2 Speech Recognition

A significant barrier is the hardware limitations of mobile devices. The accuracy of speech recognition is a function of the sizes of the dictionary and the acoustical model. Large dictionaries will increase lookup time and decrease accuracy, while large acoustical models increase accuracy and lookup time but on a much greater magnitude than dictionaries. Doubling the size of the acoustical model more than doubles the response time. We decided that a response time of less than 2 seconds for sentences would be acceptable. OpenEars met this performance requirement for most devices, but failed on older models when the acoustical model was greater than 10 megabytes. As such we have chosen to adapt the models to make them more accurate to a particular child while contributing a negligible increase in size.

Using the discussing on target words and phrases in The Source for Cleft Palate and Craniofacial Speech Disorders, we developed two dictionaries for two in-game levels [56]. There are four separate levels of targets as they relate to speech structures: isolation, syllable, word, and phrase. For our evaluation system we selected the phrase level as testing with young adults showed that the recognition engine worked best with short phrases. The Source recommends

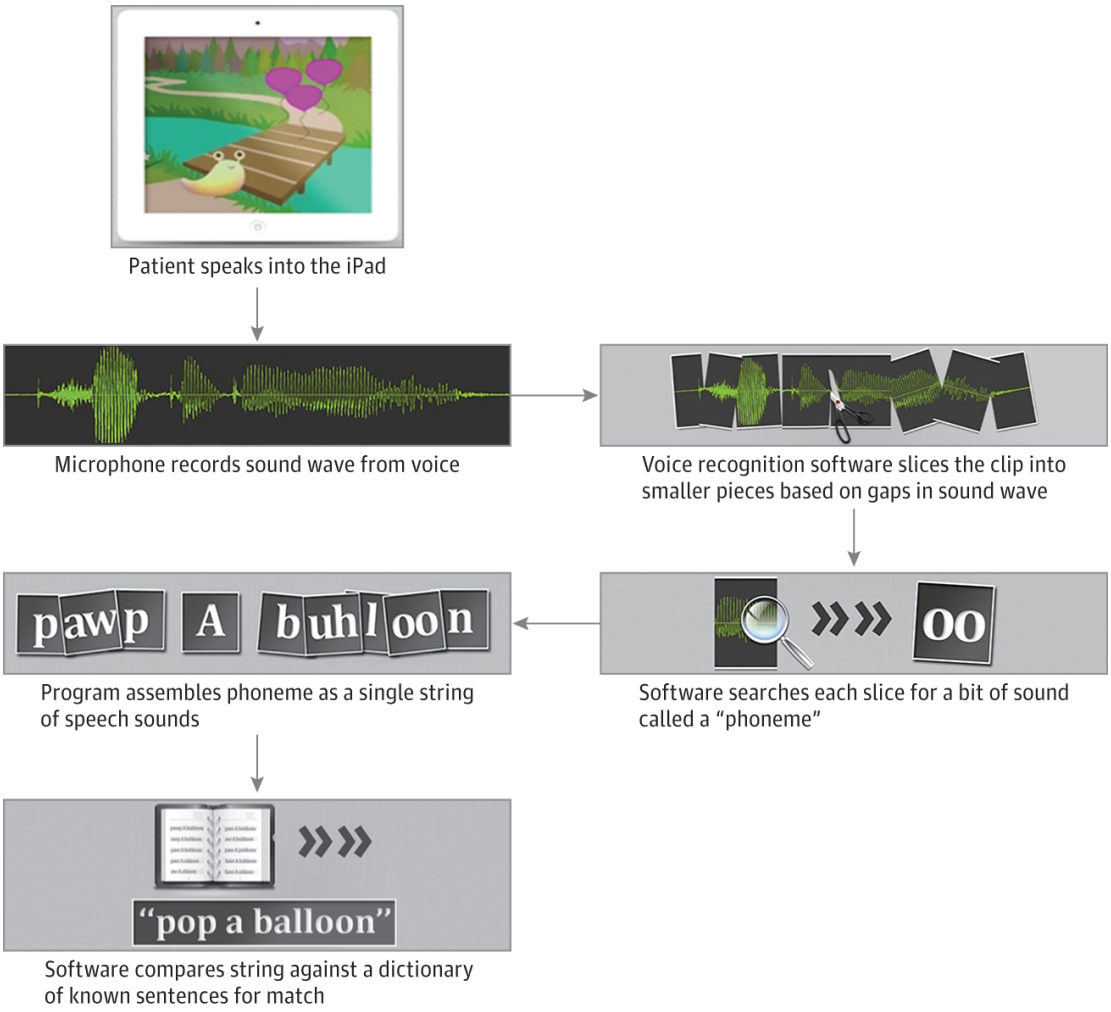

Figure 3.10: The Stages of Speech Processing in Speech Adventure

that treatment begin with glottal stops and plosives and includes the phonemes /h, p, b, t, d, k, g/, so we selected the subset /h, p, b, d, k/.

In order to create a dictionary we created a text file containing unimpaired phrases found in [56], with each phrase on a separate line. Next we fed the text file into the lmtool web tool provided by the SPHINX knowledge base [51]. This tool converts sentences into a dictionary and language model and returns them in separate files. Once we acquired those files we proceeded to open and edit them. In the dictionary file we added alternate versions of target words with these containing the impaired versions. In the language model we added the impaired versions in as well, giving them equal probability of occurring as the correct version of the speech. Though editing the weights in the language model could potentially make the game

harder or easier, we did not investigate this.

We used one mispronunciation per target word, having found in the proof of concept that it produced the highest accuracy for discerning impairments. We tested two versions of mispronunciations: a nasal emission sound characterized in ARPAbet as an HH, and a complete omission of the sound that also removed the ARPAbet phoneme. We found that the nasal emission sound, HH, produced acceptable accuracy. We also tested a variety of sentences and words with non-impaired speakers before finding 5 sentences that the ASR accurately differentiated between non-impaired speakers talking normally and with cleft speech imitations. We included two target words in all sentences except one, producing 5 target words in the first level and 4 target words in the second level.

Sentence	Target Word	ARPAbet
Put on Boots	Put	P UH T
	Put (Cleft)	HH UH T
	Boots	B UW T S
	Boots (Cleft)	HH UW T S
Wear a Hat	Hat	HH AE T
	Hat (Cleft)	HH AE
Open the Door	Open	OW P AH N
	Open (Cleft)	OW HH AH N
	Door	D AO R
	Door (Cleft)	HH AO R

Table 3.3: Speech Adventure Level 1 Dictionary

As a child progresses through therapy, so does his or her ability to pronounce the troubling syllables properly. As a result, the system must contain sentences of difficulty sectioned into levels corresponding to the therapy progress. Unfortunately, as mentioned in the previous section, the application suffers severe performance penalties if the acoustical model grows too large. A game that stalls for too long to generate a response to a user input will significantly detract from the immersion in the game, and subsequently, the potential learning benefits. Therefore, we must maintain a high level of accuracy as the child's ability improves while maintaining a small acoustical model to prevent noticeable delay in response.

Acoustical model adaptation permits us to significantly increase the accuracy of a given acoustical model without a noticeable increase in size. Adaptation operates by taking a few transcribed wave files and merging them with the existing model. Patients visit their speech therapist on a weekly basis. Trained speech therapists can discern minute changes in speech,

making them the ideal person to decide what is and is not correct speech in the situation.

As both examples of correct and incorrect pronunciation are crucial to improve accuracy, the therapist will provide an extremely important role. Previous research has demonstrated that adaptation can improve accuracy by up to 10% for the standard word error rate (WER) measurement. By using both correct and incorrect pronunciations in the data, we expect to find a more significant jump in accuracy. Using acoustical model adaptation allows us to increase the difficulty by tuning more aggressively towards correct and incorrect pronunciations without increasing the response time. Adaptation will also permit us to recycle game scenes, reducing the overall size of the game as well as the number of scenes needed. As children would play the game for only a few minutes we did not expect to need to perform adaptation, and elected to have players say the phrase "Pop a Balloon" three times to determine if any short-term differences in recognition occurred.

We implemented three different parts to assist OpenEars: A manager, a delegate handler, and a natural language processing (NLP) engine. OpenEars has a manager that controls OpenEars behavior, the delegate handler sends information to registered delegates, and the natural language processor which filter the text to make it usable as game input.

The OpenEars Manager provides a simplified way to interact with the lower-level and more complicated parts of OpenEars. Control-wise it provides access to the microphone and the recognition loop, where it can start, stop, or pause both. Initially a cut-off point of 3 seconds for recordings were added to prevent a recording from going too long but this caused bugs and crashes and was removed in the final version. The manager can also set the language model and dictionary pair, which cannot be changed while the system is recording but does not require a system restart. It creates two types of events: state changes and recognition texts. State changes let the level know when the recognition loop is ready to listen, and the recognition text is the raw recognition text. Both events are passed to the delegate handler.

The delegate handler takes the events and passes them to the corresponding delegates. A level registers as a delegate to receive the speech events and to know when to start interaction. In this style there is no delay or wasted cycles caused by polling. This style also allows for multiple levels to be played at once using the same speech input, though we did not use this feature. Once level receives the event it optionally uses NLP system to further clean up the speech response before finally using it as game input.

The NLP system scans and filters the output provided by the OpenEars engine. Levels

are not required to use it, but it cleans up the input and improves interaction when applied. OpenEars outputs its response, called a hypothesis, as a string of uppercase words. This hypothesis is noisy: in a given target phrase there may be additional words before and after, as well as filler words such as 'uhm' and 'ah'. The NLP system applies string filtering to separate out target words and phrases from the rest of the hypothesis. In the game, the two levels use different filters to achieve their specific gameplay.

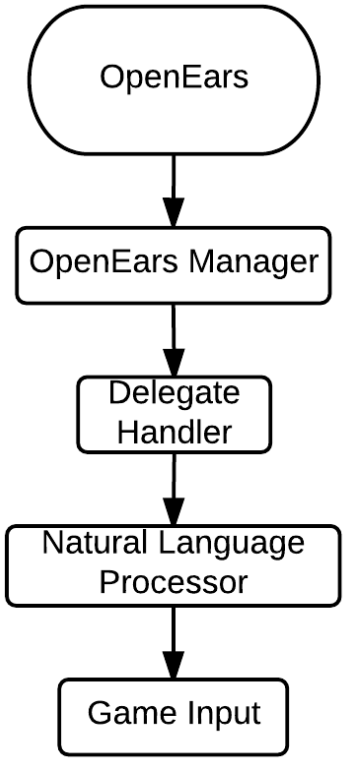

Figure 3.11: OpenEars System Flowchart

In the first level the system will highlight correctly said words in the prompt box in yellow briefly and then return the text to the default color of black. Players will know what words they said correctly, but must say the entire sentence correctly to progress in the game. In the second level the game leaves correctly said words highlighted serve as a checkpoint, allowing the player to continue progressing through the sentence where they left off. Players must say the words sequentially in order for them to highlight. As an example: in the sentence "Pop a Balloon", balloon will not highlight until the user says "Pop" first, but "BALLOON POP POP A POP BALLOON A BALLOON" is a correct utterance. Whenever OpenEars returns a

hypothesis it is stored along with the level and the target sentence. The utterances are passed to the statistics engine, which collects and stores the utterances.

3.3.3 Statistics

During our interviews with doctors and SLPs we discovered that they instructed parents to have their child practice a list of exercises for 10 minutes per day. They did not provide further instruction such as the rigorousness of the practice period, what to listen for in terms of progress, or how to evaluate if the child is performing the exercise correctly. Parents were provided more instruction overall towards motivating, incentivizing, and rewarding the child. This provided a wealth of areas for us to investigate and provide data on.

Our research stipulations prevented us from collecting any patient records. We agreed that speech recordings would be considered identifiable patient data. Since we couldn't collect any speech recordings, all of our data is text-based. This limitation let us focus on collecting data that therapists would not have access to without a mobile device. We started by creating a statistics engine that collected three pieces of data: time spent active in game, total number of correct words spoken per sentence, and total number of sentences spoken.

Similar to OpenEars, the statistics engine has a manager that allows levels to add information. It also handles server communication once gameplay is finished. The statistics package currently provides data collection, but not visualization or analysis. Though the device has the capability to perform the analysis, we did not have a data visualization package and so decided to analyze the data off-device.

We collected playtime on two different scales: per level and per playthrough. This would allow us to determine if players got stuck on a specific level. This would also help us understand the range of the number of sentences a player could say in 10 minutes with this style of play. As we had a goal of 2-3 minutes playthrough time, we would see the range of sentences over a quarter of the overall therapy time.

The smallest piece of speech data we collected is called an 'utterance'. An utterance contains the raw output from OpenEars for a given speech recording. A perfect playthrough of the game would have one utterance per target sentence, but this is not always the case. In the worst case the player says each sentence 3 times for a total of 21 utterances. Utterances are organized by their target sentence, which are in turn organized by the level. An example of a playthrough follows:

Total Playtime: 120.650590 Is Child: 1 LevelName: Tutorial LevelTime: 16.966076 TargetSentence: GO RIGHT Spoken: GO RIGHT LevelName: LivingRoomPlosives LevelTime: 62.490056 TargetSentence: PUT ON BOOTS Spoken: PUT ON BOOTS HOOTS A HAT A HAT Spoken: PUT ON BOOTS HOOTS A HAT A HAT TargetSentence: WEAR A HAT Spoken: WEAR A HAT Spoken: WEAR A HAT TargetSentence: OPEN THE DOOR Spoken: OPEN THE DOOR Spoken: OPEN THE DOOR LevelName: PopABalloon LevelTime: 40.973785 TargetSentence: POP A BALLOON Spoken: POP A BALLOON TargetSentence: POP A BALLOON Spoken: BALLOON Spoken: POP A BALLOON TargetSentence: POP A BALLOON Spoken: A BALLOON Spoken: POP A BALLOON TargetSentence: CROSS THE BRIDGE Spoken: CROSS THE BRIDGE

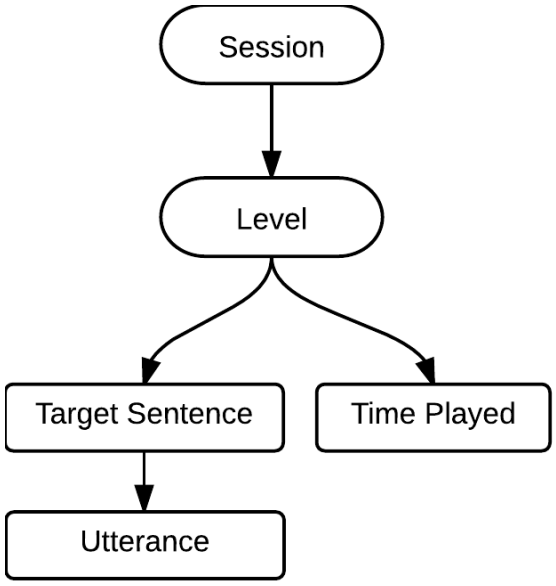

Figure 3.12: Statistics Structure

At the end of the second level the game would complete. At this time it would upload the data to a private address on UC Santa Cruz's servers for researchers to analyze later. A PHP script on UC Santa Cruz's servers handled writing the data to a text file. This section required an Internet connection, but did not check for a WiFi connection before uploading.

3.4 In-Office Evaluation

We recruited individuals that met both inclusion and exclusion criteria. Children who are followed by the UC Davis Cleft and Craniofacial Team and have been identified for

speech therapy served as the base study population. This included anyone under 18 who had a cleft palate with or without cleft lip and had a cleft palate repaired at less than 18 months of age. Exclusion criteria included children with other providers, children who have undergone secondary speech surgery, unreliable patients, additional craniofacial anomalies, neurological impairment, cognitive impairment, or if the literacy level of the parent is inadquate to obtain informed consent.

Ten children with CP and/or a cleft lip (aged 2-7 years) were identified as needing ST between May 29, 2013, and July 26, 2013. Children with cognitive impairment or profound hearing loss were excluded, as they needed to complete audio commands. The University of California, Davis Institutional Review Board approved this study. Written consent was obtained from the children's parents. During the pilot test, doctors handed out the game on an iPad to children under evaluation for various stages of cleft treatment. Doctors recorded the system's evaluation of the child's speech, and compared it against their own as a control. A statistics collection package was developed that would collect two pieces of gameplay data: time spent per level, and the raw output of the recognition system each time the user spoke. Data were recorded and uploaded at the end of the play-through to University of California, Santa Cruz's servers.

Out of 10 children who participated in the pilot test, data were collected on 9 children. The first child's data were lost due to network difficulties leading to data loss. Of the selected sentences for the pilot test, 3 out of 5 demonstrated high enough accuracy for our purposes: Put on Boots, Wear a Hat, and Cross the Bridge. These three had one false negative each, where the system reported the player as speaking incorrectly and the doctor disagreed.

Therapists found the speech recognition system extremely accurate on level one. The recognition engine correctly detected plosives mispronunciations for /h/, /p/, /d/, and /b/ for 8 out of the 9 children in the test. The 9th child produced false positives: or the system detected a mispronunciation where there was not. Sentences on level two - those with two mispronunciations per target word - graded most utterances from the children as incorrect. Therapists tested the game independently and found the second level would grade the therapists as incorrect.

Children loved the games. Upon receiving it they asked if there would be more. However, they were bored at the slow pace of the storybook-style games and found having to repeat things multiple times as tedious. Doctors said this was extremely notable during the pop a balloon section where children must pop three balloons. Doctors requested the game take 2-3

Table. Patient Characteristics

Sex/Age, y	Age at Palate Repair, mo	Diagnosis	Perceptual Speech Assessment	Speech Therapy Consonant Targets
M/2	10	BCLP	50% Intelligibility, compensatory articulation, borderline hypernasal resonance	/p,b,d/
M/2	10	UCLP	Fair intelligibility, balanced resonance	/p,b,d/
F/2	13	CP	50% Intelligibility, balanced resonance	/p,b,d/
M/7	11	BCLP	90% Intelligibility owing to hypernasality	/k,g/
F/6	11	UCLP	Poor intelligibility, compensatory articulation (glottal stops, pharyngeal fricatives)	Establish production of /p,b,t,d,k,g/ in school speech therapy
F/6	11	UCLP	Inconsistent nasal air emission	/k,g/
F/6	13	BCLP	Mildly reduced intelligibility, hypernasality	/k,g/
M/7	12	CP	Mild articulation disorder, inconsistent nasal air emission and hypernasality	/k,g/
M/6	14	UCLP	Mild articulation disorder, mildly hypernasal, moderate nasal air emission	/k,g/ in school speech therapy
F/2	12	CP	Compensatory articulation, backing and glottal stops	/p,b,t,d/

Abbreviations: BCLP, bilateral cleft lip and palate; CP, cleft palate; UCLP, unilateral cleft lip and palate.

Figure 3.13: Participants in the Study

Sentence	Agreements
Put on Boots	8/9
Wear a Hat	8/9
Open the Door	4/9
Pop a Balloon	4/9
Cross the Bridge	8/9

Table 3.4: Evaluation Agreements between System and Pathologist

minutes to play. Figure 7 shows the amount of time each child spent in game. The average game time was 171 seconds, with a standard deviation of 48 seconds, or 2 minutes to 3.5 minutes showing that the games are of correct length. This result shows that there is not enough to interact with on one screen, and players want more control over the story. Though the game provided touch input, it was used for changing settings and not interacting with the actual game.

Our medical collaborators mentioned children complained that the recognition system took a significant amount of time in between sentences. Upon investigation it was found that the phrase-based speech recognition engine is configured to wait for a silence of acceptable length. When calibrated in a quiet environment, soft noises such as a foot tapping or someone shifting in a chair would reset the silence timer. Additionally, a short but noticeable delay between the child beginning to speak and the listening indicator changing color resulted in some confusion, though it did not affect performance overall.

The medical collaborators noted that the youngest participants, a 2-year olds, were unable to play the games. Children at that age had learned the word "pop" but not "balloon" making the game unplayable for them. They mentioned they used the pause button located on

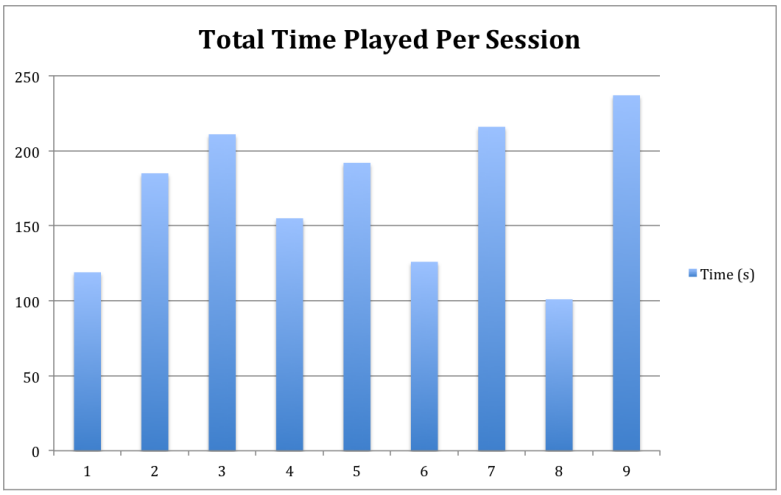

Figure 3.14: Time played by participants

the ear with almost every participant at least once in order to explain additional instructions. Because of this, the button embedded in the text box went unused as therapists preferred to explain directly to the child.

During follow-up interviews, doctors and therapists agreed that this system would be easy enough for children to play on their own at home, with some minor adjustments for the youngest participants. The goal of the system is to motivate a child to practice target words and phrases as much as possible within a 10-15 minute period, so game development moved towards this use case. Therapists also requested the ability to see the child's performance and the ability to adjust what phonemes the games would focus on for different periods of therapy.

Chapter 4

Long Term Test: Speech With Sam

As the pilot test proved extremely successful, we began work on the intended tool for the research: a tool to enhance the at-home component of therapy. Until this point in the research we were limited by either speech recognition's capabilities or device performance limitations. We also found frameworks to assist in data visualization.

We crowdfunded to acquire the supplies for this tool. This included a license for RapidEars, graphic design, iPad Minis for distribution to children, and a new development laptop. Additionally the crowdfunding would provide us extra publicity and potentially more recruitment and interest in the project. Though we were featured in a local news article, we did not find additional participants through crowdfunding. Overall we did not find crowdfunding to be an effective fundraising or publicity for research.

This tool implements RapidEars as the primary speech recognition component. Though the storybook-style game became very polished both on the game side and on the speech recognition side, the game was not effective at the primary game goal of saying as many target words as possible in a short period of time. The resulting game bears little resemblance to the game tested in the previous chapter. In the final version we converted the old game to use the new speech recognition engine and put it into a Storybook. The statistics system was fully built out, storing the statistics in an organized database, providing visual feedback about at-home performance, and initial control over games and dictionaries to direct therapy.

The suite allowed for parallel testing, so we greatly expanded our user base. As we tested with our target group of children enrolled in speech therapy we also evaluated the game with other groups who could benefit from speech instruction. We designed for this by having

the game check for other speech characteristics like rhythm and volume. Collecting data from a variety of users provided us perspective into the individual needs of those who need speech therapy.

4.1 Related Work

A study of 4,000 SLPs conducted in 2016 by the American Speech-Language-Hearing Association (ASHA) found that 52.4% of areas in the United States have more job openings than seekers for SLPs, with 82% of SLPs in Pacific states reporting that there were more openings than seekers. 69.6% of SLPs reported a high workload/caseload size as one of their greatest challenges, and 82.8% reported that the large amount of paperwork directly impacted their ability to provide therapy. A caseload characteristic report found that SLPs spent 19 hours weekly in direct intervention, 7 hours of documentation and paperwork, and less than 1 hour weekly on services to disabled students. Currently caseloads are growing at a rate of 7% per year for SLPs, and 57% of SLPs who brought in assistants or aides stated that it made no impact or increase their workload overall [4]. Telepractice, or using videoconferencing for remote therapy and the latest technological introduction to speech therapy, remains unused by 74.9% of school districts.

In the commercial realm there we found few games and tools intended for speech therapy on web and mobile. Tiga Talk provides 3D mobile games intended for speech therapy. It provides a loudness meter and checks for 23 different core phonetic sounds. While positive reviews state "I have found that the game can hold his attention for reasonable periods of time and that he attempts sounds, something we have trouble getting him to do in other situations.", critical reviews of the system report "It gives incorrect feedback for the targeted sound. Basically, if you make a sound, it praises you." The system is exclusively designed for children, with no therapist feedback component [57].

Doonan Speech Therapy provides Speech with Milo, a suite of tools intended for work with children in early language learning and speech therapy. Speech with Milo includes a speech therapy app, intended to work with therapists and children. Their work resembles the system we developed during our pilot test, with more polished graphics and gameplay. Their system differs in that it contains no ASR component, and requires a therapist to grade the child. Thus it is intended for in-office sessions only [58].

Figure 4.1: Tiga Talk Pop a Balloon Game

Rose Medical provides a variety of tools including speech therapy games, and pronunciation aids that use speech recognition. Investigation of the speech therapy games found that they had no speech recognition component, instead using voice reactivity and having the game input relate to features such as pitch and sound intensity rather than simply coherent speech. Their pronunciation aid required a desktop or laptop computer, had a 21,000-word dictionary, required a silence before processing, and seems to be optimized for British English. Rose Medical evaluated their automated speech intelligibility scorer at the University of Reading in 2014. Using recordings of adults with a variety of speech impairments, they evaluated the performance of their system against naïve human listeners. Researchers found a positive correlation between scores from the automatic evaluator and from the naÃŕve human listeners, indicating their intelligibility scores were similar. Unlike our work, their work did no investigation into the user interface and tested recordings instead of live speakers. Their work has yet to receive formal peer review [39, 38].

You-sheng Shiu created an OpenEars-based treatment system for aphasia as part of a masters thesis at the National Sun Yat-Sen University in Taiwan. After creating an app around it titled Dephasia, Shiu tested the system using 156 speech recordings from 10 patients with two different forms of input: repeating a shown sentence or answering a question. They noted that OpenEars does not process long recordings as effectively as short recordings, so they had to cut 25 second recordings into smaller samples of 1-2 seconds. Shiu found that with an ideal recording the system could grade speech on par with a therapist. He found that the system took up to 11x real-time (RT) on an iPad2, or 2 seconds to process a 22 second sentence. The

Figure 4.2: Speech with Milo game side

iPad4 came out as the best mobile device averaged 2.78x RT, 4.5 shows the performance of the devices tested compared to the speech length [53]. Notably, Shiu chose to use the recognition score provided by OpenEars and translated its score into a score of 1-6, hoping to achieve a universal intelligibility metric. Halle Winkler, developer of OpenEars, stated around the same time that the hypothesis scores are not useful as an absolute metric for all speakers. He did state that someone could use the scores to determine if a single user has improved speaking in a short session [63]. Our system is designed around a time goal of playing 10 minutes per day for 1 week, rather than an overall pronunciation performance. Our system also intended to investigate the efficacy of the in-game scoring system we developed. As a result, we did not use hypothesis scores in our system.

4.2 Requirements Gathering

After the completion of the pilot test, we interviewed doctors and SLPs again to understand the successes and failures of the tool. Doctors and therapists agreed that this system would be easy enough for children to play on their own at home, with some minor adjustments for the youngest participants. However, a number of changes were necessary to prepare the system for at-home therapy use. The speech rate in game needed to increase, the statistics

Figure 4.3: Speech with Milo Performance Section

needed polishing, and the game engine needed changes.

In adjusting the game design we consulted doctors, children, and game testers. Doctors stated that the game needed to have children saying more words and sentences. Children requested a faster-paced game, with more to interact with and more game reactivity. Game testers at the University of California, Santa Cruz, reported that the game mechanic itself was fun, but the game design and levels led to an overall low replayability.

Therapists also requested the ability to see the child's performance and the ability to adjust what phonemes the games would focus on for different periods of therapy. They requested that the game have a feature to enable simpler sentences for the youngest participants and more difficult sentences for older patients and those with subtler impairments. Additionally, the game should be able to adjust difficulty via either the speech recognition engine's grading as well as raw word difficulty.

SLPs and doctors would see statistics from the child's at-home practice in this revision of the tool, so we discussed data collection and data visualization. They informed us that consistent daily practice was the key to success, and that charting daily performance would be the most valuable to them. They stated that audio data would only assist them in determining if the tool is evaluating sentences correctly, and that this would be obvious during the following meeting with the child if the child has not improved.

Figure 4.4: Rose Medical Pop a Balloon Game Using Voice Intensity

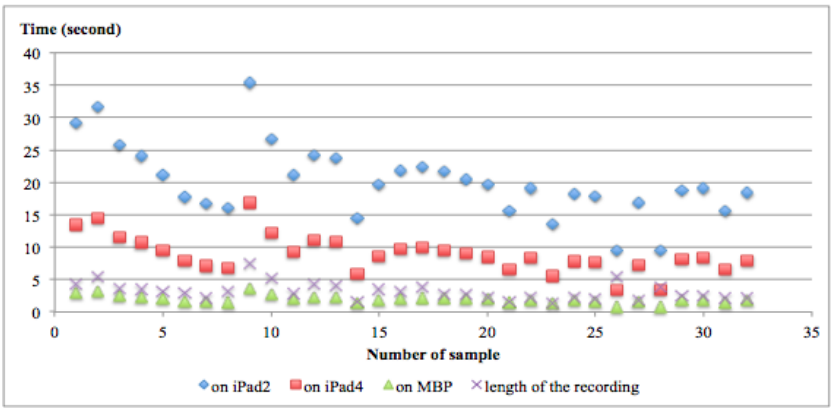

Figure 4.5: Device Performance on Shiu's OpenEar Tool

We asked SLPs what aspects of speech beyond pronunciation would benefit from grading and evaluation. They reported that rhythm (saying the word at a controlled rate) as well as production of a target word on demand were beneficial speech characteristics to evaluate. As we added these in we determined that we would need a scoring method would be needed.

The in-game narrator was originally supplied to give children additional motivating elements as well as give them a speech reference. Our chosen narrator spoke with a different regional dialect and accent than the children we tested the game on. This caused some children to imitate the speaker. Though this was our goal, doctors informed us that the differences in speech could affect therapy performance. As a result a narrator with a neutral West Coast

accent was needed. We agreed that a narrator that sounded as close to the player as possible would best assist with pronunciation.

Automatic progression through the difficulties was also requested by SLPs. Once a player achieves a certain accuracy rate or greater accuracy in a video game, the engine should provide a more difficult dictionary. As therapy progression is rarely linear the game should take this into account and dynamically adjust the difficulty downward as needed.

Requirement	Priority
Compensatory structure recognition rate of at least 95%	High
Recognition rate and game reaction in less than 0.5 seconds	High
System does not require an internet connection	High
Game fun enough to play for 10 minutes a day over 1 week period	High
Narrator speaks in a neutral American English accent	High
Games have a high level of replayability	High
Scoring of Speech Rhythm and Production On-Demand	Medium
At-home performance information displayed graphically	Medium
Game Engine and Speech Recognition Engine are bug-free	Medium
Games automatically adjusts difficulty	Low
System automatically selects games based on needs	Low
Multiple dictionaries usable in the same game level	Low
Narrator sounds similar to player	Low

Table 4.1: Take-Home Game Requirements

4.3 Development

Based on our discussions with therapists we adopted the same developed scheme for the pilot test. As such we divided the development into three sections: speech recognition, video games, and statistics. On the speech recognition end we integrated RapidEars, began utilizing loudness and volume feedback, and created dictionaries of different difficulty levels. Within the games we added more physical input, added new elements to the HUD while decreasing its total screen real estate, and moved towards a series of microgames. Statistically we converted the database structure to SQLite3, developed methods to aggregate data from multiple sessions, introduced data visualization, and provided SLPs feedback and control over consonants.

While RapidEars had existed since the pilot test tool, it was in the early stages of development. Early investigations indicated that the framework's recognition accuracy did not meet our needs, and the devices had difficulty handling the frequent recognition calls while maintaining an acceptable framerate and gameplay speed. Inside the iOS simulator we determined

that the iPad Mini 2 had the capabilities necessary.

We moved the system to a variety of mini-games, though we left the original games in as tutorial mode. We developed a total of 8 games in two of the difficulties: word, and phrase. We introduced scoring and provided extra feedback for additional characteristics. The HUD received a complete redesign and elements were added including a timer and a scoring system.

Visualization of at-home performance was fully developed in the statistics engine in this version. We converted the database SQLite3, with games providing entries based on speech input and game actions. As the goal of the SLP is to see day-to-day consistency, we created methods to aggregate multiple sessions in a day, enabling therapy in short bursts rather than all at once. We also provided early prototypes of individual consonant control and feedback screens.

4.3.1 Speech Recognition

The first improvement was adjusting the speech recognition engine to begin processing partial phrases. This affords the game the ability to process speech while still recording and was achieved using the OpenEars plugin RapidEars. With this system in place we developed games to test the new capabilities of the recognition engine. The system does provide the ability to determine the timings of processed words as they relate to other words enabling rhythm games.

Sentence	Target Word	Difficulty Level
Corn	Corn	Word
Cross the Bridge	Cross	Phrase
	Bridge	Phrase
Make a Cake for Nick	Make	Conversational
	Cake	Conversational
	Nick	Conversational

Table 4.2: Dictionary Entry Examples and Difficulty

We selected the iPad Mini 2 as our device for development and deployment. We decided to keep the dictionaries small, in order to maintain high accuracy and minimize any interference. We did opt to use the largest English acoustic model available, which at the time of testing was 17 megabytes in size. The iPad Mini 2 did not exhibit any noticeable delay with this increased acoustical model size.

The ASR engine stops at the end of a level. This provides the game with a number of benefits including recalibrating the noise floor every level to improve overall recognition accuracy

and quality. It also allowed us to play speech-based sound effects before and after play, though we did not use this feature.

Initially we had planned to do acoustic model adaptation on the device. However, the associated files took up 3-4 times the space of the game itself. We opted to leave adaptation off of the device. We did leave the tools in place on UCSC's server for adaptation, in case adaptation became necessary.

As we added in RapidEars, our sound engine went from half duplex to full duplex. This means that we could play sound effects while the recognition loop ran, as the recognition loop no longer looked for a period of silence before beginning analysis. However, recorded speech played back through the speakers would be processed as speech input. Compounded to our other issues with the narrator, we decided to remove the narrator entirely.

We implemented dictionaries for word and phrase, but did not create long sentences or tongue twisters. Dictionaries implemented at the word level are similar to command-and-control dictionaries for popular software, while conversational difficulty is intended to be closer to - though not as hard as - conventional tongue twisters.

4.3.2 Video Game

The goal of the system is to motivate a child to practice target words and phrases as much as possible within a 5-10 minute period, so game development moved towards this use case. The original storybook-style game and OpenEars combined did not produce a high enough speech practice rate to supplant existing paper exercises. With the introduction of RapidEars, we completed redesigned the game system. We determined that mini and micro games would serve as the best option, giving us flexibility in game development while keeping games simple and intuitive.

Micro games are a relatively new form of game, with the term coined by the series WarioWare. Micro games differ from traditional mini games in that they are even shorter and simpler than traditional mini games. A level takes seconds to complete, and rotate quickly [23]. 4.6 shows two sample WarioWare games. In the micro game in the player must move the goalie in order to block the ball from going into the net at the bottom of the screen. The bomb at the very bottom left of the screen counts down to indicate the time left to complete the micro game. The first game gives the user three seconds to block the goal, while the basket game gives the user five seconds to make the basket. If the user fails the task or does not complete it within

the allotted time, the user does not gain points and the game moves onto the next Micro Game. This design allows for extremely modularity as well as implementation of procedural content generation. Procedural content generation allows designers to recycle concepts and material pseudo-randomly, allowing for new game creations based on sets of rules. Mark Nelson and Michael Mateas have investigated the potential to apply procedural generation algorithms into a micro game system during their deconstruction of micro game components [40].

Figure 4.6: Micro Game Example from Wario Ware

We adjusted the game from a storybook-style game into a collection of mini-games, similar to Wario Ware and Mario Party, to address the boredom. We took inspiration from web games and mini-game collections. An individual mini-game takes between 15 and 30 seconds to play through with the goal of a child playing around 30 levels in a play through of the game. The majority of the games will follow one of three game formats: say the correct word once within a short time limit, say a target word or phrase as much as possible within a longer time limit, and play-until-fail games that will increase in speed.

One major design focus was removing as much negative feedback as possible. We designed the majority of many games with no explicit win or lose conditions. Only two mini-games developed, 'Flappy Slug' and 'Space Invaders', had the ability for the player to lose. In

Flappy Slug (A speech-based Flappy Bird clone), players could lose by falling off the bottom of the screen or by hitting the far left wall. Players did not lose if they hit the ceiling, indicating they are able to consistently produce the target word or phrase. In Space Invaders players could lose if an enemy projectile hit the player's ship, or win if they remove all of the ships from the screen. Players scores only increase, and they are reset to 0 at every new level. In highlighting phrases we did not provide additional highlighting informing players of words they said incorrectly. We added a checkmark at the end of the phrase box that appears briefly whenever a target word or phrase is said in full correctly. The majority of games used a combination of an NLP filtering system that allows players to continue phrases from a highlighted word, and one allows players to say the target phrase with unnecessary words added. These two components causes the game to give the input as much leniency as possible, making the games much easier overall.

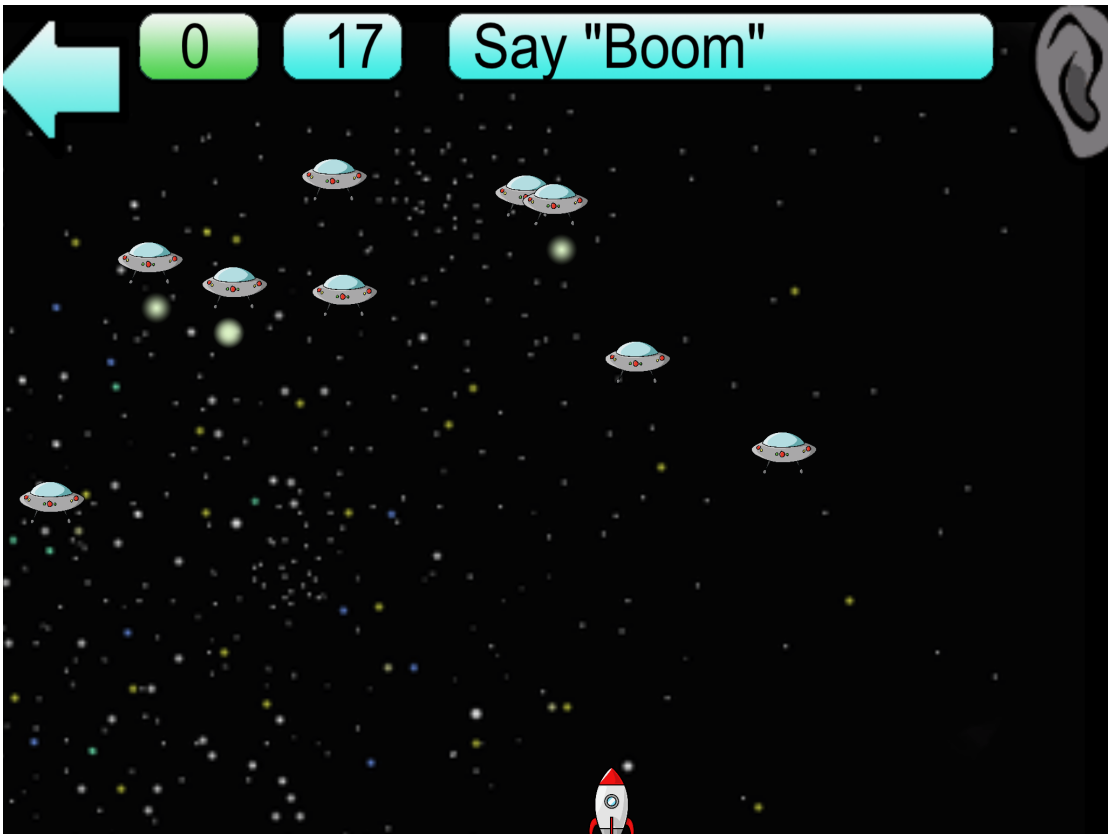

Figure 4.7: Space Invaders, One Game with non time-based win and lose conditions

As stated in 4.3.1, the introduction of RapidEars permitted us to use non-speech based sound effects freely without interfering with the performance or overall behavior of the speech recognition engine. We added in sound effects for games to increase the level of interaction.

We also included a chime that plays whenever the player says a target word or phrase correctly in order to provide additional positive feedback. Playtesters unanimously approved of the introduction of both per-level and meta sound effects.

Given our difficulties during the pilot test with a narrator, the requirement that the narrator sound similar to the player, and the difficulty of working with RapidEars and speech-based sound effects, we elected to remove the narrator entirely. Instead we provide two forms of instruction via text: a transition level that states the level name and provides a short explanation of how to play the game, and progressive instruction using the phrase box. Every time the player completes an action the phrase box provides the next action the player must do or the phrase the player must say.

The primary difficulty with mini-games is the amount of games that need to be created. We implemented procedural content generation routines to take dictionaries and their corresponding objects from the available assets and integrate them into a new game. A variant of the cow game shown uses balloons from the second level of the original game. In this game players must say "Eat Balloon" or "Eat a Balloon" in order for the balloon to pop. The game responds with a combination of the moo sound effect as well as the balloon popping sound effect.

We organized games into collections based on the consonant they target. 4.8 depicts the organization. In preparation for complete coverage of cleft lip and palate plosive impairments, we created screens where SLPs can turn on and off different consonants. Turning them off will remove their associated games from play. 4.15 shows the screen for high pressure consonants. We did not cover all of the consonants, nor all of the difficulties.

Proper game randomization became important during playtesting. Testers complained that they frequently played the game consecutive times and were more likely to skip a level due to playing it too frequently rather than disliking the level itself. Initially the next level was selected when the previous level ended without keeping any state. We found the best level selection when we scrambled all of the levels together into a list, ensuring each level was played at least once before repeating.

We developed a total of eight usable games. Six of the games do not have any extra analysis and are intended for basic sound and word production. Extra analysis includes spontaneous production (saying the word correctly within a small timeframe) and rhythmic production (saying the word correctly multiple times at a specific pace). Games at the word level are intended for initial production; games with extra analysis begin at the phrase level. However,

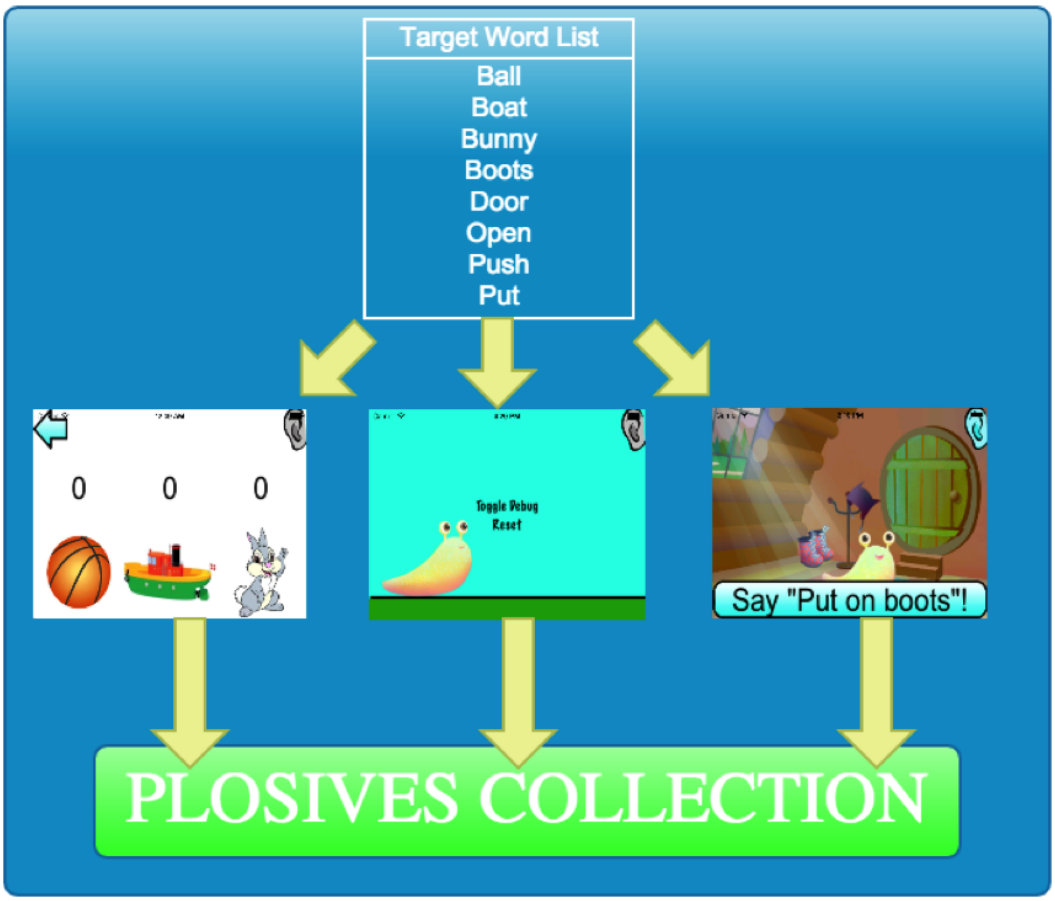

Figure 4.8: Game Organization

dictionaries are modular enough that word difficulty dictionaries could be used in phrase difficulty games. 4.3 lists the games, the difficulty level they were designed for, and any extra speech parameters included in the level design.

Game Name	Difficulty Level	Extra Parameter
Feed Mr Cow	Word	
Duck Duck Goose	Word	
Fireworks	Word	
Memory Match	Word	
Wheel Spinner	Word	
Space Invaders	Word	
Flappy Slug	Phrase	Spontaneous
Bouncy Balls	Phrase	Rhythm

Table 4.3: Developed Games

A scoring system was implemented based on the discussions with SLPs. Initially we planned to provide points for children staying in the game, but early playtesting revealed that this could lead to cheating. As a result we removed time-based points from gameplay. We implemented two different scores: an game score that resets per-level and is composed of a combination of pronunciation and any extra analysis the game is checking for. Players earn 1 point every time they say a target word or phrase.

As we have no information about cleft performance or scores in our game compared to a speaker without impairments, we used a 10 minute playtime as our primary condition for completing the game. When the user has spent at least 10 minutes in game over a 24 hour period, the timer box will turn green and will have a checkmark on it. Players can continue to play after the 10-minute goal, but we predict that players will not play the game for more than the target time after the first few sessions due to the small game selection.

Early playtesting revealed that players could leave the game running without saying anything which would still register as valid playtime on the system. Inside each game we required an initial input in order to start the game and the timer. We used two forms of initial inputs: touch and speech. Touch requires users to tap or drag an in-game object in order to start the timer. Speech requires the user to say a specific phrase in order to start the timer. By using different initial inputs in games we ensured that unintended play would be minimal. Additionally the statistical data collected would inform us if players actively skipped games by providing the initial input and doing nothing else. We solicited expert advice from human-computer interaction researchers. We selected 10 users and had them play the game the game for at least 5 minutes; the average speech rate was 17.2 target words per minute (SD = 1.92) and the fastest rate was 25 target words per minute. By comparison, Speech Adventure had a maximum speech rate of 10.5 targets per minute due to design limitations.

We adapted the old game for use with the new ASR engine, and converted it into a special mode called 'Storybook'. In Storybook mode players have no time limit, and sentences are recorded into the game as well as score. Due to the slow speech rate in those levels and a lack of a time limit we added 1 second onto the playtime counter for every correct sentence spoken in the storybook games.

We provided new HUD elements and decreased the screen real estate overall. We added time and score boxes, as well as a back button. We moved the phrase box to the top of the screen, and shrunk it to a third of its original size. The listening indicator provided enough

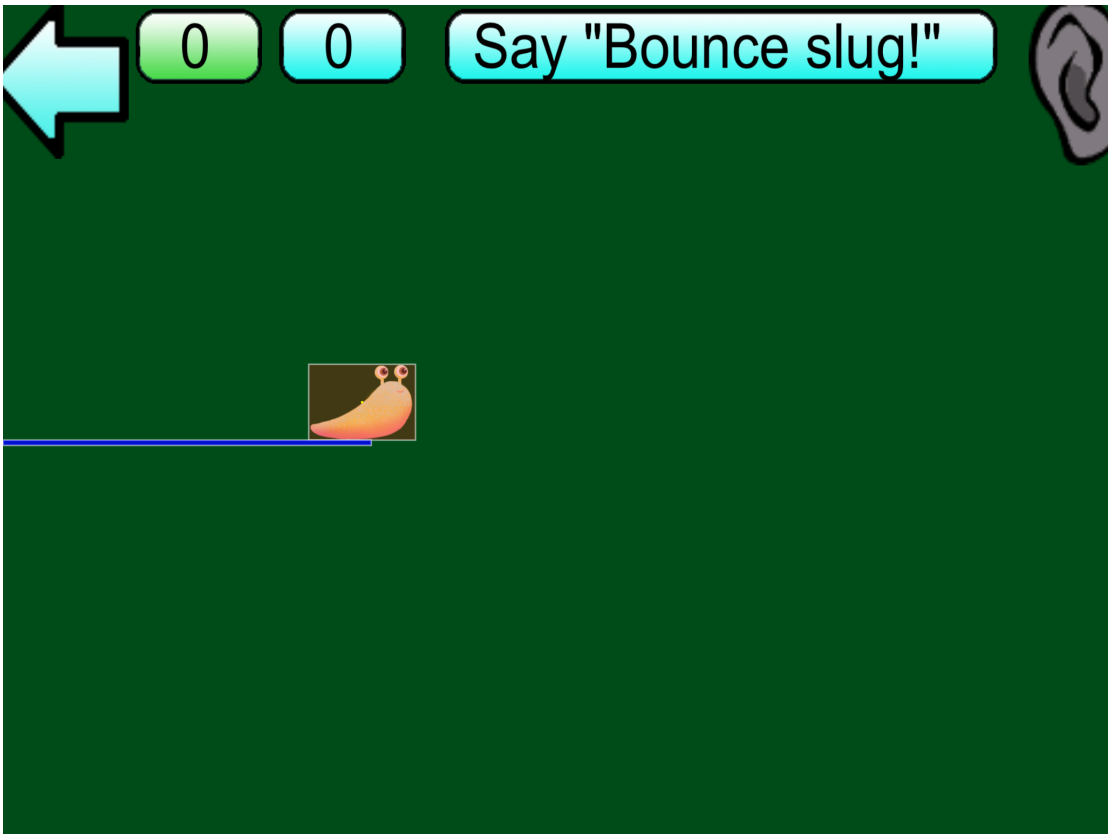

Figure 4.9: Flappy Slug Demonstrating Speech Initial Input

feedback for play inside a doctor's office, but a loudness indicator would be necessary for more general environments. Additionally, the short delay caused between starting speech and the indicator changing caused some confusion. We added additional states to the listening ear: the ear changes color from grey, green, yellow, and red to represent the loudness. As we previously used yellow to indicate that the recognition system is paused, we introduced a red X over the grey ear for when the system is paused. The introduction of the loudness indicator also repaired an issue with a short delay in the indicator changing.

4.3.3 Statistics

The improvements to the statistics can be divided into three sections: SQLite3 implementation, data analysis, and data visualization. In the SQLite section we designed the database layout and the interconnections between the databases. In the visualization section we combine the data into an appealing and intuitive graphical display using open-source frameworks. In

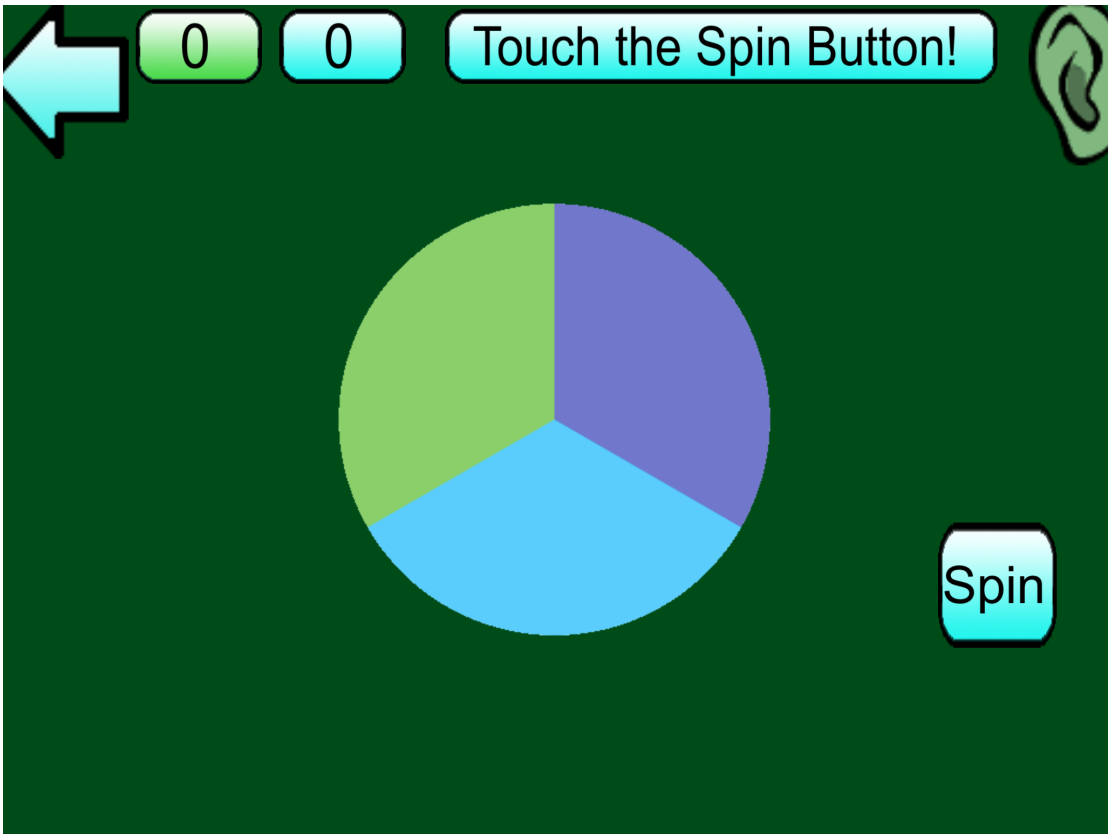

Figure 4.10: Wheel Spinner Demonstrating Touch Initial Input

the data analysis section we analyze the database to combine data into days, extract useful information about play, and input it into the graphical display.

The original statistics system stored the information in a text file, and then uploaded the text to UCSC's private servers when complete. In this version therapist uses the statistics collected by the application in combination with their evaluation of the child's speech in-office to determine the direction of therapy as well as if the application was successful in speech practice. In order to make this possible we decided to build a completely new database based on our interviews with SLPs.

4.12 shows the relationship between the databases. On the high level we wanted to determine if children chose to break up their speech therapy practice and if it was beneficial. We started our database structure with a high level 'Session' table. Touching the start game button creates a new session in the database, and pushing the back button closes the session, finishing up all data associated with that session and closing the database. The session also contains the time played, number of levels played, and the game score. We wanted data about

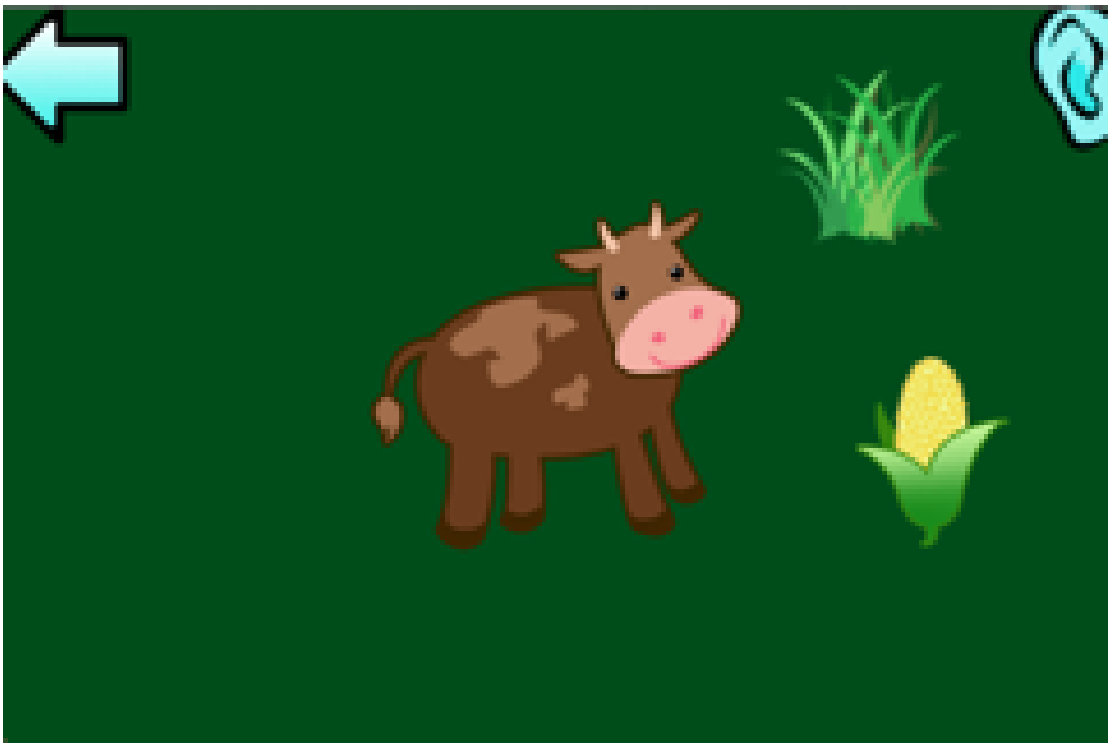

Figure 4.11: Real-Time Speech Game Example

each individual level in a session to determine what levels, if any, players actively skipped. A level table was created, associating with a corresponding session. This records the time played and score achieved for each level. Each sentence has an individual entry associating it with a level giving us an idea of the total sentences spoken, which allows us to separate actual sentences from the overall score. A sentence entry is added anytime a correct sentence is said and the game reacts. From discussions with therapists we determined that the smallest 'grain' of useful data is the raw output of the speech recognition engine.

SLPs needed a way to visually understand the data without any additional training and without taking too much therapy time. We decided that line graphs provide the quickest way to determine performance, and searched for existing iOS frameworks that could supply us this capability while still integrating with Cocos2D. We found Coreplot to meet our needs and selected it for use. 4.13 shows the resulting plot screen. From the discussions on statistics we determined that we would need metrics other than time played per day. We included options to sort the data also by score or sentences spoken.

Selecting a point reveals additional information about the player's performance that

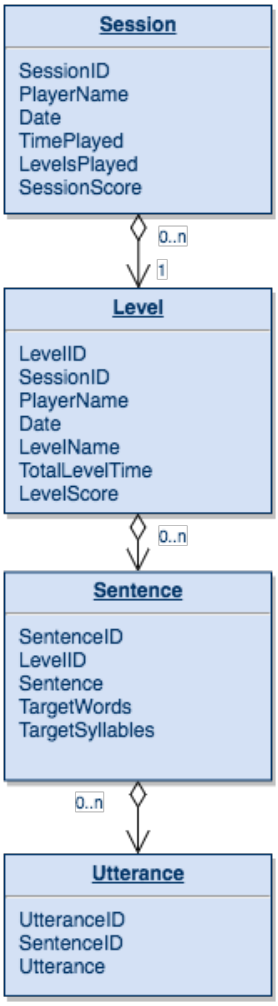

Figure 4.12: Database Relationship Diagram

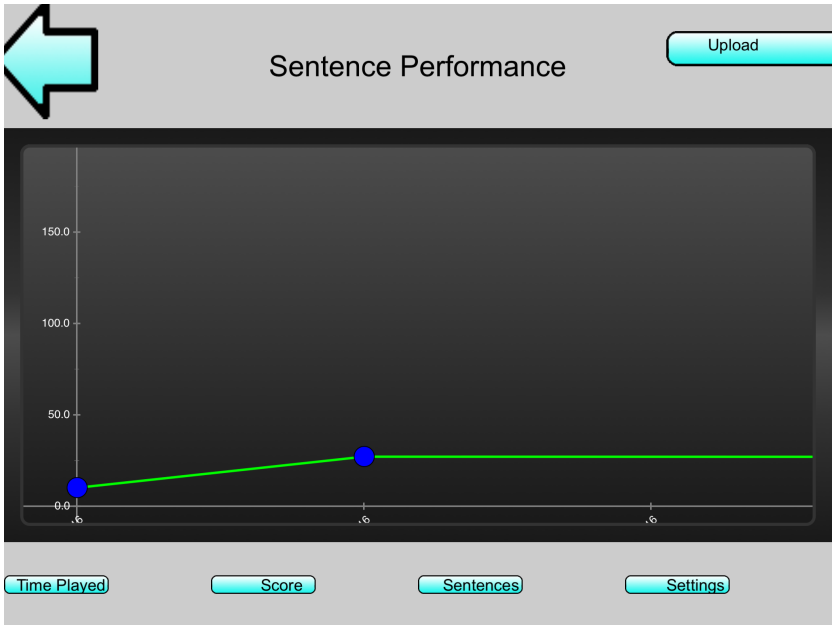

Figure 4.13: Sentence Statistics Example

day. Therapists can view sentence, score, and the amount of time spent in game that day. SLPs can also select any of these as the Y-axis. The data points also provide an additional piece of information: the total number of levels played. This information would be more useful to researchers in understanding game progression and determining the number of levels played on average in 10 minutes. The statistics database itself did not provide a day table to aggregate sessions. We developed a system to combine data. We first used it to combine sessions from the same day. This enabled players to separate therapy into multiple sessions throughout the day, providing a new potential into therapy through using the technology.

Inside the statistics screens of the app we also included a high fidelity prototype of a syllable editor. In the syllable editor therapists can enable and disable different consonants, which will enable or disable the corresponding games and dictionaries in the system. This screen can also report to the therapist the difficulties the game has set the patient to for various consonants. The form of instant visual feedback will aid therapists by allowing them to enable multiple sound sets at once to maximize play, variety, and an even spread of progression or have a child practice a specific sound set to make rapid progress.

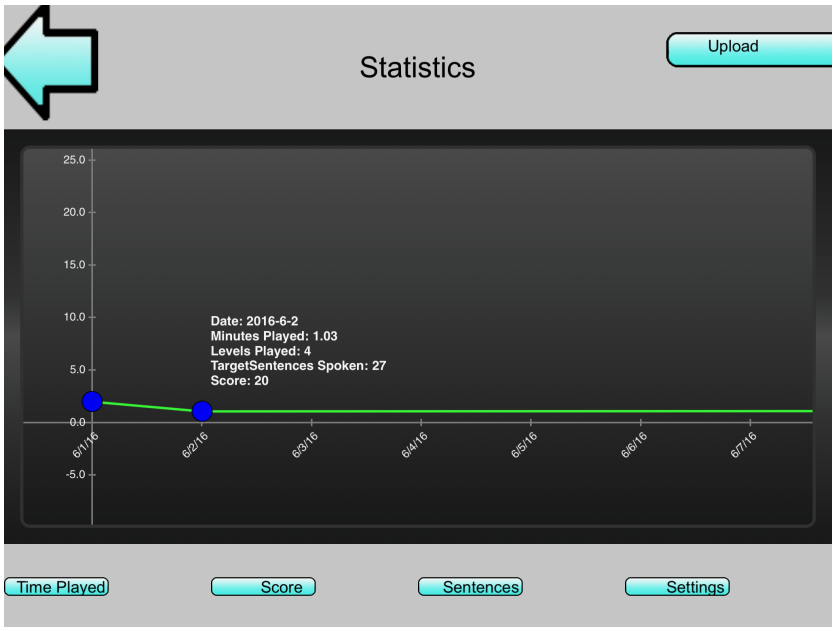

Figure 4.14: Day

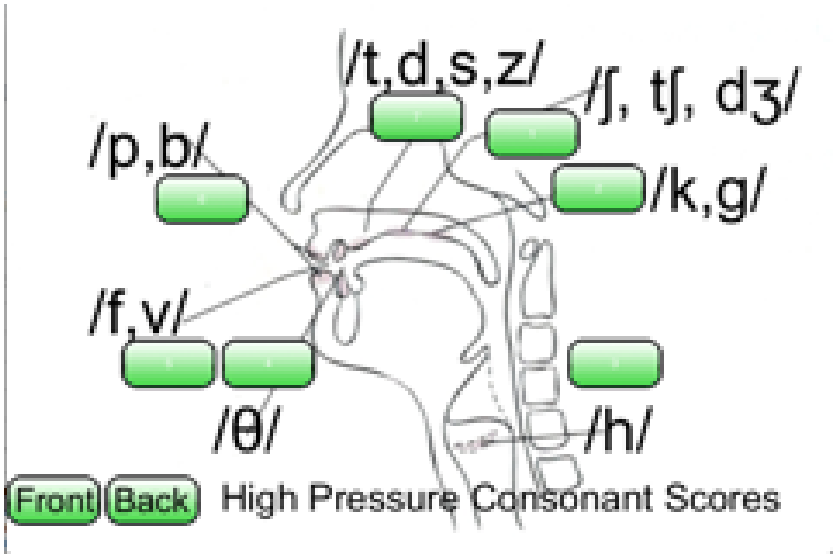

Figure 4.15: Syllable Editor High Fidelity Prototype

4.4 Studies

Though the tools were designed for a specific user group, extensive playtesting revealed that the overall game was enjoyable enough to play that it merited investigation with other audiences. As we recruited children for our long-term study we also began evaluating the game for broader speech motivation and practice. We also made the system self-contained requiring no intervention from therapist, researcher, or programmer in order to run or play. As a result we could run as many studies as we had devices.

We began interviewing speech pathologists in the Summer of 2016. We developed two groups of study. The first group was a pilot test involving speech therapists who were enrolling children in their program for the first time. The purpose for this group was to determine if the system made children more motivated and more likely to pursue therapy. The second group included developmentally disabled adults who had a speech impairment, and had been enrolled in therapy for an extended period of time. The purpose of this group was to investigate the practice behaviors of individuals who have been enrolled in therapy for beyond the normal amount of therapy time, as well as determine if practicing made any impact at all on their speech performance.

Individuals with developmental disabilities require more intensive and advanced intervention, and must receive specific education and training for these individuals. The SLP's adopts a variety of additional roles: they evaluate compensatory communication strategies, provide instruction on broader social functioning, and support individuals throughout their lives. Early intervention becomes even more critical, as impairment correction takes longer [7, 55, 54, 45, 33, 24]. We designed the games to be playable by children as well as adults, however the motivational aspects focused on children. We did not know how older individuals would react to the design.

4.4.1 Pilot Test: Children with Cleft Palate

We began recruiting young participants with cleft lip and palate at the University of California, Davis medical center in July of 2016. We provided 5 iPad Mini 2s to the craniofacial surgery team there to provide to children that enroll in the study. Therapists distributed out the devices to children in three different categories: those enrolling or re-enrolling in speech therapy for cleft, those presently enrolled in cleft speech therapy, and those who are no longer

in therapy either through completion or attrition. Within each group we have different focus points.

We will provide the game for short-term testing with children who are enrolling or re-enrolling in speech therapy for cleft lip and palate. The child will play the game during the speech evaluation done by the SLP. At the end the SLP will upload the game database to a secure location on UCSC's web servers. Researchers will compare the intelligibility scores provided by the SLP to the in-game scores provided by the game to determine if there is a correlation. This will allow them to potentially condense therapy further for patients who are more intelligible.

Children who are already enrolled will have their at-home therapy component replaced for one week when the plosives under practice match the target sentences in the video game. We will investigate their performance over the week, seeing if they play for the 10 minutes per day. If they do not we will investigate if they achieve a certain level of performance before stopping play. For children that do play the game we will learn about how the game affects speech during play, and potentially allow SLPs to condense therapy by providing them performance targets rather than time targets.

Post-therapy are invited to play the game for up to a week. These participants will serve as expert playtesters with the game. Primarily we hope to receive their feedback on how to make the game more fun, and if they would choose the game if they were offered it during their time in therapy. We will also acquire another gold standard through their play, seeing how children who speak post-therapy perform in-game.

There are some early results with children SLPs evaluated the system live by providing it to a patient re-enrolling in speech therapy. The SLP reported that she was "very impressed at the patient's attention and increased effort during the game, vs his effort in speech therapy". The SLP reported that the patient played the game for over 5 minutes, stopping only because the session ended and he had to leave the therapist's office. These early results suggest that the system can keep the child motivated for at least half of the required daily time in the first session. The SLP reported that the BouncyBalls game had a low recognition rate, making the game unplayable even for the SLP.

4.4.2 Pilot Test: Developmentally Disabled Adults

Individuals with cerebral palsy (CP) struggle with conditions such as dysarthria, dysphagia, and dyspraxia as they speak. While speech therapy is successful in practice, outside practice requires increased commitment and effort from caregivers. We developed a speech recognition game designed to encourage out-of-office exercises and motivate users to practice. Next they recruited a participant with cerebral palsy to investigate the performance of the system in a live environment. The participant joined the game after demonstration from the caregiver and temporarily increased speech loudness and clarity during play. The participant found sound effects more rewarding than animations. The total number of sentences spoken during the session was found to be less than half that of a speaker without any impairment. We also observed two instances of cheating.

Researchers at the University of California, Berkeley investigated computerized speech recognition of individuals with dysarthria caused by cerebral palsy. Nine athetoid cerebral palsy speakers and one spastic speaker took part in a test investigating the performance of the Shadow/VET voice entry system. Researchers found that dysarthric speakers had a similar pattern of correct recognition to non-disabled speakers but the correct recognitions of dysarthric speakers was less than half that of non-disabled speakers. The researchers noted that the low recognition accuracy remains a serious problem [10].

A graduate and undergraduate student developed a total of two levels. Five levels intended for plosive speech practice had their dictionaries replaced with command and control dictionaries containing fewer than ten words. Words were translated into their ARPAbet equivalent via the lmtool provided by Carnegie-Mellon University. These dictionaries were put into a rotating system of a total of seven mini-games. The game engine cycles through the games randomly, making sure each game is played at least once before repeating. Each level's time was set to 20 seconds, if the user does not finish the level in the allotted time the game moves to the next level. The system provides a timer for each level as well as on-screen prompts to guide the user what to say. As the user speaks the on-screen phrase highlights as the user speaks. If the user speaks the word or phrase correctly, an action occurs such as a sound effect or an animation.

We evaluated the system on four native English speakers with no speech disorders. Three females and one male aged 21-27 took part in a playtesting session lasting a half hour. The mean speech rate in game was 17.2 words (SD = 1.92) per minute of playtime. Next under-

graduate students from the University of California, Santa Cruz recruited a 34-year-old female participant with spastic cerebral palsy as part of a course introducing beneficial technology to disabled groups. The participant had previous experience playing a variety of games on her own tablet. Testing took place in a quiet indoor location on campus. The participant's home support was present in place of the primary caregiver.

Initially the participant was reluctant to participate. The home support motivated the participant by playing tutorial levels. On the second in-game sentence the participant attempted to play, but let the home support finish the sentence. The participant began playing the game on her own on the second level of the tutorial and played along with the home support for the remainder of the levels as they played the main game together. The participant reacted positively to all in-game sound effects. During play of the main game (levels 3-10), the home support and observers noted a significant increase in the participant's pronunciation and volume.

We witnessed the participant cheating in the game on two separate occasions. Cheating in this context was defined as any attempt to gain points or progress in the game without saying the on-screen sentence. In the first instance the participant exploited the caregiver's verbal motivation for all of game level seven. The participant would attempt to have the caregiver say the word loud enough so that the recognition engine would respond to the caregiver. In the second instance the participant could sometimes trigger events in the game through any vocalization. This began on level eleven and continued until the game concluded.

The game reported that the participant played for 4 minutes. During that time they played 12 levels and spoke 34 valid sentences, for a speech rate of 8.5 sentences per minute. Of those 8 were utterances from the participant; the remainder came from the home support. Nine of the twelve levels played were unique and three were repeats. The participant expressed disinterest in the game at level ten, and was unable to complete any level before time ran out. The home support reported that the participant enjoyed the game overall, but grew bored primarily due to the repetition of the games.

4.4.3 Long-Term Study: Developmentally Disabled Adults

We began recruitment of developmentally disabled adults for a long-term study in Winter of 2016. We distributed 5 iPad Mini 2s to various developmentally disabled adults who meet regularly with the speech therapist. We met with users in November for a demo, and then adjusted the game side based on user responses. Following the demo, we provided the

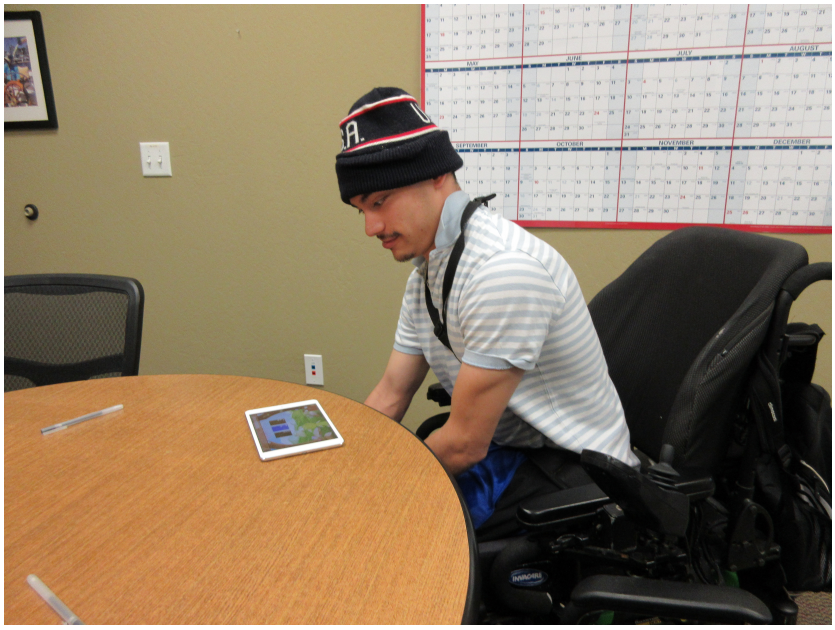

Figure 4.16: A participant uses the app during the initial test

participants the iPads for a month-long test. We conducted qualitative pre-and-post interviews with participants and the SLP.

We first met with the participants in November of 2016 to conduct a pilot test of three different speech games developed under the tool suite. This included the Speech Adventure storybook game, the current iteration of the tool and a variant of Speech Adventure (the game used in the pilot test) titled SpokeIt. Participants requested the mini-games move slower, have more sound effects, and have voice prompts. Researchers increased the level lengths from 20-30 seconds to 40-60 seconds, added more sound effects to each level, and added voice prompts for each level. We then selected five participants from this pilot test to participate in our long-term study.

We selected 5 individuals with mild-to-moderate mental retardation (MR) or Downs syndrome who had a speech impairment the game could detect and correct. We collected gender, age, mental age, disability, and speech impairment. Mental age was calculated from their intelligence quotient (IQ) result as determined by their diagnosis. Interviews with the SLP confirmed that she currently instructed participants to practice 10 minutes a day at home, and that she did not know the amount of time the participants spent practicing on their own. She was aware that participants were not spending the requested amount based on their progress in the office. We investigated the game's motivating effect and amount of time put in outside of

Participant	Gender	Age	Mental Age	Disability	Speech Impairment
Participant 1	M	32	7	Downs: Moderate	Hypernasality, Consonant difficulty
Participant 2	F	32	7	MR: Moderate	Substitution: W for R
Participant 3	M	30	9	MR: Moderate	Low Volume
Participant 4	M	36	9	MR: Moderate	Low Volume, Slow speech rate
Participant 5	F	48	10	Downs and MR: Mild	Low Volume

Table 4.4: Details of participants recruited for the study

therapy. We set the gold standard at 10 minutes where any day a participant spends at least 10 minutes in-game meant the game successfully motivated the individual. Additionally, we investigated the efficacy of the graphs provided to the therapist. The SLP informed researchers that the participants are not interested in improving their impairment, and that the impairment is very ingrained in their speech. As a result we did not predict the game would improve the speech characteristics of the participants. We gave users a total of 6 weeks with the app; we hypothesized that the majority of users would drop out within 2 weeks, and no users would remain after 4 weeks.

After a one month period we retrieved the iPads and analyzed the results of the data as processed by the speech and statistics engines. We found participants were bursty, with increased effort shortly after a therapist meeting. We hypothesized that users might play past 10 minutes for the first game session, but hypothesized that participants would play less than 10 minutes per day throughout the month. 4.17 exemplifies the behavior. This indicates that participants found the game initially motivating enough to play beyond the daily requirement, although this was not sustained.

Participant five was the only one that produced the hypothesized attrition line. Her drop-off occurred over 3 weeks rather than the predicted 2 weeks. 4.21 shows her resulting graph. The other four participants played in much more variable amounts as their performance graphs revealed.

The system provided three graphs to view the data: time, sentences spoken, and score. In the time graph we found the daily time participants spent to be highly variable: over 23 days of play participants played an average of 7.3 minutes/day with a standard deviation of 8.9 minutes. 4 out of the 5 participants spent one or more days playing the game for more than 10 minutes. Of the 7 days with over 10 minutes of play the average playtime was 21.8 minutes with a standard deviation of 7.8 minutes. These 7 days represent 30.4% of all days where the

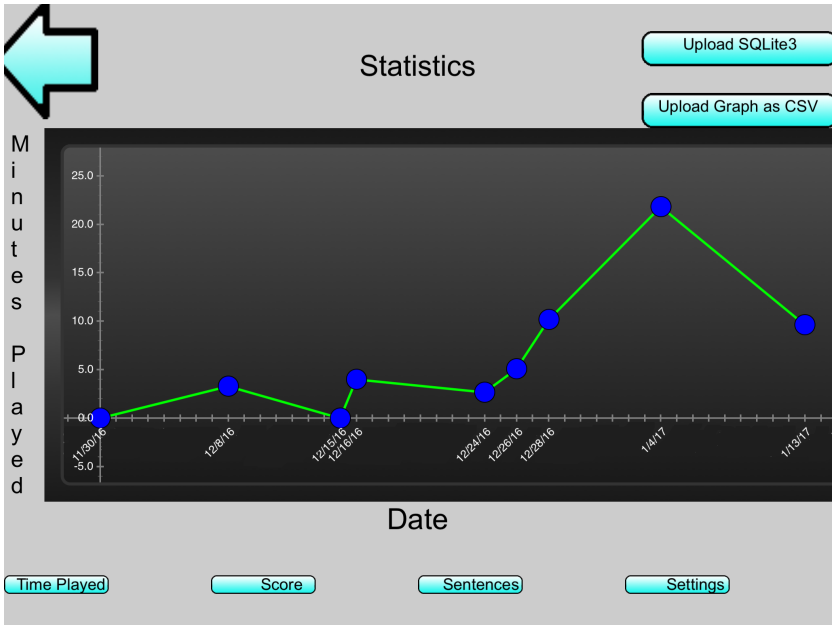

Figure 4.17: Participant's play. Each dot represents a day when the participant played. Note the breaks in between play periods, and periods past the 10 minute requirement

Pariticipant	Days of Play	Total Time (Minutes)	Total Sentences
Participant 1	7	60.2	534
Participant 2	9	46.4	3
Participant 3	4	17.22	48
Participant 4	4	3.63	5
Participant 5	3	36.23	96

Table 4.5: In-game results of participants

game was played. 4.5 shows the in-game performance of each participant. 4 of the 5 participants (1, 2, 3, and 5) agreed they spent more time practicing outside of therapy than before, these participants were also the ones who spent more than 10 minutes in a single day.

The SLP reported that the time graph provided the most useful feedback for outside-office practice. When compared with the sentence graph it aided the SLP in finding the specific failure point outside of the office. As an example the SLP noted participants two and four, who have made similar progress in therapy. However, the system revealed the differences in behavior outside of the office between the two. Participant 2 put in the second-highest amount of time in of all the participants, but maintained a low overall in-game score indicating a motivated individual with a problem in the exercises. Participant 4 put in the lowest amount of time out of all of the participants, showing the therapist that the issue lay with the participant's personal

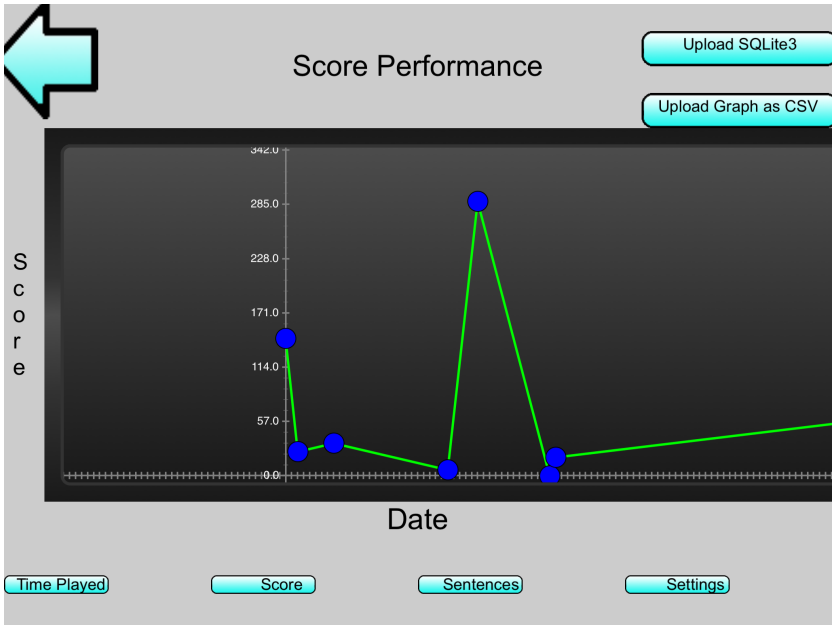

Figure 4.18: Participant's score data showing widely variable performance. The participant's time performance was similar.

motivation first.

The sentence graph was useful when the participant was not making progress, but having a low sentence performance despite playtime informed the SLP that the current at-home exercises were not working. However, the sentence graph was not useful with 3 of the 5 participants who had a high time variability and who could perform the exercises. In our game design we hypothesized that participants would not play very far past the 10-minute requisite time, and we predicted that the play time of participants would decrease over the month. During testing we found users frequently played beyond the requisite time. The SLP reported that the score graph did not provide any useful information, as it correlated almost directly with the sentence graph that already correlated closely with the time graph for these participants. The score graph was intended to show subtler performance metrics and to separate basic and more complex speech production. However both sentence and score were too influenced by time, and thus the score graph did not show the nuanced metrics. We replaced the score graph with a speech rate graph that graphs the number of sentences a user speaks per minute in-game. This graph provided a new metric, enabling researchers and the SLP to look at a metric that provided consistency across days independent of play time. 4.19 shows a section of the speech rate graph for participant one.

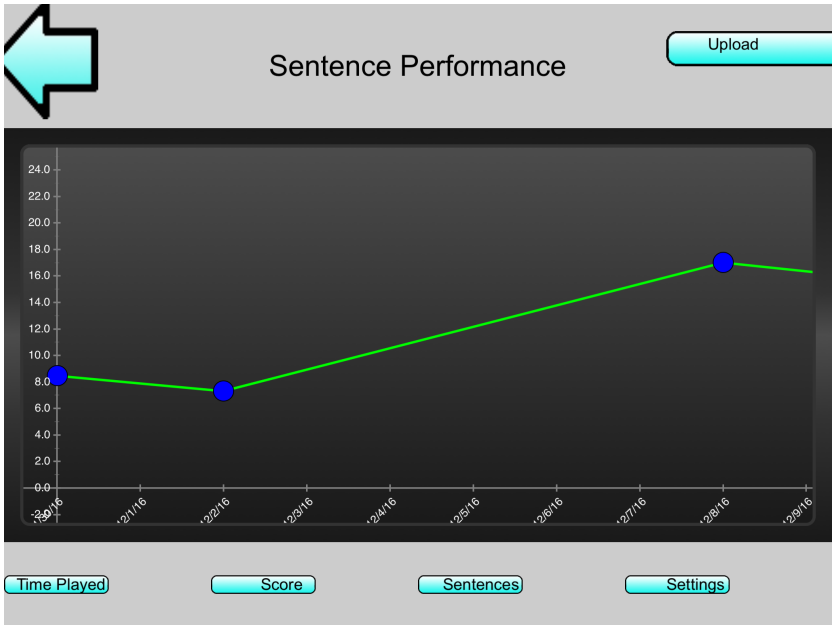

Figure 4.19: Participant Improvement Over Nine Days of Therapy

Participant one experienced the greatest motivating effect. Additionally all but two of the game levels contained target words for the participant's particular impairment, resulting in him experiencing mild improvement in his speech. The SLP confirmed that participant one increased his sentence length, meaning the participant grew more comfortable forming and saying longer sentences. Lack of harder difficulties with longer sentences meant the overall benefit was miniscule - increasing his sentence rate from 8.4 sentences/minute at the start to 9.2 sentences/minute at the end of the study. Participant one played for a total of 60 minutes over the 1-month period, which is less than one-quarter the amount of time requested by the therapist. 4.19 shows the speech rate graph over a nine day period. This shows the participant making large improvements over the 9-day period. 4.20 Shows the full month of practice. As expected performance decreased when the participant stopped playing for a few days, but then increased again when play resumed. With therapy resuming the participant hit the highest consistent performance - peaking at an average sentence rate of 9.2 sentences per minute over 33 minutes of play.

Participants 1, 2, and 5 made use of the multi-session capabilities of the app, particularly on days where play time exceeded 20 minutes. Participant one made n one particular day participant one split gameplay into 3 separate sessions: one lasting 13 minutes, one lasting 5 minutes, and a final 15 minute session. The speech rate for each session was 8.2 sentences/minute,

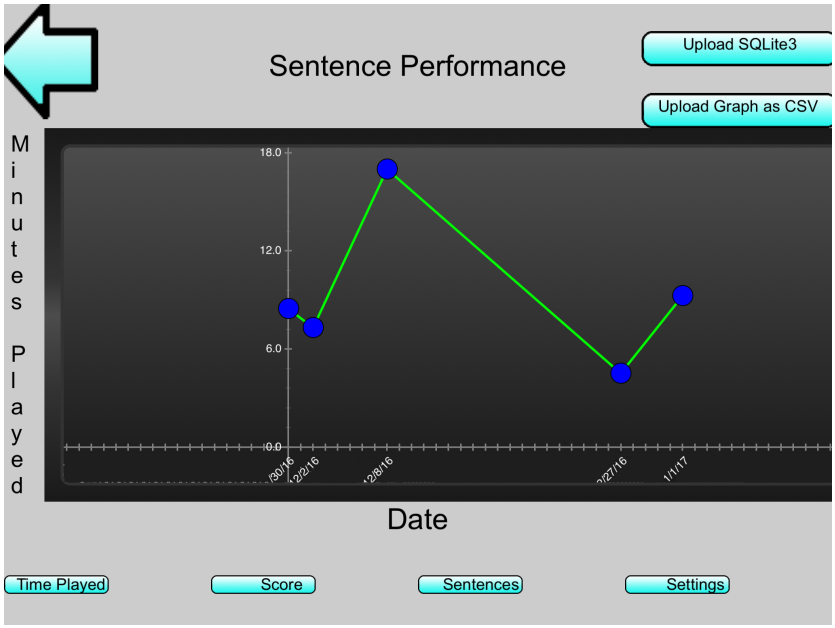

Figure 4.20: Participant's Improvement Over Month

9.2 sentences/minute, and 8.7 sentences/minute respectively. The speech rate improved during play, and the participant achieved a higher average speech rate than when play began. Participant one not only demonstrated an improvement, he retained it in a later session that day.

Participants played the most after visiting with the SLP, indicating that the meetings produced the greatest influence. 3 out of 5 participants played the game after meeting with the therapist in person on December 8th, 2016 with participants 1 and 5 playing the game for more than 10 minutes on that day. The behavior was similar the next week, with 3 out of 5 participants again playing the game between the 15th and the 16th. However, this time no participants played beyond 10 minutes, demonstrating the playtime decrease we hypothesized. On December 16th participants 4 and 5 quit playing after 17 days of play. On December 24th participant 3 stopped playing after 25 days with the app. Participants 1 and 2 continued to play until the end of the study playing 3.52 minutes and 9.62 minutes respectively during the final week. One participant was playing for the required time at the end of the study, 40% of participants played to the end of therapy, and 60% of participants played beyond the hypothesized 2 week drop out.

Participants 2, 3, 4, and 5 reported difficulty playing the more complicated games (i.e. Flappy Slug). Players 2, 4, and 5 reported difficulty playing simpler games (i.e. Space Invaders). In post-interviews participant two requested music, and participants two and three requested

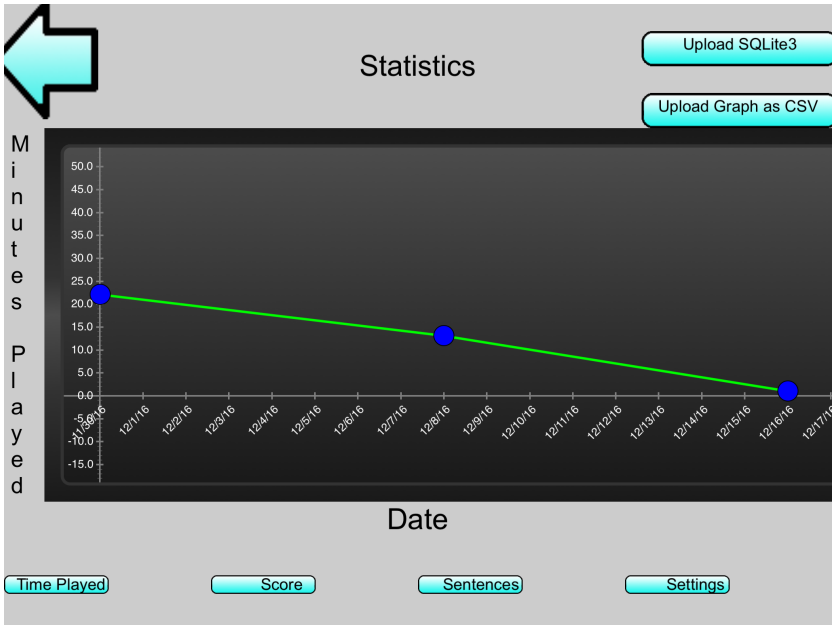

Figure 4.21: Participant Drops Out After 3 Weeks. Researchers predicted most participants would drop out after 2 weeks.

Target Word	Arpabet
Rabbit	R AE B IH T
Rabbit (Cleft)	R AE HH IH T
Rabbit (W for R Substitution)	W AE B IH T

Table 4.6: Dictionary Adjustment Needed for Participant 2

additional sound effects in games. Participants 4 and 5 reported that the games were determined to be too fast-paced. Participant 5 requested a way to mute the game's sound effects.

Follow-up discussions with the SLP found that the game was accurate at grading, but the lack of configurability on the therapist side prevented the system from achieving greater success. The SLP stated the greatest problems with the tool were the lack of target word customization and mispronunciation configurability. While many of the levels contained target words for the participants, the engine was configured for cleft mispronunciations. 4.6 shows an example of a dictionary change needed to match participant two's specific speech impairment. Similar to participant one, the lack of variable difficulty and more targeted dictionaries also hindered progress.

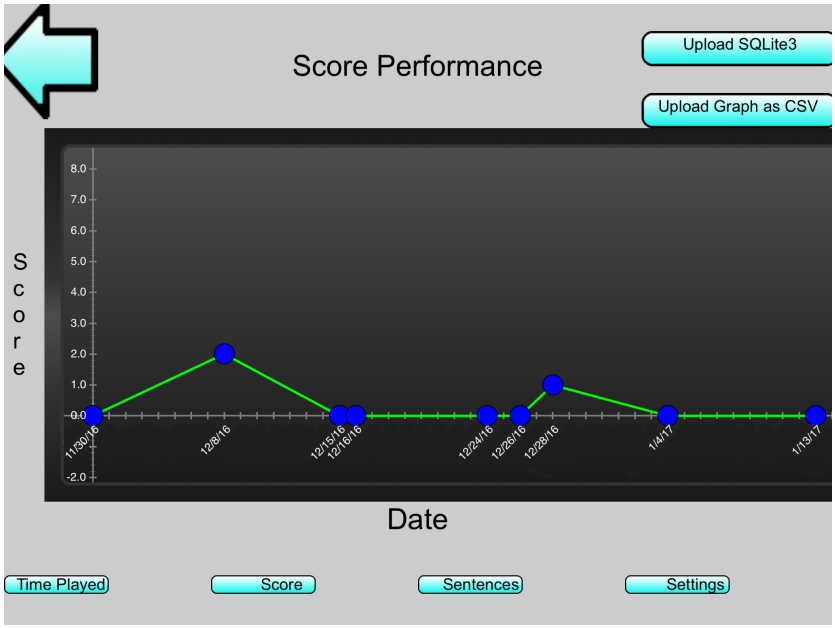

Figure 4.22: Participant plays for multiple days unsuccessfully. The participant played for a total of 46.4 minutes over the course of the study only successfully saying 3 target words.

4.5 Future Work

There remain a number of unanswered questions at this point in the therapy and a number of directions that the work can continue along. Before any changes to software are performed, the system is capable of testing and evaluation with a variety of groups. On the speech recognition front tweaking models produces a variety of benefits, and dictionaries of the hardest difficulty are not developed yet. On the game side, adding more mini-games increases the replayability and reduces the rate of the game becoming stale resulting in an increase in attrition. Finally, feeding the data analyzed on the statistics side back into the game side would allow the system to automatically adjust the difficulty and phonemes potentially shortening therapy times.

The average score spread for children without impairments is not known. This provides us with a gold standard for the game, enabling us to evaluate against an standard speech speed. This piece of information will allow both the game and the therapist to determine if a child has achieved an acceptable speech rate for the target sound. It will also allow a therapist to calibrate the game for faster or slower natural speakers.

This system has the capability to work with virtually any group with a speech impair-

ment. With this capability we have already reached out to disabled groups. Reaching out to more users with speech impairments will allow us to improve the system and increase its audience. It will also permit us to expand out potentially to foreign language learners and people wishing to practice or improve their speech.

Collecting enough speech samples to create an impaired acoustic model provides an interesting and potentially valuable resource. While this is currently not possible by our research obligations, the opportunity may arise in the future. As such OpenEars has provided a plugin called SaveThatWave, enabling us to save utterances after processing. This plugin will also assist when adaptation is used.

The speech recognition system, while mostly designed, can always achieve better performance with better models. With respect to the acoustic model, providing the system with a child acoustic model would yield the greatest increase in performance and accuracy. The system performs adequately without one but may still benefit from adaptation. On dictionaries OpenEars contains a way to dynamically create dictionaries and language models using a string of uppercase words as an input. This would enable the system to adjust the game for different target words and speech impairments without programming. Adjusting the weights inside the language model could make the game easier or harder as the game will attempt to recognize one over the other. Weights in the tools described have all been 50% weight between impaired and non-impaired speech, but weights from a non-impaired child speaker from a pre-specified amount of play could provide another form of gold standard. As the recognition score can assist in determining if pronunciation improves in a short session or over a series of similar short sessions implementing a system that utilizes recognition score as another scoring metric could yield even more performance improvements.

The video game design is presently optimized for the requirements. Adding new minigames improves the novelty of the game and the replayability. Adding accurate mouth placements for the produced phonemes and sounds on the in-game characters would improve imitation and connectivity to the players. Introducing 3D graphics would expand our game design capabilities, though likely at the cost of increased design complexity. Discussions with SLPs suggested that two versions of the game be created: one to test time-based play and one to test speech rate-based play. The speech rate-based play would have the game stop when the user meets a speech rate target as opposed to meeting time requirements. We would need to organize a focus group of SLPs to dicuss the implications of both system. 4.23 shows an example of the

speech rate-based game: when the user hits 15 targets/minute (1 standard deviation down from 'average unimpaired' speech) the user is alerted to their success - if they have no more practice phonemes they are informed they can stop playing for the day.

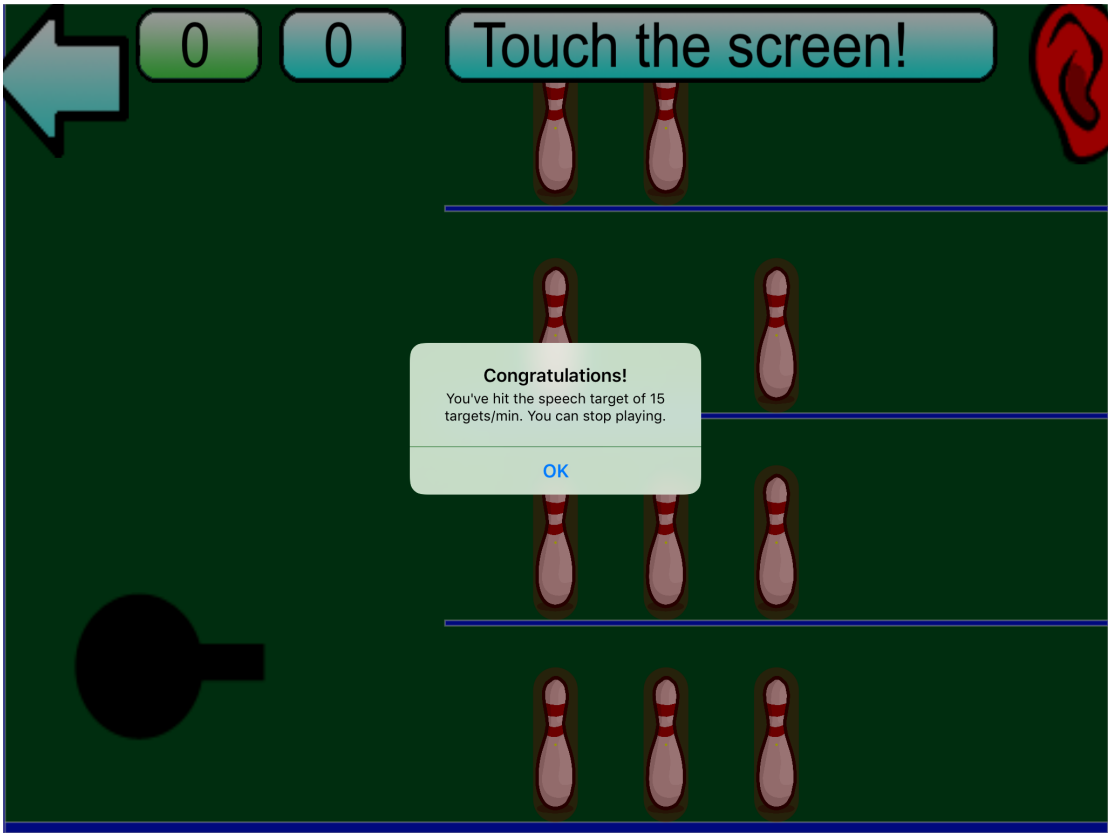

Figure 4.23: Speech Rate-based Game example, showing the participant hit 15 targets/minute

The statistics system has achieved a milestone where the data it collects from the at-home performance will be directly seen and used by the therapist in the following meeting. The next step in the process is to have the system use the statistical information gathered to guide the system into making the games harder, or possibly moving the player onto other phonemes if they achieve a high accuracy at high difficulty in under the expected time.

The system currently exists in the form of an app developed for the iPad Mini 2 coded in Objective-C. The system is designed around modularity, as such is it easy to migrate to different devices as well as entire architectures. The selected speech recognition engine, PocketSPHINX, is both open-source and cross-platform. Frameworks and setup instructions currently exist for Android as well as all desktop and laptop computers, and all three models (acoustic, word, and language) are immediately migrateable between systems without any file

changes. Cocos2D, our game engine, is also open-source and cross-platform. The existing code, written in Objective-C, is directly convertible to C++ as well as convertible to desktop, web, and other mobiles. The code is self-documenting and contains no magic numbers. Finally, the statistics engine communicates with an SQLite3 database; the SQL code is directly portable to any SQL database. The engine also supports exporting the currently viewed graph in CSV format (date, value). This allows SLPs and researchers to import data directly into the analysis tool of their choice.

Bibliography

[1] S. Accardo. Rock band 2 preview. http://xbox360.gamespy.com/xbox-360/rock-band-2-game-only-/890663p1.html. Published 7/16/2008. Retrieved 2/10/2015.

[2] US Food & Drug Administration. Mobile medical applications. http://www.fda.gov/MedicalDevices/DigitalHealth/MobileMedicalApplications/default.htm#c, September 2015. Accessed: 2016-09-23.

[3] ASHA. How does your child hear and talk? http://www.asha.org/public/speech/development/chart.htm, 2012. Accessed: 2016-07-22.

[4] American Speech-Language-Hearing Association. 2016 schools survey report: Slp workforce/work conditions. www.asha.org/research/memberdata/schoolssurvey/.

[5] F. Weber, K. Bali. Enhancing esl education in india with a reading tutor that listens, 2010.

[6] Caroline Bowen. *Children's speech sound disorders*. John Wiley & Sons, 2014.

[7] Nancy C Brady, Susan Bruce, Amy Goldman, Karen Erickson, Beth Mineo, Bill T Ogletree, Diane Paul, Mary Ann Romski, Rose Sevcik, Ellin Siegel, et al. Communication services and supports for individuals with severe disabilities: Guidance for assessment and intervention. *American journal on intellectual and developmental disabilities*, 121(2):121–138, 2016.

[8] Carnegie-Mellon. Adapting the default acoustic model. http://cmusphinx.sourceforge.net/wiki/tutorialadapt, March 2016. Accessed: 2016-09-27.

[9] Carnegie-Mellon. Training acoustic model for cmusphinx. http://cmusphinx.sourceforge.net/wiki/tutorialam, September 2016. Accessed: 2016-09-27.

[10] Colette Coleman and Lawrence Meyers. Computer recognition of the speech of adults with cerebral palsy and dysarthria. *Augmentative and Alternative Communication*, 7(1):34–42, 1991.

[11] Allison Druin. *The Design of Children's Technology*. Morgan Kaufman Publishers, Inc., San Francisco, CA, 1998.

[12] A Danilo et al. Social adjustment in french adults from who had undergone standardised treatment of complete unilateral cleft lip and palate. *The Cleft Palate-Craniofacial Journal*, 43(5):598–605, 2006.

[13] Andreas Maier et al. Intelligibility of children with cleft lip and palate: Evaluation by speech recognition techniques. *Proceedings of the International Conference on Spoken Language Processing,*, 4, 2006.

[14] Andreas Maier et al. Automatic evaluation of characteristic speech disorders in children with cleft lip and palate. *International Conference on Spoken Language Processing*, pages 1757–1760, 2008.

[15] Andreas Maier et al. Automatic detection of articulation disorders in children with cleft lip and palate. *The Journal of the Acoustical Society of America*, 4:2589–2602, Nov 2009.

[16] C. Munteanu et al. Alex: Mobile language assistant for low-literacy adults. *MobileHCI*, pages 427–430, 2010.

[17] G.A. Korsah, et al". Improving child literacy in africa: Experiments with an automated reading tutor. *ITID*, 2010.

[18] Ling He et Al. Automatic evaluation of hypernasality and speech intelligibility for children with cleft palate. *8th Conference on Industrial Electronics and Applications*, June 2013.

[19] Ling He et Al. Automatic evaluation of hypernasality and consonant misarticulation in cleft palate speech. *Signal Processing Letters*, June 2014.

[20] Ling He et Al. Automatic evaluation of hypernasality based on a cleft palate speech database. *Journal of Medical Systems*, March 2015.

[21] O Hunt et al. Self-reports of psychosocial functioning among children and young adults with cleft lip and palate. *The Cleft Palate-Craniofacial Journal*, 43(5):598–605, 2006.

[22] Cleft Palate Foundation. Speech development. `http://www.cleftline.org/publications/speech/`, 2012. Accessed: 2016-07-22.

[23] Chaim Gingold. What warioware can teach us about game design. `http://www.gamestudies.org/0501/gingold/`, May 2005. Accessed: 2013-05-13.

[24] Howard Goldstein. Language intervention considerations for children with mental retardation and developmental disabilities. *Language Learning*, 2006.

[25] Edie Hapner, Carissa Portone-Maira, and Michael M Johns. A study of voice therapy dropout. *Journal of Voice*, 23(3):337–340, 2009.

[26] Mary Hardin-Jones, Kathy Chapman, and Nancy J Scherer. Early intervention in children with cleft palate. *The ASHA Leader*, 11(8):8–32, 2006.

[27] Tsung-Yen Hsieh, Jamie L Funamura, Christina Roth, Zachary Rubin, Sri Kurniawan, and Travis T Tollefson. Developing a novel speech intervention ipad game for children with cleft palate: A pilot study. *JAMA facial plastic surgery*, 17(4):309–311, 2015.

[28] Xuedong Huang, Fileno Alleva, Hsiao-Wuen Hon, Mei-Yuh Hwang, Kai-Fu Lee, and Ronald Rosenfeld. The sphinx-ii speech recognition system: an overview. *Computer Speech & Language*, 7(2):137–148, 1993.

[29] K Hufnagle. Therapy techniques for cleft palate speech and related disorders. *The Cleft palate-craniofacial journal: official publication of the American Cleft Palate-Craniofacial Association*, 41(3):340–340, 2004.

[30] David Huggins-Daines, Mohit Kumar, Arthur Chan, Alan W Black, Mosur Ravishankar, and Alexander I Rudnicky. Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE, 2006.

[31] Ann W. Kummer. *Cleft Palate and Craniofacial Anomalies: The Effects on Speech and Resonance, 2nd Edition*. Delmar Cengage Learning, New Albany, New York, 2008.

[32] Zak Rubin, Sri Kurniawan. Speech therapy tool for children to use after cleft lip surgery. *1st Workshop on Games and Natural Language Processing at JapTAL*, 2012.

[33] Thomas L Layton and Mary A Savino. Acquiring a communication system by sign and speech in a child with down syndrome: A longitudinal investigation. *Child Language Teaching and Therapy*, 6(1):59–76, 1990.

[34] Akinobu Lee and Tatsuya Kawahara. Recent development of open-source speech recognition engine julius. In *Proceedings: APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference*, pages 131–137. Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, International Organizing Committee, 2009.

[35] Akinobu Lee, Tatsuya Kawahara, and Kiyohiro Shikano. Julius—an open source real-time large vocabulary recognition engine. *Eurospeech*, 2001.

[36] K-F Lee, H-W Hon, and Raj Reddy. An overview of the sphinx speech recognition system. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(1):35–45, 1990.

[37] J Liu. Hey you, pikachu! review. `http://www.gamerevolution.com/review/hey-you-pikachu`. Published 11/01/2000. Retrieved 1/7/14.

[38] Rose Medical. Speech therapy software. `http://www.rose-medical.com/`. Accessed: 2016-10-13.

[39] Lotte Meteyard, Carol Fairfield, Paul Sharp, Hannah Bettle, Sarah Philpott, and Laura Wood. Testing the ecological validity of an automated procedure for measuring speech intelligibility (icspeech intelligibility scorer). 2014.

[40] Mark J Nelson and Michael Mateas. Towards automated game design. In *Congress of the Italian Association for Artificial Intelligence*, pages 626–637. Springer, 2007.

[41] Paul Placeway, S Chen, Maxine Eskenazi, Uday Jain, Vipul Parikh, Bhiksha Raj, Mosur Ravishankar, Roni Rosenfeld, Kristie Seymore, M Siegler, et al. The 1996 hub-4 sphinx-3 system. In *Proc. DARPA Speech recognition workshop*, pages 85–89. Citeseer, 1997.

[42] Doru-Vlad Popovici and Cristian Buică-Belciu. Professional challenges in computer-assisted speech therapy. *Procedia-Social and Behavioral Sciences*, 33:518–522, 2012.

[43] F. Provo. Hey you, pikachu review. `http://www.gamespot.com/reviews/hey-you-pikachu-review/1900-2650135/`. Published 11/03/2000. Retrieved 1/7/14.

[44] N Ramou. Automatic detection of articulations disorders from children's speech preliminary study. *Journal of Communications Technology and Electronics*, pages 1274–1279, November 2014.

[45] Joanne E Roberts, Johanna Price, and Cheryl Malkin. Language and communication development in down syndrome. *Mental retardation and developmental disabilities research reviews*, 13(1):26–35, 2007.

[46] S. Ronanki. Mobile pronunciation evaluation for language learning using edit distance scoring with cmu sphinx3, copious speech data collection, and a game-based interface. `http://google-melange.appspot.com/gsoc/project/google/gsoc2012/ronanki/5822463824887808`. Accessed: 2013-05.

[47] T. Lee, S. Ronanki. Pronunciation evaluation for language learning using edit distance scoring with cmu sphinx3, copious speech data collection, and a game-based interface. `http://google-melange.appspot.com/gsoc/project/google/gsoc2012/troylee2008/5205088045891584`. Accessed: 2013-05.

[48] DI Rosenthal, FS Chew, DE Dupuy, SV Kattapuram, WE Palmer, RM Yap, and LA Levine. Computer-based speech recognition as a replacement for medical transcription. *AJR. American journal of roentgenology*, 170(1):23–25, 1998.

[49] Zachary Rubin, Sri Kurniawan, Taylor Gotfrid, and Annie Pugliese. Motivating individuals with spastic cerebral palsy to speak using mobile speech recognition. *SIGACCESS*, 1, 2016.

[50] Zachary Rubin, Sri Kurniawan, and Travis Tollefson. Results from using automatic speech recognition in cleft speech therapy with children. In *International Conference on Computers for Handicapped Persons*, pages 283–286. Springer, 2014.

[51] Alex Rudnicky. Sphix knowledge base tool – version 3. `http://www.speech.cs.cmu.edu/tools/lmtool-new.html`. Retrieved 10/5/16.

[52] K.B. Feragen, A.I. Borge, N. Rumsey. Social experience in 10-year-old children born with a cleft: exploring psychosocial resilience. *The Cleft Palate Craniofacial Journal*, 46:65–74, 2009.

[53] You-sheng Shiu. Speech recognition for aphasia treatment based on openears. 2013.

[54] Annette E Smith and Stephen Camarata. Using teacher-implemented instruction to increase language intelligibility of children with autism. *Journal of Positive Behavior Interventions*, 1(3):141–151, 1999.

[55] Martha E Snell, Nancy Brady, Lee McLean, Billy T Ogletree, Ellin Siegel, Lorraine Sylvester, Beth Mineo, Diane Paul, Mary Ann Romski, and Rose Sevcik. Twenty years of communication intervention research with individuals who have severe intellectual and developmental disabilities. *American journal on intellectual and developmental disabilities*, 115(5):364–380, 2010.

[56] Sandra Sulprizio. *The Source for Cleft Palate and Craniofacial Speech Disorders*. Linguisystems, 2010.

[57] Tiga Talk. Tiga talk. `http://tigatalk.com/`. Accessed: 2016-10-13.

[58] Doonan Speech Therapy. Speech with milo. `http://www.speechwithmilo.com/`. Accessed: 2016-10-13.

[59] Zak Rubin, Sri Kurniawan, Travis Tollefson. Speech adventure: Using speech recognition for cleft speech therapy. *PErvasive Technologies Related to Assistive Environments*, 43(5):598–605, 2013.

[60] Mary Tudor. An experimental study of the effect of evaluative labeling of speech fluency. 1939.

[61] Keith Vertanen. Baseline wsj acoustic models for htk and sphinx: Training recipes and recognition experiments. Technical report, Technical report). Cambridge, United Kingdom: Cavendish Laboratory, 2006.

[62] W. Wahlster. *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, Berlin, 2000.

[63] Halle Winkler. Openears: How do you improve recognition? `http://www.politepix.com/forums/topic/how-do-you-improve-recognition/`. Accessed: 2016-10-13.

[64] Atif Zafar, J Marc Overhage, and Clement J McDonald. Continuous speech recognition for clinicians. *Journal of the American Medical Informatics Association*, 6(3):195–204, 1999.