Sample Interface Heuristics Evaluation Form

Heuristics Evaluation of

By **Date** 10/24/2018

Heuristic Evaluation of Prototype Interface

Heuristic	***Evaluation*** In the space below, enter your observation and evaluation of the degree to which the heuristic has been satisfied. Use as much space as you see fit.
1. Visibility of system status • Always keep users informed about what is going on. • Provide appropriate feedback within reasonable time.	Very few screens have a “title” block to them causing issues with user direction such as the problem encountered with suddenly selecting an avatar when expecting to select a game when going through free play route. Generally provides either screen transition or toast on actions to inform users.
2. Match between system and the real world • Speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. • Follow real-world conventions, making information appear in a natural and logical order.	The app appears to be fine due to not much system-related stuff presented, the closest being a code that must be entered to connected students with a class, but that would generally be easily understandable for the user group; however depending on how young the children get, codes may be hard to understand.
3. User control and freedom • Users often choose system functions by mistake. • Provide a clearly marked "out" to leave an unwanted state without having to go through an extended dialogue. • Support undo and redo.	Back button on almost every screen, all screens have it if you include the android built-in. One of the screens where a back-button or repeat button is not is the fill in the blank sentence activity. This is a major issue as the user should be provided an option to look back at the sentence or replay the audio. Generally, none of the actions need or could even viably support an undo/redo. This was not shown in the user test, but there needs to be a way to remove oneself or delete classes.
4. Consistency and standards	Standards Generally Intact.

• Users should not have to wonder whether different words, situations, or actions mean the same thing. • Follow platform conventions.	There is a fair degree of similarity between various screens in the application; however, the action when selecting the similar elements of these screens, images for example, produce different results on occasion such as the production or lack thereof of audio. The “rhyme” games were identified by images of different animals, and these animals seemed to correspond to animals whose names the student spelled out previously in the “spell” games. If this is the case, then this connection needs to be clarified more.
5. Error prevention • Even better than good error messages is a careful design which prevents a problem from occurring in the first place.	Not too much room or places at all to get errors. The only places that where this could be improved would be explicitly stating what are the requirements and constraints for entering various fields on user registration and only allowing numerical input for when a code is entered. Another way to prevent some common registration errors would be to provide a way for teachers to register students , rather than having students enter a classroom code. This could be done by having a mechanism by which the teacher transmits a code through bluetooth that the student’s phone automatically reacts to.
6. Recognition rather than recall • Make objects, actions, and options visible. • User should not have to remember information from one part of the dialogue to another. • Instructions for use of the system should be visible or easily retrievable whenever appropriate.	Everything that is visible is connected to button or picture. There is a lot of actions that are heavily connected to audio cues, but are not always shown in a visible format.. In certain places the end of an audio line immediately transitions with no repeat or back button which is a serious issue.
7. Flexibility and efficiency of use • Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user so that the system can cater to both inexperienced and experienced users. • Allow users to tailor frequent actions.	Not much room to add in shortcuts.However, there may be some room for teacher’s to be able to give assignments for kids to do which would automatically direct them to the appropriate screens, rather than having to navigate through many. There may also be room for users to ‘favorite’ or ‘come back to later’ certain assignments if they wish to repeat them or cannot finish them at the current moment. Many of the pages did not contain a home button to redirect the student back to the home page. This would be a good addition as a shortcut as you end up having to click back many times to get back to home from many screens. For the teacher’s view, adding new classrooms and adding

	new students seemed needlessly tedious as you had to go through the "view classrooms" and "view students" pages, respectively. A shortcut for these tasks can increase efficiency.
8. Aesthetic and minimalist design • Dialogues should not contain information which is irrelevant or rarely needed. • Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.	It does not appear to be an app where having an avatar would provide any benefit or would be in a situation to be displayed to other classmates. Frankly, it felt forced and sudden to select one as well. A lot of thought is put into audio cues but there isn't an option to turn it off and there are a few places such as the sentence game where audio supersedes text or user input(automatically goes to next window) which is a issue. In short, make sure audio supplements text and the user, don't make audio the focus point and determine user actions. Single function screens with obvious controls are definitely a good idea.
9. Help users recognize, diagnose, and recover from errors • Expressed in plain language (no codes) • Precisely indicate the problem • Constructively suggest a solution.	During the test, there was a lack of demonstration of error scenarios. Assuming the standard error messages appear for invalid input of fields or code not found, the app should be fine in this regard, however none of this was demonstrated. There should also be error messages for if the user cannot connect to the internet since due to the class system, it would appear to be heavily reliant on feeding information of completed assignments over it. It is highly recommended to include whose side the errors are on in this regard -- phone not connected to internet, internal server error, etc.
10. Help and documentation • Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. • Help information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.	There is a lack of clear direction for many screens or routes throughout the app. Perhaps add this as an info button or just as instructions or within a title block where appropriate. It is very important for audio instructions to be present on every single page; while the researchers did mention that audio instructions would exist on every page, this was not a part of the demo. One issue that occured during one the tasks was not knowing that a screen was to select an avatar as opposed to choosing specific assignment, the latter being what the user testee expected. It also had to be verbally explained by the tester that sound

	cues occur on certain actions, which actions are connected to sound cues should be explicitly mentioned at least for first time users.

Miscellaneous Comments:

The codes for the classes are only 4 digits, so at most 10,000 codes. These codes need to be made so they are unique as one could fairly reasonably get the same code out of 10,000. There also needs to be some functionality setup for expandability of the code system because if there was either inactive classes(filling up slots of the 10,000) or if the app just became popular enough, this 10,000 limit could be easily reached. It also seemed unclear whether classroom codes were unique to each class or to each student -- if the latter is true, then the above concern is amplified. An alternative could be to use a QR code or some sort of a bluetooth mechanism for automatically registering students.

The user test generally was very linear and didn't appear to leave much room for us to experience an error.

There should also probably be a way for a teacher to add students as the adult should be more adept at making sure students get in the appropriate class.