Speech Adventure

Cleft Therapy via Speech Recognition

Dylan Gardner
UC: Santa Cruz
SURF-IT
Santa Cruz, United States of America

Abstract—**Children with cleft palate or lip have difficulty manipulating airflow through their nose and mouth, leading to unintelligible speech. To aid in the premise of traditional speech therapy, an interactive game has been developed to progress the development of speech- targeted at children ages two to seven. Multiple scenes have been developed to enhance the breadth of speech therapy, as well as the visual experience.**

Keywords—Speech Therapy; Human-Computer Interaction; Speech Adventure

I. INTRODUCTION

Speech Adventure is a game in which promotes the engagement, accessibility, and recovery of post cleft surgery children throughout their speech therapy via human-computer interaction. Currently, the UC: Santa Cruz mascot, Sammy the Slug, is the main character that walks the child through the successive levels. This promotes the interactivity of the game by manipulating Sammy the Slug with each sentence the child is required to say. The efforts set forth for this game is designed to improve recovery time after a child has gone through cleft surgery. We tackled the boredom, redundancy, and other common reports concerning traditional speech therapy by delivering motivation to children during home exercises. The goal of this summer was to enhance the experience of Speech Adventure by developing yet another scene for the enjoyment of diligent children.

II. BACKGROUND

Many children are born with cleft palate or lip. Children with cleft have difficulties manipulating the airflow through their nose compared to others who do not have cleft. Once the child's features are further developed, around the age of two, they undergo cleft surgery. The child has initially learned how to pronounce sounds with their cleft. After surgery, they are expected to relearn how to speak. The unintelligible speech require about five years of speech therapy to correct. Some of the problems with traditional therapy include: engaging with the attention span of a two year old; lessening the confidence of children when asked to perform exercises they cannot successfully do; an untrained ear, the parents/guardians of the child, at home during their exercises produce an inability to finely correct the child's speech as a pathologist would, further regressing their speech.

Ph.D student Zachary Rubin headed the development of Speech Adventure and the research leading up to it. He has developed the Intro, Living Room and Pop A Balloon stages of the game. The premise of his graduate work is to improve the recovery time of children with cleft as well as providing traditional speech therapy an additional tool/aid/substitute. I looked to him for guidance and advice for the layer I was implementing. Our work differed in such that the level I was working on required the use of a physics engine to further enhance the interactivity of Speech Adventure.

III. IMPLEMENTATION

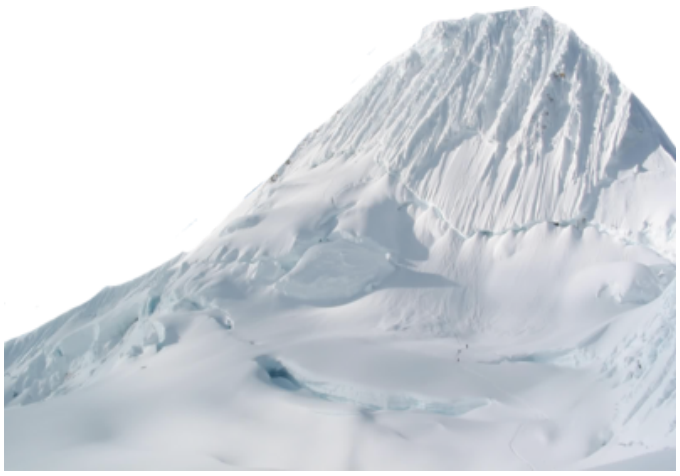
A. Initial Ideas

The initial focus was to investigate child games. We explored free online games on the Web as well as classic Nintendo 64 games in search of motivation for the level I was soon to develop. The simplicity of these games inspired scenes with environments that instantly attract the human eye, rather than focusing on animations that flood the screen. Briefly speaking, we proposed the ideas of a Beach, Dog or Animal Shelter, Safari, Outer Space and the Zoo. With each environment, we expanded on the ideas that could come along with them, respectively. As we waited for the permission to start on the progression of a selected environment, we tempered with the sound files to enhance the narration through each level. From switching narrators to unlinked .wav files in sync with Github, we eventually stabilized the sound files to permit us undivided attention for further implementation.

B. Prototypes

The prototype that was given involved a penguin at the bottom corner of the screen. The general idea was a reaction based scene, depending on the correctness of speech from the child. A timer would arise above the penguin and would count down a specified interval (5-10 seconds). This time would give the child to speak a targeted plosive word or phrase as many times as she/he can. We chose the word 'Go' for simplicity. After the phrase has collected the correct and incorrect data from the analyzed speech, the penguin would then shoot across the screen. This would imply that the more correct the child spoke, the farther the penguin flew. The penguin that we were trying to manipulate was at the discretion of myself, viewed as a placeholder until further artwork was given.

(Taken from the movie Madagascar, used as penguin for prototypical scene)


(The environment layered behind the penguin)

After the scene was in place with the penguin, we soon changed the direction of the prototype. It was difficult to change the main character in the iOS programming from the Slug to the Penguin. In addition, we tried to pursue the most simplistic approach to the new development of the scene.

The next prototype consisted of a scene with Sammy the Slug and a football. With the reaction inspired theme in mind, this scene placed the football over the shoulder of Sammy the Slug. The stopwatch portraying the interval of time given to say the targeted plosive was scratched. Rather, the reaction of the scene would instantaneously manipulate the football as the child would speak. The prototype consisted of an initial pause of the new environment in screen while the narrator revealed the instructions of the layer. Once the narration ended, the football would then project itself across the screen (as if Sammy the Slug threw the football). The difference in this scene was that the football was manipulated by gravity, velocity and speed with correspondence to the correctness of speech. Again, the targeted plosive was "Go" for simplicity. While each word was analyzed, the football would be 'pushed' upwards and outwards, as if it was thrown again, across the screen, further passing the football across the screen. This implied that the more correct the speech, the farther the football flew. However, there were two implications before the prototype could progress.

(Primary particle manipulated)

Speech Adventure used Open-Ears speech recognition throughout its previous levels. This, in turn, affected the way our speech engine analyzed speech. While the sound files narrate the child throughout the game, the speech engine was turned off. Once the .wav files were completed the speech engine clicked back on, awaiting sound waves to be heard. The toggle between on and off with direct dependence of sound files proved to be difficult for the prototype. Zachary Rubin implemented Rapid-Ears speech recognition this summer, alleviating this implication. Rapid-Ears ran as a continuous loop that 'listened' for speech at all allowed times, giving an immense amount of flexibility.

The second problem we ran into was there were little implemented animations throughout this Cocos2D driven app. The current animations we had was the 'walking' of Sammy the Slug from one side of the screen to a designated coordinate on the screen. With directions 'up' and 'right' the floating of the football across the screen could not take advantage of previous animations implemented. With that being said, we needed to integrate Box2D.

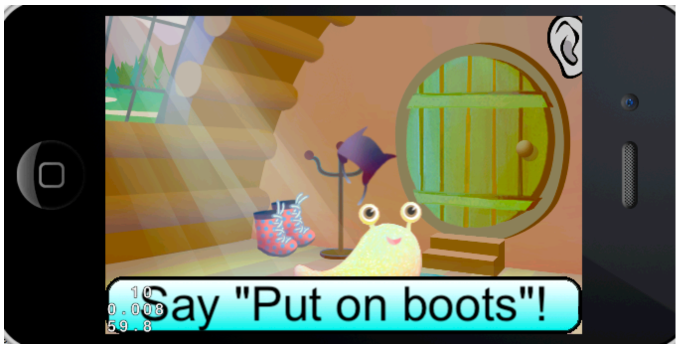
IV. METHODOLOGY

As we started the implementation of the final prototype, we ran into a few problems. First off, we put 'placeholders' in the new level. These consisted of the sound files, backgrounds, Sammy the Slug, the football, language model files, dictionary files, as well as others. The idea behind the placeholders was to arrange the scene and coordinate the objects as needed for the initial set-up of the level, before implementing the physics and animations. The mistake we made was the copy and pasting of previous levels onto the new scene, renaming the files to adjust to the layer. The build was great, and everything was in place. We soon realized that the further we progressed throughout the programming of this level, the more complicated the builds were. The copied and pasted files were still linked to the Github server, and only my version of Xcode knew the differences between the names of the files. This led to the complication of wrong files in wrong levels. Once this was realized, we deleted all of the pasted files in the new prototype and renamed the original files to the correct names. We then recreated the prototype, this time from scratch, and developed files that would not conflict with previous builds.

Box2D, the physics engine dedicated to Cocos2D, was needed to derive the gravitational simulations for the prototype. Although documentation proved the actual programming would be enjoyable to further progress development, the initial implementation of Box2D was more difficult than we expected. In fact, many applications start out as a template from a Box2D project. This means that the entire project starts with the physics engine already implemented. Speech Adventure, however, was not. Integrating Box2D into an existing project provided tribulations that we have not experienced before. The troubleshooting consisted of countless techniques and strategies that led to different outcomes and problems, respectively. A common solution was changing the .m files to .mm files. However, this conflicted with the build of linked files. Exporting the physics engine granted all of the files into the project, but could not extract the commands from them successfully. We also pushed and downloaded Box2D via Unix and tried to combine them. Again, this was unsuccessful when trying to access commands and header files. After all, Box2D was eventually integrated into the existing project in numerous ways. Once we tried to pull methods from Box2D, however, errors ran wild. From switching build settings to extracting the files from a different project, the same project, an adjacent project, there were compile issues if we tried to use any of Box2D's commands and header files.

V. Results

In all, we experienced the necessary obstacles needed to further understand the integration of Box2D into an existing project. Speech Adventure will soon transform itself again into an even more appealing visual tool that will improve recovery time, engagement and accessibility throughout.



(Living Room Scene/Layer)

VI. Conclusion

In conclusion, the development of the level proposed this summer provided a deeper understanding of the software currently being used as well as the engine required for gravitational simulations. The troubleshooting of Box2D stalled the progression of the prototype by hindering the ability to program freely with the physics' commands and header files, respectively. Future work will provide further implementation to enhance the realism concerning the objects promoted in Speech Adventure.

Acknowledgments

I would like to acknowledge Zachary Rubin for providing me a graduate level experience. I would also like to thank Professor Sri Kurniawan for giving me the opportunity to work in her Assistive Technology Lab. Colt Hangen, thank you for giving me insights and guidance throughout the well-experienced SURF-IT program.

References

Zachary Rubin. "Speech Adventure: Using Speech Recognition for Cleft Speech Therapy"

Penguin Picture.

http://www.google.com/imgres?imgurl=http://i199.photobucket.com/albums/aa178/DarkNinja93_2007/Renders/Private-Penguin.png&imgrefurl=http://s199.photobucket.com/user/DarkNinja93_2007/media/Renders/Private-Penguin.png.html&h=924&w=898&sz=529&tbnid=wukOt1gBgdcUiM:&tbnh=86&tbnw=84&zoom=1&usg=__qhXJ7NJIOCQdHPCySzmtakW1H-g=&docid=YzS5z5ZQzsSDbM&sa=X&ei=5XUyUs7bF4rY9AST6YDgDA&ved=0CC0Q9QEwAA&dur=440

Snow Mountain Picture.

http://www.officialpsds.com/Snow-Mountain-PSD83332.html