INTRODUCTION

The Problem
Many children are born either wholly or partially deaf, leaving parents to wonder how they can communicate with, and teach their child. For a hearing parent, unfamiliar with American Sign Language (ASL), traditional methods can be exasperating. Additional methods are helpful to reinforce this learning process. The teaching method needs to be effective, yet easy enough for a 2 year old to pick up. Likewise, the premise needs to actually hold the child’s attention.

The Proposal
SmartSignPlay was proposed as an interactive smartphone, or tablet, game designed to make teaching ASL easier and fun. The game was to be point and click, where the child could touch the screen and interact with the in-game characters and objects. Likewise, the art style was chosen with the intention of being fun and enjoyable for children, and parents as well.

Making the Game
Many programs were used in the development process of SmartSignPlay.

Unity
- Unity2D was the main software used, where all the coding, graphics and scene creating was done. All sprites were animated using Unity’s in-built engine and user interface.

MonoDevelop
- Mono was used as the main script editor and debugger for SmartSignPlay. The entire project is visible in its true form here. When Unity is downloaded, Mono always comes bundled with it.

Blender & Gimp
- Blender was used to create the 3D models of in-game family and animate them. Gimp created the graphics, colors and many of the sprites seen in the game.

Game to Date
- SmartSignPlay takes a 45 degree perspective showing the inside of main character, Lily’s house.
- Touch a room and Lily will go to it.
- Certain objects in the room glow. Press these objects to enter a mini-game.
- All glowing objects will cycle through funny animations when you don’t do anything. Some shenanigans and mild cartoony slap-stick will keep the kids laughing!

DISCUSSION AND CONCLUSIONS

After more than a year of development, SmartSignPlay is nearing its early pre-alpha release.
- Researchers are using the app to determine what age-group will most benefit from playing the game, and the impact it will have on children learning ASL.
- Sample age-groups are being rounded up by UNF researchers to play and test the game.
- SmartSignPlay will continue to be developed into the foreseeable future with the goal of improving functionality and the overall player experience.

Future Plans
- Add a brother for Lily as a playable character.
- Implement a randomness system where objects appear on screen depending on player input over time, so no game sessions are the same.
- Make all objects in a room interactable.

The Home Scene
- Each mini-game will have a sample video showing how a hand-sign is done.
- The player must then choose the correct hand shape.
- The player is rewarded with in-game stickers if they do well.
- They will then be given the opportunity to sign the shape real-time by using their camera.
- The player can try again as many times as they like.

The Mini-Game Scene
- Certain objects in the room glow. Press these objects to enter a mini-game.
- All glowing objects will cycle through funny animations when you don’t do anything. Some shenanigans and mild cartoony slap-stick will keep the kids laughing!

The Making of the Game
- Unity2D was the main software used, where all the coding, graphics and scene creating was done. All sprites were animated using Unity’s in-built engine and user interface.
- Mono was used as the main script editor and debugger for SmartSignPlay. The entire project is visible in its true form here. When Unity is downloaded, Mono always comes bundled with it.
- Blender was used to create the 3D models of in-game family and animate them. Gimp created the graphics, colors and many of the sprites seen in the game.

Development Environments
- Unity2D was the main software used, where all the coding, graphics and scene creating was done. All sprites were animated using Unity’s in-built engine and user interface.
- Mono was used as the main script editor and debugger for SmartSignPlay. The entire project is visible in its true form here. When Unity is downloaded, Mono always comes bundled with it.
- Blender was used to create the 3D models of in-game family and animate them. Gimp created the graphics, colors and many of the sprites seen in the game.