Words With Kinect

Final Document

Team Chimera

Taylor Kohn, Mario Leuang, Kevin Fox, Rui Li

Advisor

Dr. Diane Rover

Client

Dr. Donald Bear

Words With Kinect

Project Design

System Design

Introduction

Purpose

This design document will include specific details about the technology and hardware used in the creation of the "Words with Kinect" project. We will also use this document to further specify detail in regards to screen designs for the games and menus.

Scope

This document will cover the low level details on the system design of our project. This includes programming languages, API, and hardware specifications. It will also cover higher level concepts such as screen flow and games. Screenshots of the software are located towards the bottom of this document.

Requirements

Game Requirements

The following are requirements from our client, Dr. Donald Bear:

1. A portion of the games must have a timed component
2. Scoring is based on accuracy
3. All games must be designed to categorize words (e.g. Long A, Short A, or neither)

Functional System Requirements

The following are our functional requirements:

1. Local database of words
2. Randomization of words
3. Accurate hand/gesture recognition
4. Hand draggable controls

Non-Functional System Requirements

The following are our non-functional requirements:

1. Visually appealing to children. Colorful, but unobtrusive to the objective of helping children learn
2. Intuitive for children; children should be able to clearly know the goals of lessons or games and be able to complete them with simple controls
3. All images and content will be age-appropriate for a childrens' game

Software Overview

Games

Words with Kinect will have many different games to help children learn English. These games are based off of the research of Dr. Bear. The games described below are designed to help children understand the structures of word parts by distinguishing them into categories and visualizing what category of word they need to be sorted in. There are currently three games in Words with Kinect: Matching, Word Sort, and Memory.

a. Matching (*Figure 1*)

In this game the user will be faced with 2 columns of pictures. The objective of the game is for the user to select pictures from either column and match them with a picture from the opposite column with the same category of word. There are 3 categories of words in this game: Long Vowel, Short Vowel, and Oddball. Per Dr. Bear's requirements, there will always be an "Oddball" category that is neither Long Vowel or Short Vowel. Correct matches receive 10 points added to the score, incorrect matches deduct 4 points from the score. There is an option to time the game.

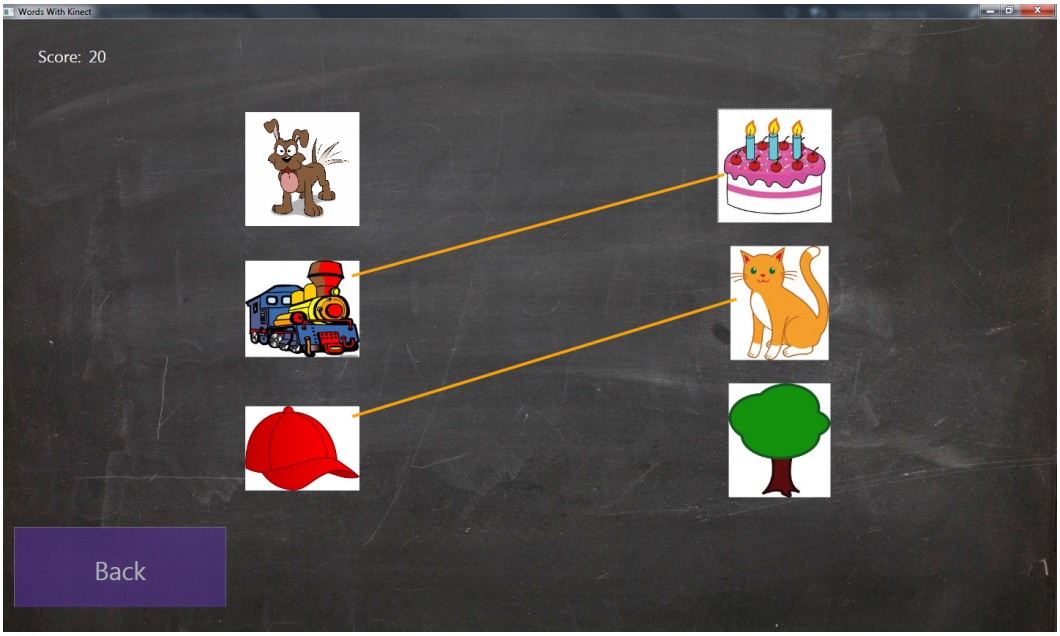


Figure 1 Matching Game

b. Word Sort (*Figure 2*)

In this game the screen is divided into 3 columns based on the category of the words (Long Vowel, Short Vowel, Oddball). The user must grab and drag the words from the word pool into the correct bin. The game is completed when all words are sorted correctly. There is an option to time the game.

Figure 2 Word Sort Game

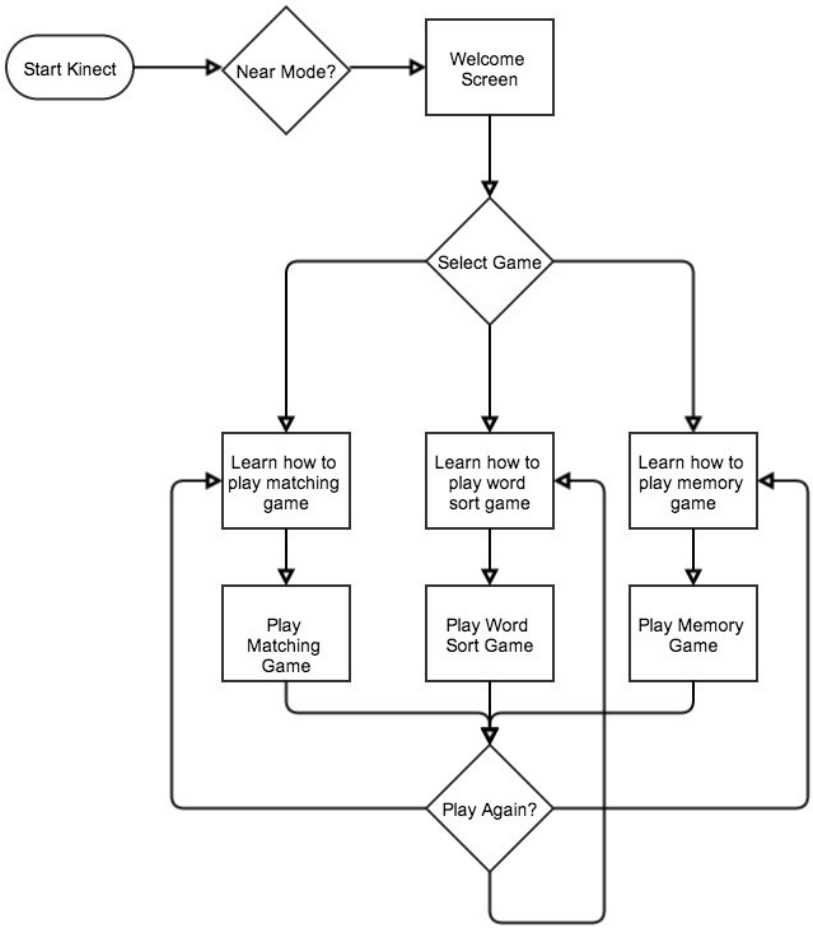
c. Memory (*Figure 3*)

In this game there will be be 12 facedown cards (4 columns, 3 rows) containing words. The user must pick 2 cards to match together. If the cards match, they disappear and 10 points are added to the score. If they do not match, they flip back to their original state and 4 points are deducted from the score. There is an option to time the game.

Figure 3 Memory Game

Functional Decomposition

This section will describe the process of a typical user who interfaces with our system. Below is the sequence of events that a typical user will experience:



Functional decomposition

Detailed Design

Input/output, Hardware specification

Input/Output specification and Hardware specification can be found from Microsoft for their Kinect. These specifications can be found here:
http://msdn.microsoft.com/en-us/library/jj131033.aspx

Hardware specification for the Kinect SDK is as follows:

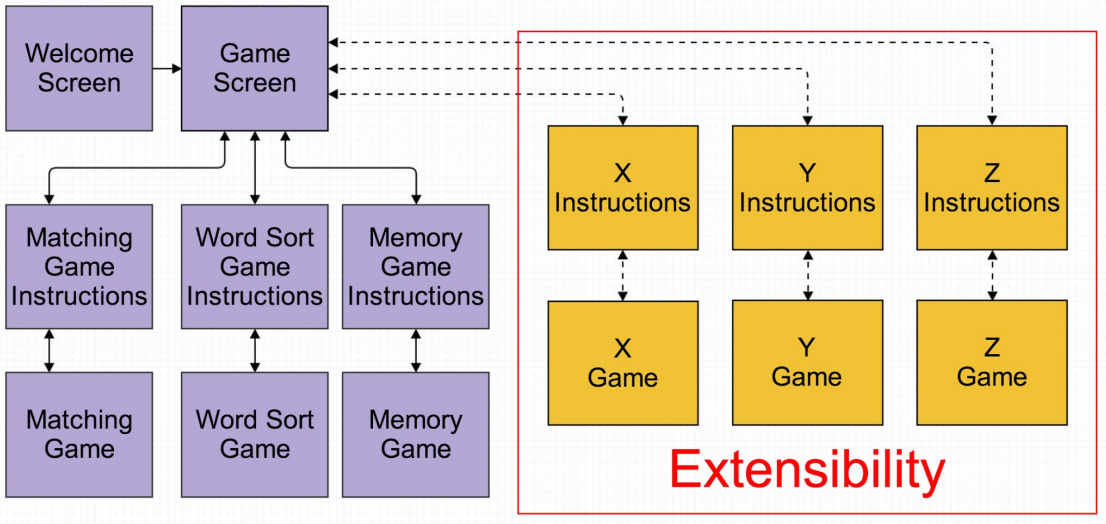
- 32-bit (x86) or 64-bit (x64) processor

- Dual-core 2.66-GHz or faster processor
- Dedicated USB 2.0 bus
- 2 GB RAM
- Microsoft Kinect for Windows Sensor

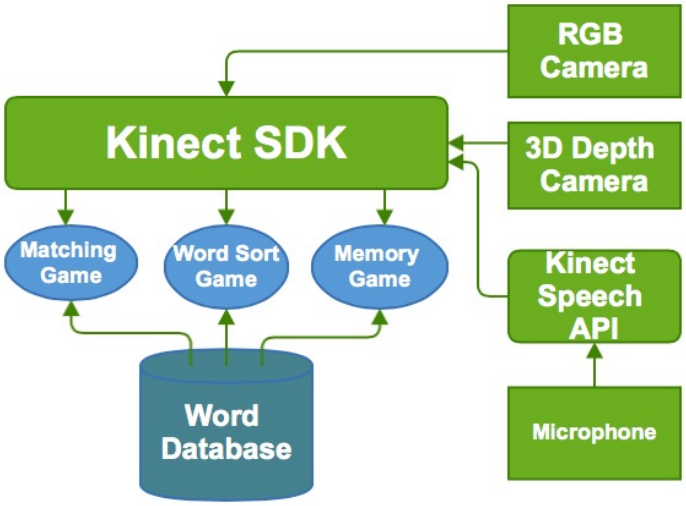
Hardware and Input/Output specification for the Microsoft Kinect itself is as follows:

- Communicates serially via USB
- The standard frame rate of 30 fps cannot be exceeded
- A display with a resolution of at least 640x480

Screen Flow



Architecture



Software specification

Use-Cases

Interacting with system

1. Users can select buttons by hovering over them with the virtual Kinect hand and pushing forward with their physical hand
2. Users can select buttons using a mouse to click them
3. Users can grab and drag items by hovering over them with the virtual Kinect hand and clasping their physical hand

Non-Functional Requirements

1. Extensibility; game modules will be designed with further extensibility in mind. Developers will be able to create their own games or use different categories of words for current games
2. Maintainability; software will be organized modularly and code will be uniform in order to maximize maintainability
3. Accessibility; buttons are designed to be large enough to be easily pressed with a hand gesture on a Kinect
4. Quality; Words with Kinect will accurately measure the child's performance
5. Accuracy; gesture and voice recognition will be accurate

Test specification

Prototyping

Small sample games will be designed based on the information we learn from Dr. Bear. The game is then taken to Dr. Bear where we will decide if they are teaching the children properly.

Testing

See attached testing plan below.

Works Cited (pictures for the Matching Game)

Greenwich Public Schools, “Tree”, https://lh4.ggpht.com/w8h4zH11acGsdzXl3BFkpIeMbM5cHyWD5cDooKcJ9x413uvtbhRoj6FD0N3KpEZ78nzaFYY=s85, 5/27/2014

jicheng, “Train”, http://www.jicheng168.com/subimages/hcys2.jpg

colourbox, “Cake”, http://images2.colourbox.com/thumb_COLOURBOX5677084.jpg

drawing now, “How to draw a cat”, http://www.drawingnow.com/file/videos/image/how-to-draw-a-simple-cat-1.jpg

dglassme, “mastectomy", https://dglassme.files.wordpress.com/2014/03/happy_cartoon_dog.jpg

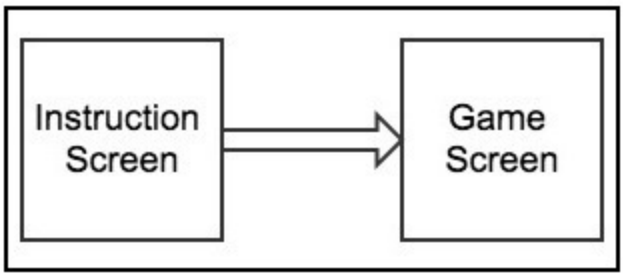
tomei, “Baseball Cap”, http://media.wths.net/zaban/Tomei/images/baseball_cap.jpg

Words With Kinect

Testing Plan

Abstract

This document outlines a testing procedure for adding a new game to the Words With Kinect application. The new game will henceforth be referred to as a game module or module. It will consist of both an instruction screen and a game screen as seen below.



References

The following are references that used throughout this document.

Reference	Document
Document A	“Creation Of New Modules”
Document B	“Details of Games”
Document C	“Screen Images”

Smoke Tests

The following smoke tests are in place to ensure the basic functionality of the module is correct and to ensure that it adheres to the specification outlined in Document A. These tests must be passed before a module can be accepted into the application

Instruction Screen

Title Text

1. Title of the screen is spelled correctly
2. Font size of the title is 85 point with Comic Sans font

Instruction Text

1. Instructions are spelled correctly and are free of grammatical errors
2. Font size is set to 48 point with Segoe UI font

Start Button

1. Start button is the appropriate size as described in Document A.

2. “Clicking” the button with either the mouse or through the kinect will bring the user to the *Game Screen*.

Back Button

1. Back button is the appropriate size as described in Document A.
2. “Clicking” the button with either the mouse or through the kinect will bring the user to the *Game Screen*. (See Document A for Game screen).

Timer

1. Clicking timer will prompt the user for a number of seconds

Game Screen

Back Button

1. Back button is the appropriate size as described in Document A
2. Clicking the back button with either the mouse or kinect will bring the user back to the *Instruction Screen* that corresponds with the game being played.

Game Logic

1. The logic of the game works as described in Document B.

Regression Test Procedure

Regression will be run every time a new module is introduced into the project. This will ensure that the game integration was successful and no bugs were introduced into the project. These tests will only test the functionality and not the content (To check content, compare screen to Document C). Completion of regression testing implies successful integration.

Welcome Screen

Start Button

1. Start button should advance the user to the game screen.

Game Screen

Matching Button

1. Click Matching button
2. Screen advances to Matching Instruction Screen

Word Sort Button

1. Click Word Sort button
2. Screen advances to Word Sort Instruction Screen

Memory Button

1. Click Memory button
2. Screen advances to Memory Instruction Screen

Game Instruction Screens

To test these screens follow the procedure outlined in Smoke Test above.

Updates

Updates will be done to this document when new modules are successfully added.

1. Update regression tests
    a. Add test to test functionality of button that is placed on Game Screen
2. Update document version number

Words With Kinect

Test Results

Abstract

This document details the result of the testing that we performed while working with Dr. Bear's students.

Introduction

On April 10th we had the opportunity to interact with children that Dr. Donald Bear is in charge of teaching. We were able to observe them interacting with our application and below we have recorded some of our observations.

Observations

What we Learned

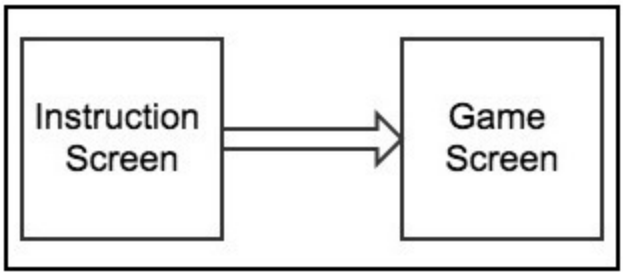
- The students that had never used a Kinect before had a hard time with the grabbing and clicking technique. That being said they improved dramatically as they continued to use the kinect
    - To remedy this we should add a tutorial on how to perform different gestures. Due to time restrictions we were unable to add this feature to the project.
- The teachers that we had a chance to talk to all seemed to be impressed by the application. It mimicked accurately what they were teaching the students on paper.
- The students themselves seemed to find the application engaging and were excited to use it. They gathered around and watched their peers play and were excited to play.

Words With Kinect

Module Creation

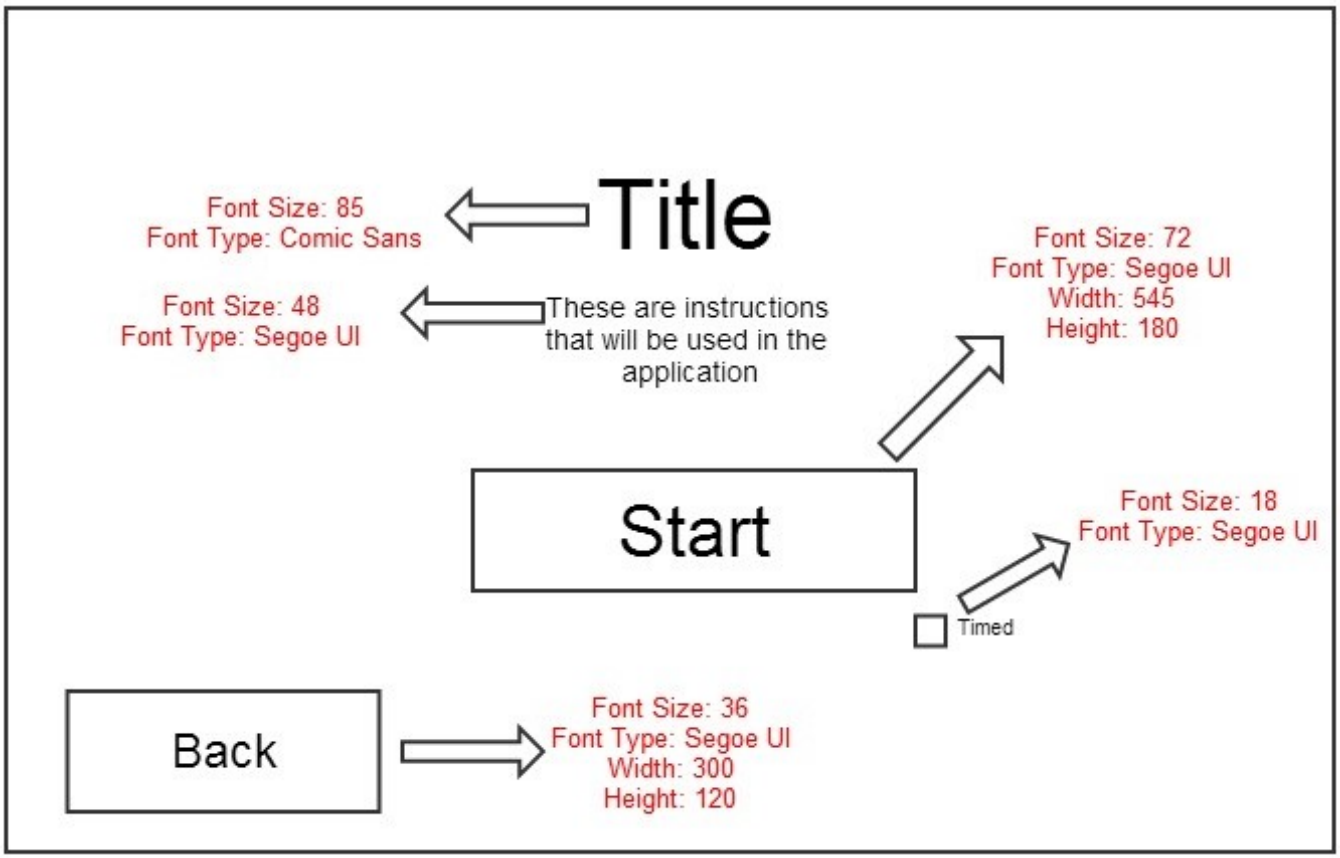
Abstract

This document outlines the procedure to create a new module that will integrate into the Word With Kinect application. A module consists of an instruction screen and game screen that have the ability to navigate to each other. See the picture below for more detail.



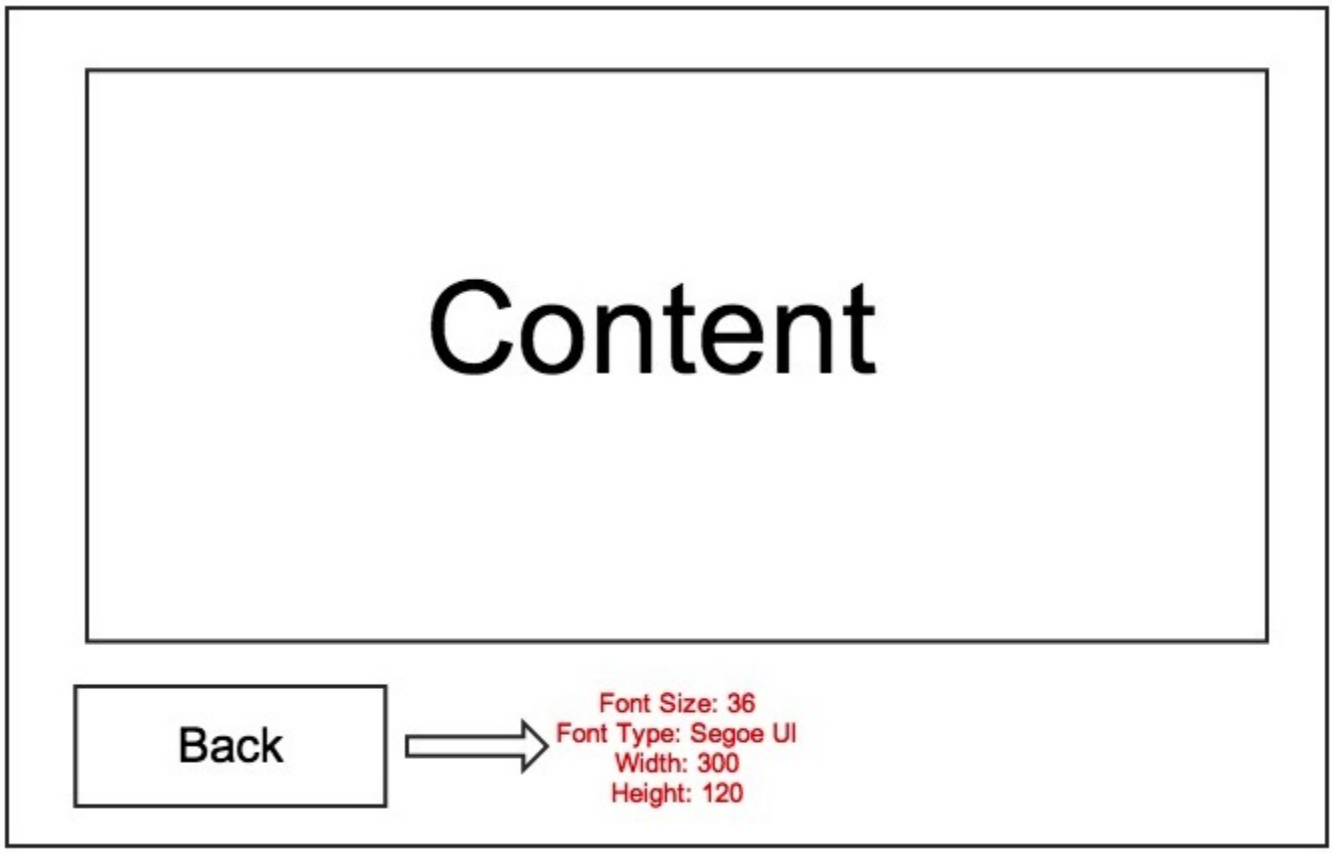
Creation Of Instruction Screen

The Instruction Screen introduces the user to the module and instructs the user on information they will need to interact with the game screen.



Creation Of Game Screen

The Game Screen is what appears after the start button is clicked on the Instruction screen. It contains the game content/logic and a back button.


Content
Back
Font Size: 36
Font Type: Segoe UI
Width: 300
Height: 120

Appendix I

Words With Kinect

Instruction Manual

Introduction

Overview

Words with Kinect is an application that is intended to teach children english phonetics. The children interact with the application by playing games via the Microsoft Kinect. These games are based on the research of Dr. Donald Bear.

Games

There are currently three games that are implemented in the application. These games teach the children the phonetic sounds of “long-A” and “short-A”. The three games that are implemented are: Matching, Word Sort, and Memory.

Specifications

Hardware

- Microsoft developer Kinect or Microsoft Xbox 360 Kinect
    - For best results we recommend the developer Kinect
- 32 bit or 64 bit processor
- Dual-core 2.66 GHz or faster processor
- Dedicated USB 2.0 bus
- 2 GB RAM[1]

Software

- Windows 7 or Windows 8

Users

Ages 6-8

This application is intended to used by children ages 6-8. At this point in their academic development the children should be learning phonetic sounds of words.

Ages 8+

This application can also be extended to help teach children and adults who are learning english as a second language how to read and write. The ability to plug-in modules makes the application very flexible.

[1] These requirements are based on what is needed to run the kinect. For more information visit: http://www.microsoft.com/en-us/kinectforwindows/purchase/sensor_setup.aspx

Input/Controls

Kinect

The main source of input for this application is the Microsoft Kinect. With the Kinect the user is able to perform a variety of gestures to interface with the application. The Kinect tracks the movement of the users hand and depicts this hand as an animated hand in the application

Gestures

Clicking

The user can perform a click with the Kinect by hovering their hand over an object and then slowly moving their hand forward. This will cause the Kinect hand to slowly fill purple. When the hand is all the way fill this denotes a click operation. This click is the same click that can be performed by a mouse.

Grabbing

The user can perform a grab gesture by moving their hand so that it is over an object. Once over an object the user can perform a grab by making a fist with their hand. The animated hand will also make a fist if the grab is successful. The user can release the grab by unclenching their fist. The animated hand will return to normal when this is successful.

Dragging

The user can perform a drag operation by grabbing a control and then moving their hand around the screen. The object will continue to follow the users hand until the user releases the grab.

Mouse

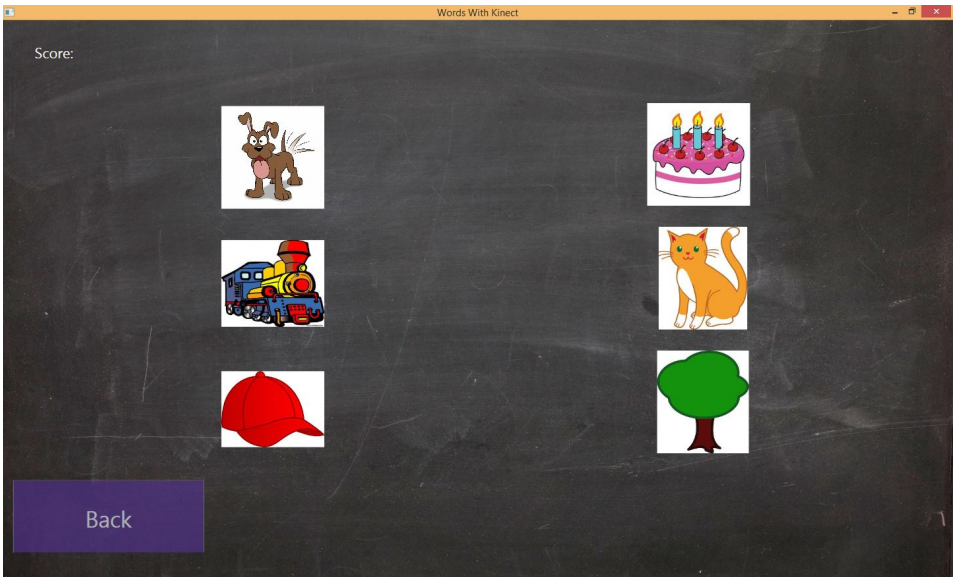
The user may also use the mouse to interact with the application. Note, while many of the features will still work with the mouse there are features that are Kinect exclusive including the grabbing and dragging features. For this reason it is recommended that the Kinect be used.

Games

Matching Game

Introduction

The Matching game consists of two columns that have pictures in them (see screen shot below). The object of this game is to draw a line between pictures that have the same phonetic sound. There is a match for “long-A”, “short-A”, and oddball.



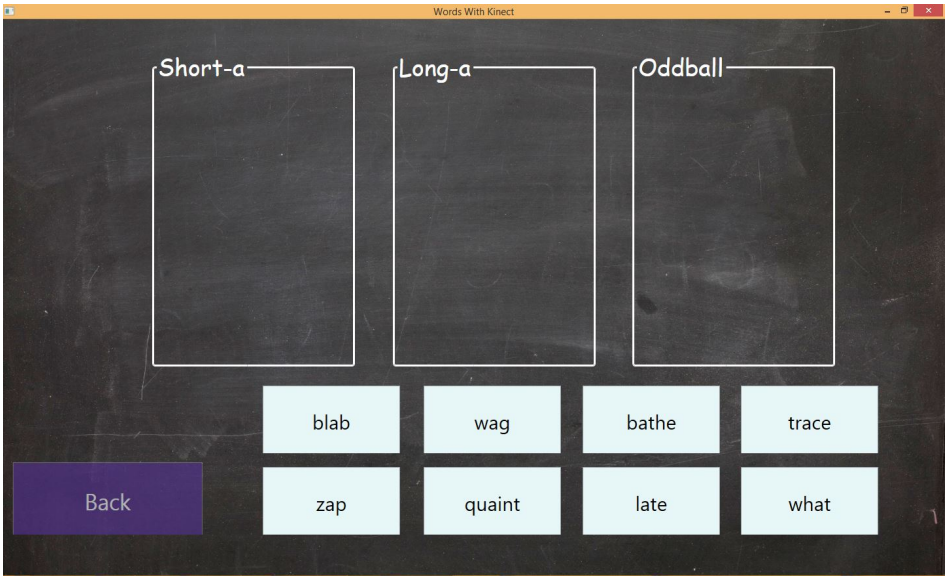
GamePlay

The user starts by selecting a picture in the left hand column. This is done by “clicking” on a picture. The user then clicks on a picture in the left column. If the pictures have similar phonetic sounds then the user gets points and a line is drawn between the two pictures. If a user is incorrect, points are taken from their score and no line is drawn.

Word Sort Game

Introduction

The Word Sort game has three columns that represent three different phonetic sounds. The user also has a bank of words at the bottom of the screen. The job of the user is to drag a word into the correct column to earn points.



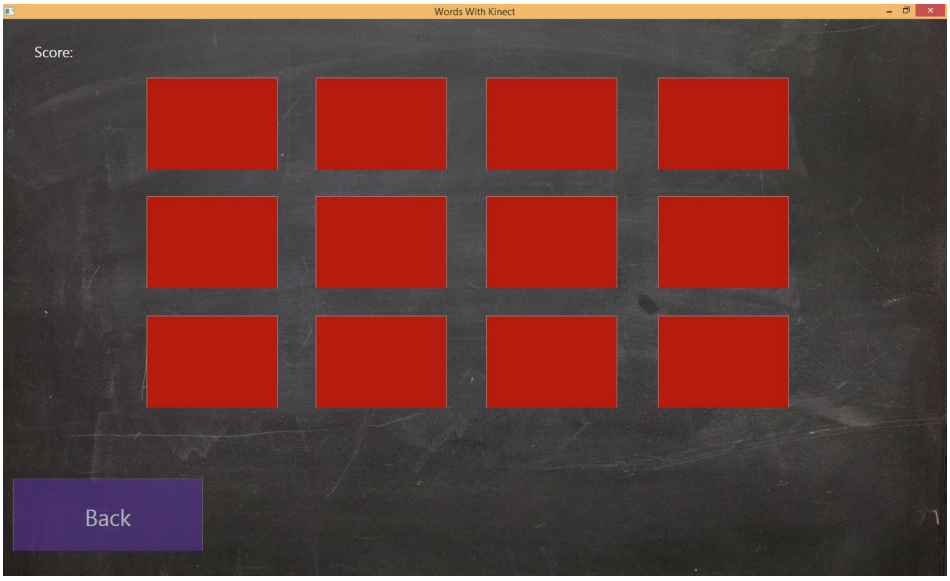
Game Play

The user will select a tile on the bottom of the screen by using the grab technique. The user will then drag the word to the appropriate column and release the grab. Once in the correct column the user will click on the tile to verify that it is in the correct position. If the user was correct then the column will light up green, the tile will disappear and the user will get points. If the user was incorrect then the column will light up red and the tile will stay in place and wait to be moved to the correct position.

Memory Game

Introduction

This game is laid out in the familiar memory fashion. The object of the game is for the user to find two cards that have the same phonetic sound.

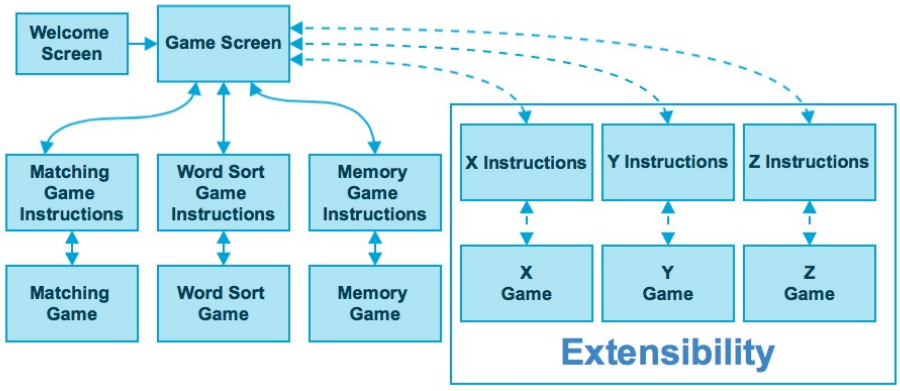


Gameplay

The user will flip a card over by clicking on one of the red tiles. The user will then flip over another card by clicking on a separate tile. If the two cards that are flipped over have the same phonetic sound then the user will get points and the cards will stay blue. If the user is incorrect the cards will remain face up for a short amount of time and then flip back over.

Screenflow

Below is the screen flow diagram that shows how to navigate the application.



Contact

Please direct any questions or comments to: chimera.isu@gmail.com

Appendix II

Words With Kinect

Design Document

System Design

Introduction

Purpose

This design document will include specific details about the technology and hardware used in the creation of the "Words with Kinect" project. We will also use this document in order to go into more detail with screen designs for the games and menus. For more information, including a timeline of the project please refer to the Project Plan.

Scope

This document will cover the low level details of our project. Including programming languages, API, and hardware specifications. It will also cover more higher level concepts such as screen flow and games. Located at the bottom of the document you will find sample screen sketches of what the final games will look like.

Requirements

Game

The following requirements are what Dr. Bear requires to have in our games:

1. A portion of the games have to have a timed component
2. There will be an assessment before the student begins.
    a. This assessment will be a spelling test to determine the appropriate level for the student to start at
3. Score is based on speed
4. Score is based on accuracy

Other

The following requirements are other requirements that improve the overall game play and organization:

1. Each user will have a seperate profile
2. Progress of the users will be kept and stored locally

Functional Requirements

The following are our functional requirements:

1. Each lesson or game has a tutorial of some sort to the side that shows how to complete each objective
2. A persistent menu button and settings button
3. Words can be grabbed and dragged with the Kinect SDK grab gesture in the word sort

game

4. Home screen with options for Lessons, Games, Placement Test, Progress, and Settings

Non-Functional Requirements

The following are our non-functional requirements:

1. Visually appealing to children. Colorful, but unobtrusive to the objective of helping children learn
2. Similar sized buttons and colors throughout the application
3. Simple and easy enough for children to use. They should be able to clearly know the goals of lessons or games and be able to complete them with intuitive controls
4. All images and content will be age-appropriate for a childrens game

Software Overview

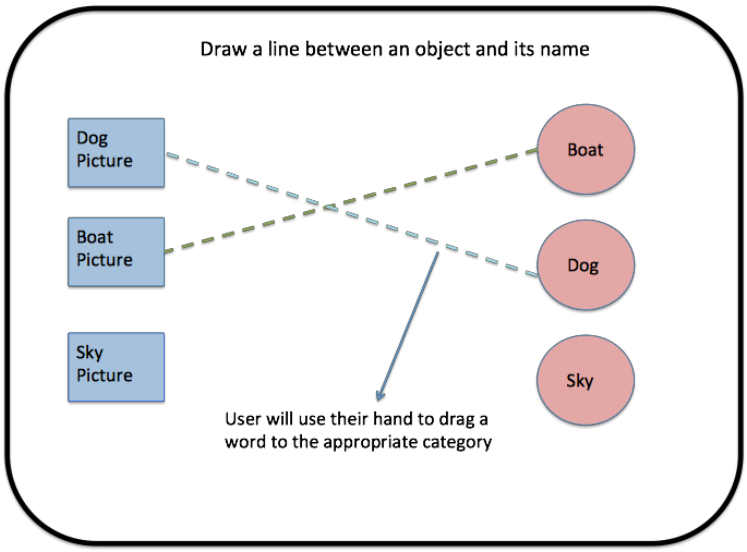
This section will contain information about the games and the menu.

1. Games

Words with Kinect will have many different games that the user can unlock as they progress through various lessons. These lessons are based off of the research of Dr. Donald Bear. The games described below are designed to help children understand how the words are sorted and memorize the vocabulary.

a. Matching

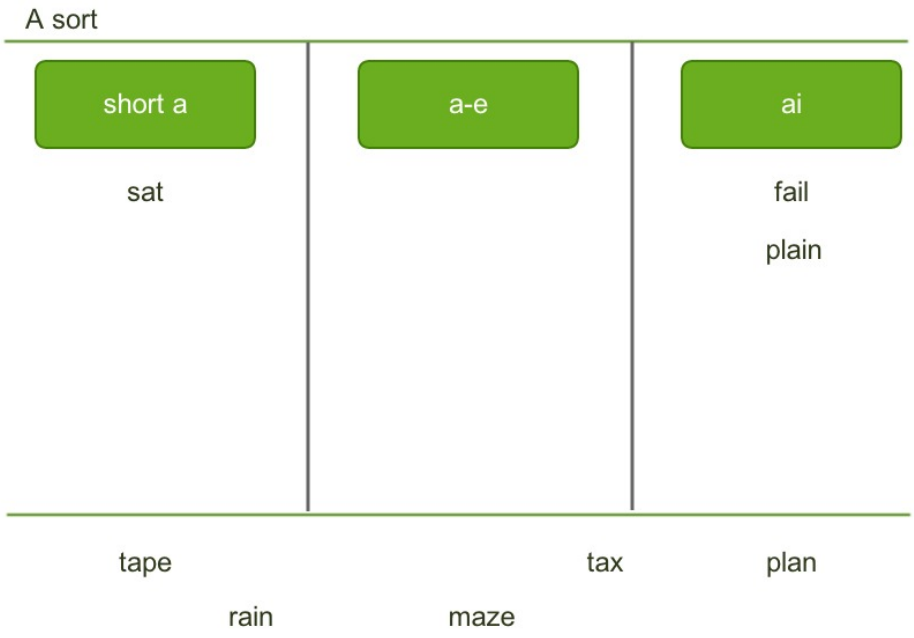
In this game the user will have a list of words and a list of pictures. See figure 1 below of the basic setup of the game. The object of the game is for the user to drag a given word over a picture. The difficulty of the words will be determined based on the users score from the assessment. This exercise will also be timed per Dr. Bears requirements.



Matching game

b. Word Sort

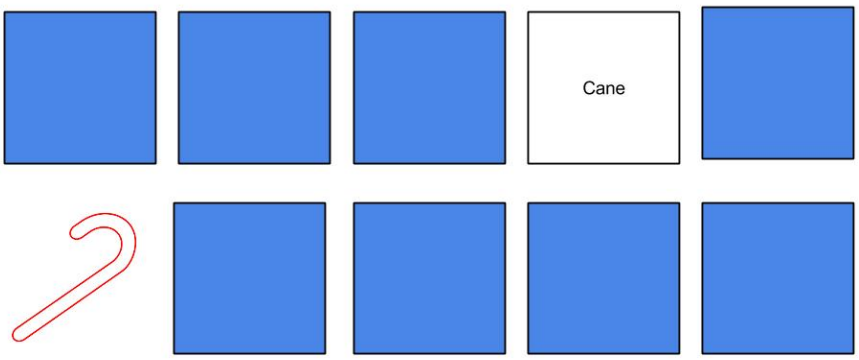
In this game the screen is divided into three columns based on the sound of the words. For example you are given a list of words and there are three columns: short a, a-e, and ai. The player must grab and drag the words from the word pool into the correct bin. This game is timed per Dr. Bear's recommendation.



Word Sort game

c. Memory

This will be a game where there will be 10 facedown cards (5 words in one row, 5 pictures in a row below it. The child can pick 2 cards to match together. If the word card matches the picture card, the cards disappear.



Memory game

2. Menus

On the main screen, there will be options to select lessons, games, the assessment test, and the application settings.

mockup of the menu screen

a. Lessons

The assessment will place you in the correct skill level of lesson. After the assessment, the user will then progress through lessons in order of difficulty.

mockup of lessons screen

Here, the child will be able to practice speaking different phonetic sounds. For instance, in the picture above, the child will be able to read different words with the Ch- sound.

b. Games

Games are based on the skill level that the user is currently on. As mentioned earlier, here are three games to choose from.

c. Assessment

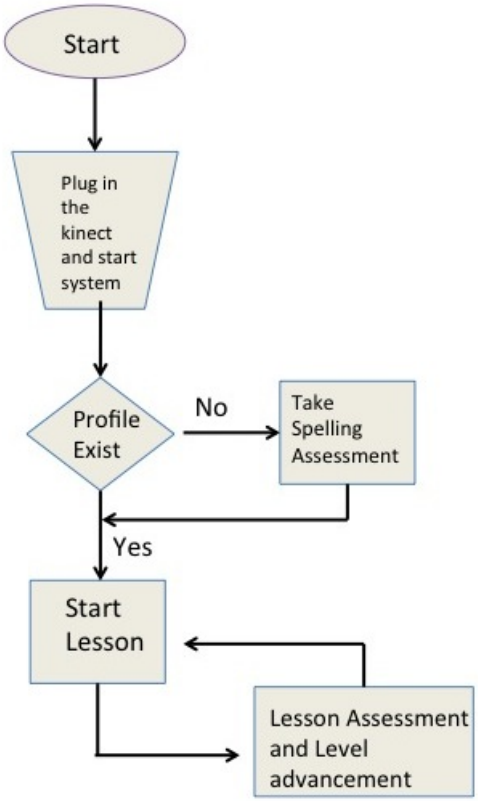
The assessment will evaluate the user's skill level. It will prompt the user to spell the word spoken to them. Games and lessons will be based off of the skill level that is determined by assessment.

d. Settings

Here we will have volume adjustments for different audio of the game like the sound effects, narrator voice, and music. There will also be profile creation and deletion controls.

Functional Decomposition

This section will describe the process of a typical user who interfaces with our system. The user will also have the options to go straight to games, however this is not typical. Below is the sequence of events that a typical user will experience:



Functional decomposition

Detailed Design

Input/output, Hardware specification

Input/Output specification and Hardware specification can be found from Microsoft for their Kinect. These specifications can be found here:
http://msdn.microsoft.com/en-us/library/jj131033.aspx

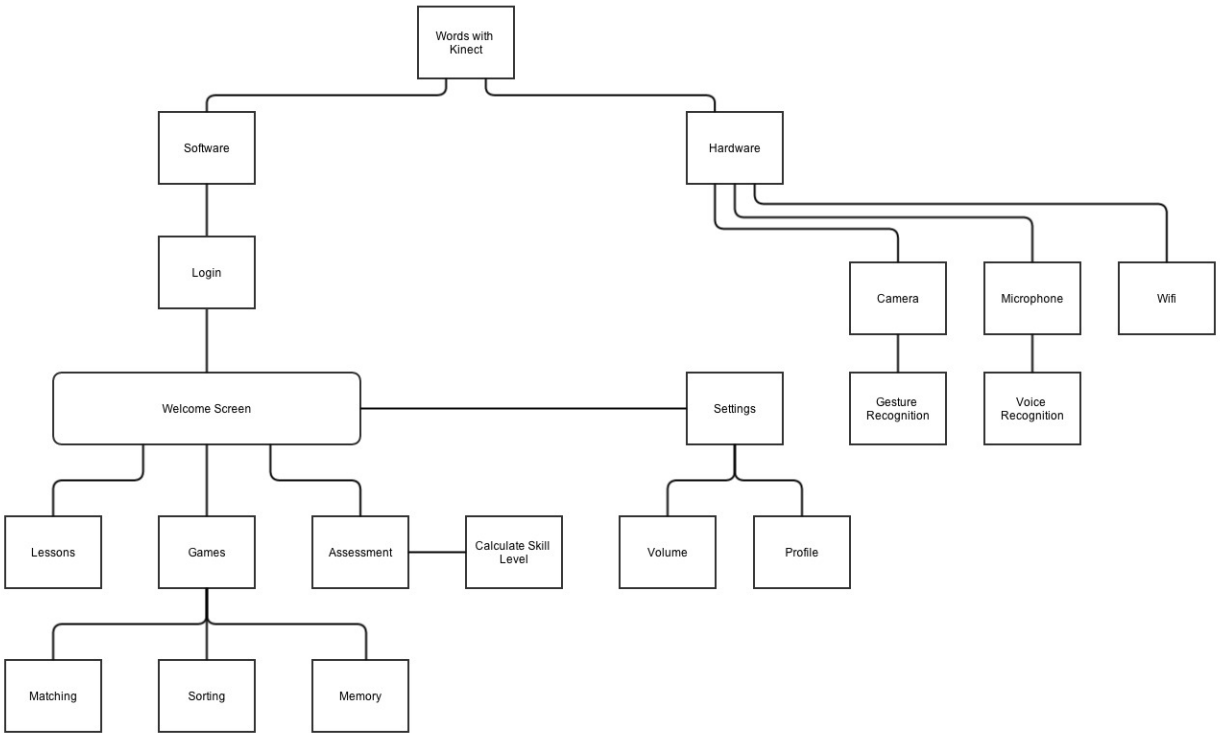
Hardware specification for the Kinect SDK is as follows:

- 32-bit (x86) or 64-bit (x64) processor
- Dual-core 2.66-GHz or faster processor
- Dedicated USB 2.0 bus
- 2 GB RAM
- Microsoft Kinect for Windows Sensor

Hardware and Input/Output specification for the Microsoft Kinect itself is as follows:

- Communicates serially via USB
- The standard frame rate of 30 fps cannot be exceeded
- A display with a resolution of at least 640x480

Block Diagram



Software specification

Use-Cases

Interacting with system

4. Users can select items using their hands to hover over an item on screen
5. Users can use a keyboard or controller to select items

Speech recognition

1. The user should speak clearly
2. The software will interpret their voice and produce an appropriate output

Ability Assessment

1. The users ability to pronounce a word will be assessed by the speech recognition of the Kinect
2. The users ability level will be assessed by a spelling test in order to determine their level in reading and vocabulary

Non-Functional Requirements

Information is secure - Information from profiles will be available only to teachers, children, and parents

Extensibility - Words with Kinect can be extended with more games and lessons and the software allows for future additions

Quality - Words with Kinect will accurately measure the child's performance

Accuracy - Gesture and voice recognition will be accurate

Test specification

Validation - Does the software meet user requirements?

The Microsoft Imagine Cup competition can be seen as the audience, but the overall end audience will be the teachers and parents. The application will be used by children. We can validate that our project meets the Microsoft Imagine Cup requirements by viewing the guidelines and rules on the imagine cup website. These rules can be found here:

http://www.imaginecup.com/Custom/Index/Rules#?fbid=6bRCNwfRB3m

We can validate that our project meets requirements from teachers and parents by conducting

customer surveys with them and running beta tests for our software with real classrooms.

Verification - Does the software meet specifications?

To verify that our project works according to specifications, we will use several different testing methods. We will use unit tests to test the functions in our software. We will test components or modules with regression testing. Our project will initially be ad-hoc tested to ensure that basic functionality is found

Simulation/modeling

Written Spelling Assessments

We could give these to kids to model and refine our spelling assessments in the game.

Paper Games

We could have pieces of paper with words and categories and have kids play these games so we can simulate how our games would work with the Kinect and decide what works and what doesn't.

Prototyping/testing

Prototyping

Small sample games will be designed based on the information we learn from Dr. Bear. The game is then taken to Dr. Bear where we will decide if they are teaching the children properly.

Testing

We can test against ourselves and well as small children. Children should be able to navigate the menus and use the interface regardless of their reading level.

Appendix III

Words With Kinect

Other Considerations

Unimplemented Features

Lessons

We had originally planned to have lessons to go along with the games to help kids learn further. We never got around to finalizing how this feature would work and how to implement it in a way that made sense with standard education practice. We focused on the games because we had those approved by Dr. Bear and we knew they would be important.

Matching Game Picture Refresh

The matching game was designed so that after all pictures are matched, 6 new images appear for matching. Due to a shortage of time, this was not implemented.

Picture Database

The picture database would've been used in the matching games and other future games that would utilize pictures from Dr. Bear's book *Words Their Way*. Due to a shortage of time, this was not implemented.

Spelling Assessment

This was designed to test the child's reading level. After this was determined, difficulty levels could be set for the games and lessons in that child's user profile (see *User Profiles* below). Due to a shortage of time and complications with the speech recognition API, this was not implemented.

Spelling Game

Due to complications with voice this feature was not implemented. We began experimenting but the main issue is that the Kinect was not able to pick up single letters.

Demo/Tutorial

We discovered that a demo was needed after giving a demo to kids to children. Due to a shortage of time, this was not implemented, though we do not feel like this would be very hard to implement.

User Profiles

User profiles with their own data was planned in the beginning. Due to a shortage of time, this was not implemented.

Progress Tracking on a Website

We had planned to create a website to accompany the user profiles for our software. Through this website, children, parents, and teachers would have been able to track their progress and growth online. Due to a shortage of time, this was not implemented.

Things We learned

Kinect Dragging

The Kinect has grab gesture recognition built in to the SDK. This is however only implemented fully with an object called a scrollviewer. A scrollviewer is a GUI object that holds more content than what fits on the screen and is able to be grabbed and scrolled to reveal more content. So grabbing was natively intended to be for scrolling these viewers. It took a while to figure out how that scrollviewer utilized grab events and related code and to translate that to dragging a button. There weren't any easy to follow solutions that others had found to this problem and we mostly had to figure it out on our own. This was a big breakthrough for the word sort game since it is essential to how it is played.

Speech Recognition

As mentioned above we had a plan and had begun implementation of a spelling game. While working on this we found that the speech SDK for the Kinect works best when recognizing full words. When given single letters it's word recognition confidence is not nearly as reliable.