Language Acquisition Intervention: A Prototype in Supplemental Children's Education Through Media

Kelly A. Vidovic

Follow this and additional works at: https://huskiecommons.lib.niu.edu/studentengagement-honorscapstones

Recommended Citation
https://huskiecommons.lib.niu.edu/studentengagement-honorscapstones/723

This Essay is brought to you for free and open access by the Undergraduate Research & Artistry at Huskie Commons. It has been accepted for inclusion in Honors Capstones by an authorized administrator of Huskie Commons. For more information, please contact jschumacher@niu.edu.
Caitlin Yahnke

Faculty Mentor: Daniel Rogness

Final Reflection on “Language Acquisition Intervention: A Prototype in Supplemental Children’s Education Through Media”

The overall purpose of this honors capstone project was to develop a prototype of a game called Wordbert’s Burrow that would assist young children in the process of learning nouns. Since the first few years of a child’s life are a critical period in learning language, and not all children are exposed to language with the same level of support, this game was intended to help bridge the gap between children in households of different income levels and track the learning progress of the children for possible research use later. As such, a body of research was collated by my capstone partner, psychology student Kelly Vidovic, that we used to together design the structure of the game.

Our initial planning period went well, as we were able to set a number of specific goals for the game’s functionality based on her research, including providing both visual and auditory cues for the nouns the children were to learn, sharing frequent praises both visibly and audibly to encourage engagement, implementing a leveling-up system as children progressed, and providing a dyslexia friendly interface. During the design phase, I developed a basic set of use case models to represent our interaction goals for the game which included the roles of the player, researcher, and database within the system. Creating these use case models, however, also made us realize how helpful it would be to encourage the child’s guardian to participate in the process. For this aim, we decided to keep track of the words the child missed and display them in a list at the end so that their guardian could note them down and practice with them outside of the game environment. We notably made the decision to not display any completion time or error per round information outside of that, as it both does not particularly benefit the young child and could encourage hostility in the child’s environment.

During the actual implementation of the game, most of the process went relatively smoothly. The first order of business was to create the art assets for the game, which consisted of a set of over a hundred pictures drawn by my partner. The artwork was intended to be engaging but not overwhelming, featuring soft lines and vivid colors but not excessive complexity. This design choice was also taken into the appearance of the initial menu interface itself, as I chose to include as few buttons as would allow the game to function. The menu screen itself, coded along with the rest of the game in a C# .NET framework, allows the child to indicate whether they have played the game before, beginning as a new player, resuming their last session, or beginning a new session with their pre-established profile. For the purposes of anonymous tracking, the child is asked to enter a nickname the first time they play which they will be identified by in both the game and the research interface. The game also tracks their age, used to chose which difficulty level they begin at, and their gender. The question of gender is proposed as a question of whether they want to be referred to as “he”, “she”, or “they”, as any question of greater complexity would likely be too confusing to a child that young.
The process of programming the game itself also went well. The game’s basic format is to ask children to match a set of words with a set of pictures, ending the round once all four on screen have been matched correctly. The number of errors and completion time for the round are stored in the database for later display in the research interface. After every correct match, if a sufficient period of time has passed, the mascot of the game, Wordbert, would provide encouragement. Additionally, hovering over one of the displayed words with the mouse plays an audio file of the word being said. This was interesting, and surprisingly simple, to learn to program, as I had never worked with any in-program audio before, especially not with non-default sounds. The audio files also had to be in .wav format, which required the use of special software. I recorded a large bank of audio files of Kelly speaking, one for each word and each piece of praise, which I then cut the samples so they would be clearly audible but not include too much empty time. Further, in the programming of the main game, I needed to account for the fact that a child may not always click things in a logical order. As such, the game knows to discard any attempted matches consisting of two pictures or two words without displaying or checking the match, instead moving the selection box from the first image or word to the second. Finally, at the end of each round, the child is presented with a pop-up box asking them to advance to the next level or quit. If they choose to quit, the list of words they missed is displayed to the screen, with words more frequently missed displayed in a larger font.

The main set of setbacks I faced within the programming of the game came in regards to the database. Since the university’s database is not open to access by outside programs, I had to find a different database to store the game’s player and session data. As we did not have the resources to freely choose any paid database host available, I ended up using AWS’ DynamoDB. This was a challenging prospect for me, as it is a NoSQL database, a format I have never worked with before. This required me to not only throw out my intended database schema and start from scratch late in the process, but also learn to use an entirely new library of commands in order to interface with the database and query it for information. Once I had a handle on this, I was able to update the database as the game was played, storing information from each session to be displayed in the research interface I created. This interface simply allows a user to display and filter players, sessions, and words by certain factors such as age, gender, word level, etc. The user can then select an item to display more information about, querying the database for anything stored on it.

Overall, I learned a lot from this process, especially how to adapt to unexpected hurdles. I practiced my skills at managing user input, handling paint events, working with audio, and using a NoSQL database. All of these skills will be useful in a future career and additionally allowed me to create a game prototype which could be incredibly helpful within the field of children’s language acquisition.