THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ATTENTION TRAINING APP FOR IOS

ROBERT DICK
SPRING 2014

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Computer Engineering
with honors in Computer Engineering

Reviewed and approved* by the following:

John Hannan
Associate Professor of Computer Science and Engineering
Thesis Supervisor

Lee Coraor
Associate Professor of Computer Science and Engineering
Honors Adviser

* Signatures are on file in the Schreyer Honors College.

ABSTRACT

Mild cognitive impairment (MCI) is a clinically diagnosed ailment affecting 10 to 20 percent of people aged 65 or older, and there are currently no FDA-approved medications for treating it. As such, alternative treatment methods are often used to improve or slow the decline of the attention and memory abilities of those suffering from MCI. This thesis documents the design, implementation and evaluation of a mobile application designed to act as an evaluation and treatment tool for MCI.

The Attention Training App (ATA) is an iOS application designed for use with Apple® iPad® devices. In the short term, it is intended to be used in a study that will track the progress of participants as they use the app. The main focus of the app is the completion of a mental exercise called n-back task completion. Completing n-back task exercises has been shown to increase the memory and attention abilities of the human brain. The reason for turning this type of training into an app is to make the training more easily disseminated and to allow for automation of collection of user data. The resulting app is fully functional in administering and collecting user data for 1-back and 2-back task completion with visual, auditory, and dual stimuli with adaptive logic for varying levels of MCI.

TABLE OF CONTENTS

List of Figures iv

Acknowledgements v

Chapter 1 Introduction 1

- The Problem 1
- The Treatment 2
- Why Make an App 2
- Selecting App Design Parameters 3

Chapter 2 Development Process 1

- Formulating the Idea 1
- Choosing a Platform 1
- Setting Specifications and Goals 2
- Developing the App 2
- Testing 3
- Deployment 3

Chapter 3 Description of the App 4

- Start Screen 5
- Training 6
- Instructions 7
- User Progression 8
- Settings 11

Chapter 4 Discussion of the Code 12

- Code Organization 12
- Challenges 13
- Other Interesting Facets 16

Chapter 5 Design Choices and Motivations 18

Chapter 6 Extending the App 20

- Interactive Tutorials 20
- User Data Retrieval 20
- More In-depth User Data 22

Chapter 7 Future Plans for Testing and Conducting a Study 23

Beta Testing 23
Conducting a Study 24

Chapter 8 Conclusion 25

Appendix A Attention Training App Specifications (*as provided by Dr. Hill*) 26
Appendix B SSRI Proposal (Partial) (*as provided by Dr. Hill*) 31
BIBLIOGRAPHY 33

LIST OF FIGURES

Figure 1. Start Screen 5

Figure 2. Training Screen 6

Figure 3. Instructions Screen 7

Figure 4. User Progress Table 8

Figure 5. User Progress Session Details 9

Figure 6. User Progress Graph 10

Figure 7. Settings Screen 11

ACKNOWLEDGEMENTS

I would like to thank Dr. Hannan for being flexible and working with me to find a thesis idea that suited my desires to work in applications programming. I would also like to thank Dr. Nikki Hill for developing the idea to make an app for her research in mild cognitive impairment and for providing guidance and feedback on the app along the way.

Chapter 1

Introduction

The goal for this thesis is to solve a real-world problem through the design of a mobile application. Dr. Hill, currently in the School of Nursing at Penn State, has a problem she believes can be solved with an app. The formation of this thesis then came from the idea that advances in modern technology can be used to expand upon treatment for people with medical issues. Specifically, Dr. Hill has been studying and researching Mild Cognitive Impairment (MCI) and developed an idea for a mobile app that can be used as an alternative form of treatment. The app would be a digital version of an attention training activity called n-back task completion. Through her research, she compiled a set of specifications and settings for an app designed to suit the needs of varying levels of cognitive impairment and stimulate the user's brain so as to act as treatment for MCI. Her specifications are provided in Appendix A. The app has been developed to be used in a study and possibly further as a recurring form of treatment for people with MCI.

The Problem

According to the Alzheimer's Association MCI is a clinically diagnosed disorder that "causes a slight but noticeable and measurable decline in cognitive abilities, including memory and thinking skills." Furthermore, MCI has been found to increase the risk of developing Alzheimer's or other forms of dementia. Moreover, studies suggest that MCI affects 10 to 20 percent of people aged 65 or older. There are currently no FDA approved medications for treating MCI, but evidence suggests that daily cardiovascular exercise and participating in mentally stimulating and socially engaging activities may slow the decline in thinking skills ("Mild Cognitive Impairment"). Given that in 2012 there were over 43 million people in the United States aged 65 or older, that means that MCI is affecting roughly 4 to 8 million elderly Americans ("USA"). Thus, there is a large market for developing and experimenting with forms of treatment.

The Treatment

The treatment method chosen for the app is called n-back task completion. In n-back task completion, the user is presented with a series of visual or auditory stimuli and is required to identify if the current stimulus matches the stimulus presented n intervals back. In dual n-back task completion, the user is presented both visual and auditory stimuli at the same time and could be required to identify a match in visual stimuli only, auditory stimuli only, or in both simultaneously. For instance, in 1-back task completion, a match occurs if the same stimulus is presented twice in a row. For 2-back task completion, a match occurs when stimuli are presented in an A-B-A pattern.

Why Make an App

Currently, administering n-back task completion as a form of treatment would require someone to be present to administer it. In this case, the treatment would be administered in the form of a desktop or online application. This is not a viable solution to treating MCI in someone's everyday life. Additionally, mobile versions of n-back task completion do exist, but they are presented as a game geared towards average mobile users, rather than as a treatment method geared towards those suffering from MCI. As such, there are currently no mobile versions of an n-back task completion application appropriate for use as treatment by those suffering from MCI. Creating an app that would allow for a treatment regimen to be specified ahead of time would enable the regimen to be completed on the patient's own, thus solving this problem. Hence, the goal of this app is to make n-back task completion as a form of treatment readily available and easily manageable.

Selecting App Design Parameters

The intention of the app is to take the n-back task completion treatment method and turn it into a simple, straightforward and easy-to-use mobile application that can be used in studies and possibly as treatment administered by a physician. From there, it was necessary to choose a single mobile platform, as developing for multiple platforms would extend the workload beyond the possibilities of this thesis. It was of our opinion that iOS has one of the simplest and easiest to navigate interfaces, as well as being extremely popular in the United States. Therefore, we decided the app should initially be designed for the Apple iPad. This also matched up with my skill set, as I am familiar with iOS application development.

Chapter 2

Development Process

The development cycle for any application can be broken down into distinct parts, each of which plays an important role in the outcome of the app. For this project, the distinct parts can be viewed as formulating an idea, choosing a platform, setting specifications and goals, developing the app, testing, and deploying.

Formulating the Idea

Dr. Hill developed the idea for the app through her research in Mild Cognitive Impairment. She realized that one of the types of treatment she was studying, n-back task completion, could easily be represented in digital form as a mobile application. From there, she contacted the Department of Computer Science and Engineering, and in turn they setup my connection with Dr. Hill.

Choosing a Platform

Based on the popularity of iOS and its simple, intuitive interface, we chose iOS as the platform for development. Furthermore, given the nature of the application, the iPad was a much more suitable target device than the iPhone or iPod Touch would have been. These decisions were further supported by the simplicity of the development process for iOS applications.

Setting Specifications and Goals

Hill created a detailed set of specifications (see Appendix A), which served as the basis for development of the app. Additionally, I set my own personal goals for the project which were as follows:

- Create an enjoyable user experience
- Make the app intuitive to use
- Use concepts or frameworks I had not previously used

With these things in mind, the actual development part of the process could begin.

Developing the App

My overall approach for the developing the app was to work from the outside in. By starting with the big picture and working down to the details, it ensured that I kept the end goal in mind at all times rather than getting lost in the minor details. I started by creating a shell of the main functionality and process flow of the app. This meant setting up each of the main screens, the navigation between the screens, and some of the basic functionality. Next, I worked on adding detail to the game screen, including animations and functionality. From there, I moved back and forth between adding to the game functionality, creating the flow between the different screens, improving the user interface, and designing the necessary data models. By working on each of these pieces together and intermittently, rather than any one piece all at once, I was better able to see how everything needed to work together. Additionally, it prevented me from attempting to complete one part without considering how it needs to interact with another part. Once the app was actually functional and fulfilled all of the minimum requirements, I began working on incremental improvements, additional features, and user interface polishing. These steps continued throughout the testing phase.

Testing

Incremental testing was done throughout the majority of the development phase, initially by myself alone and eventually by Dr. Hannan and Dr. Hill as well. The testing and development phases meshed together as they should and eventually became a continuous cycle. Formal testing, however, will not be done until after completion of the thesis work. Plans are in place for beta testing the app in the near future.

Deployment

Official deployment of the app will take place after beta testing is complete and improvements are made based on testing results. The current plan is deploy the app for in-house use in a human study.

Chapter 3

Description of the App

The app is an iOS application targeted for the Apple iPad line of products. Its main function is administering n-back task training and recording the user's progress as they complete the training. As per the app specifications found in Appendix A, the training can be set to take on the form of an 8-week progression of varying stimulus and back number, with adaptive logic to suit the needs of varying levels of MCI. Additionally, the user may choose particular settings for a practice session not included in the 8-week progression. The overall user experience with the app is described in the following sections.

Start Screen

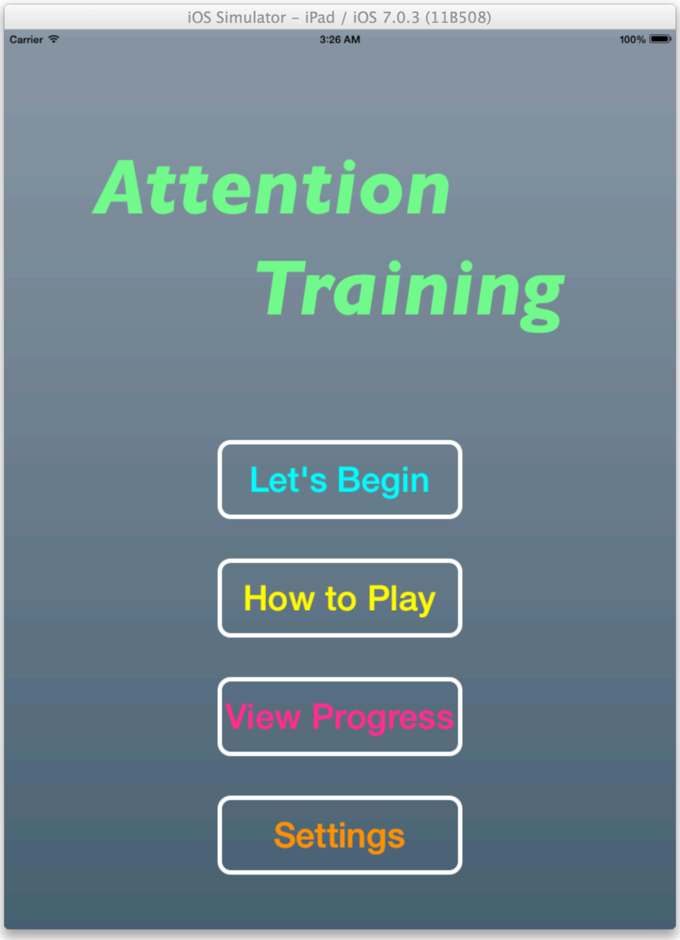


Figure 1. Start Screen

The user's experience with the app begins with a start screen, shown in Figure 1, which contains buttons for beginning the training, viewing the instructions, viewing past progress, and changing the app's settings. Tapping one of the buttons takes the user to the corresponding section of the app.

Training

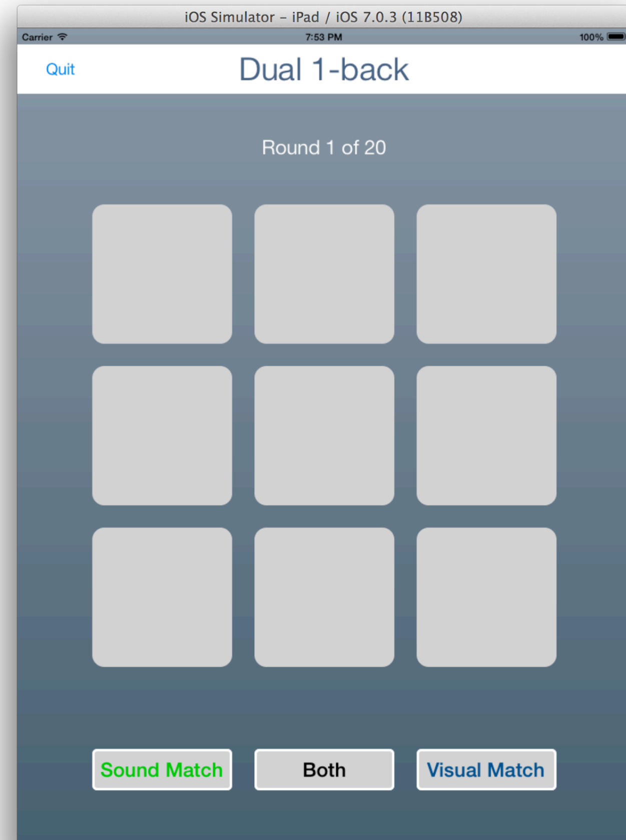


Figure 2. Training Screen

The training portion of the app is the app's main function and purpose. The user is presented a 3x3 grid of squares and 3 buttons for identifying matches: one for visual matches, one for auditory matches, and one for dual matches. An example of the layout of the training screen for round 1 of a dual 1-back training session is shown in Figure 2. Each training session consists of 20 rounds of training, but the user may choose to end training at any time by pressing the "Quit" button. Each round of training consists of 20 pairs of stimuli, one visual and one auditory, presented at 3-second intervals. For each stimulus pair, one of the outer eight squares will flash red and one of eight letters will be vocalized. The user is responsible for identifying matches in stimuli based on the settings for the current round. More details on the specifics of the stimuli, game settings, and matching requirements are described in Appendix A.

Instructions

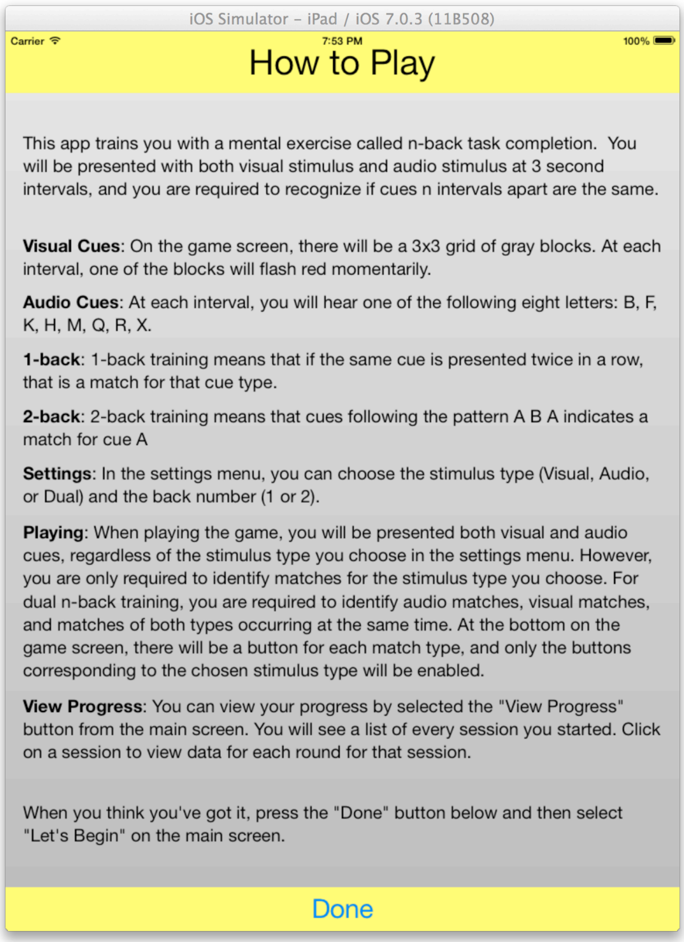


Figure 3. Instructions Screen

At the time of the writing of this thesis, the instructions screen is a simple text-only screen describing n-back task completion, what the user will be expected to do, and how the user can change the settings as shown in Figure 3. However, development of interactive tutorials to teach the user how training should be completed is currently under way.

User Progression

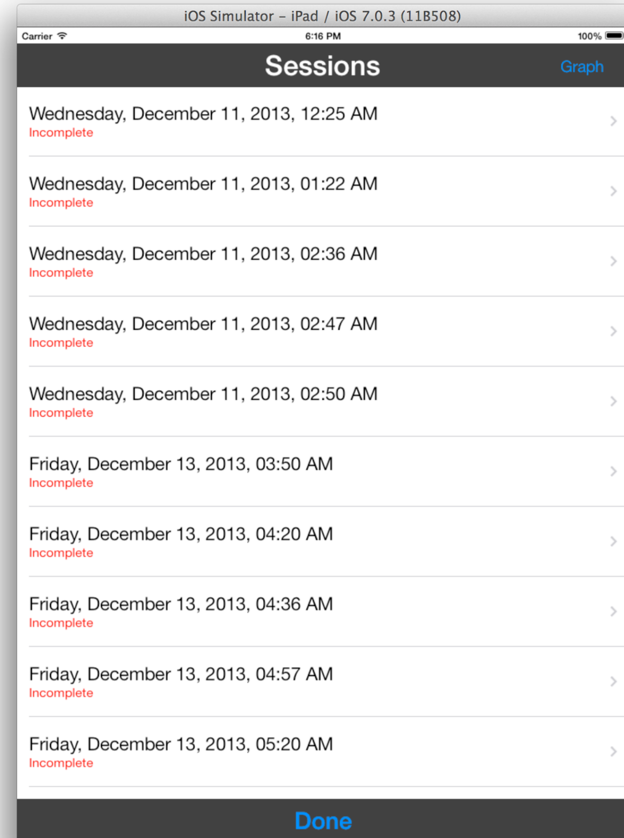


Figure 4. User Progress Table

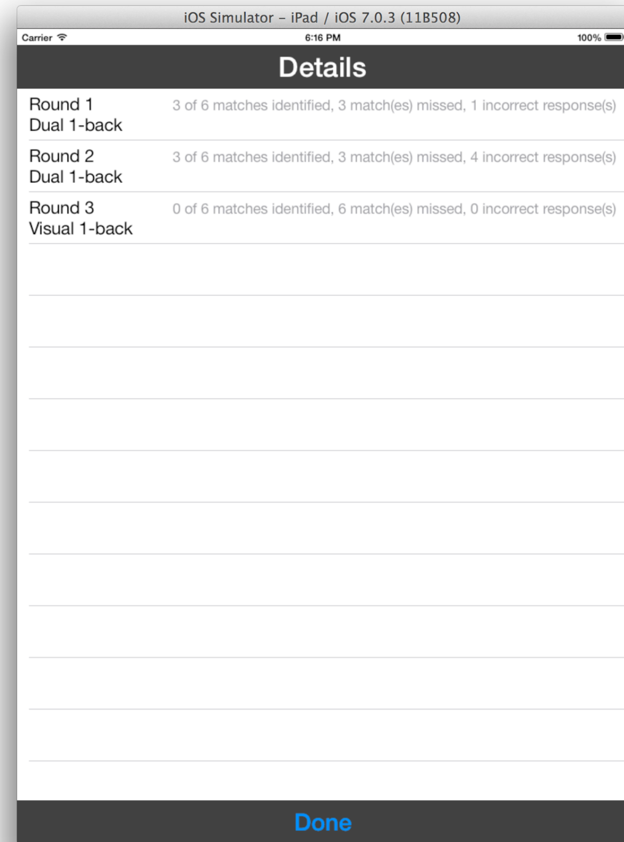


Figure 5. User Progress Session Details

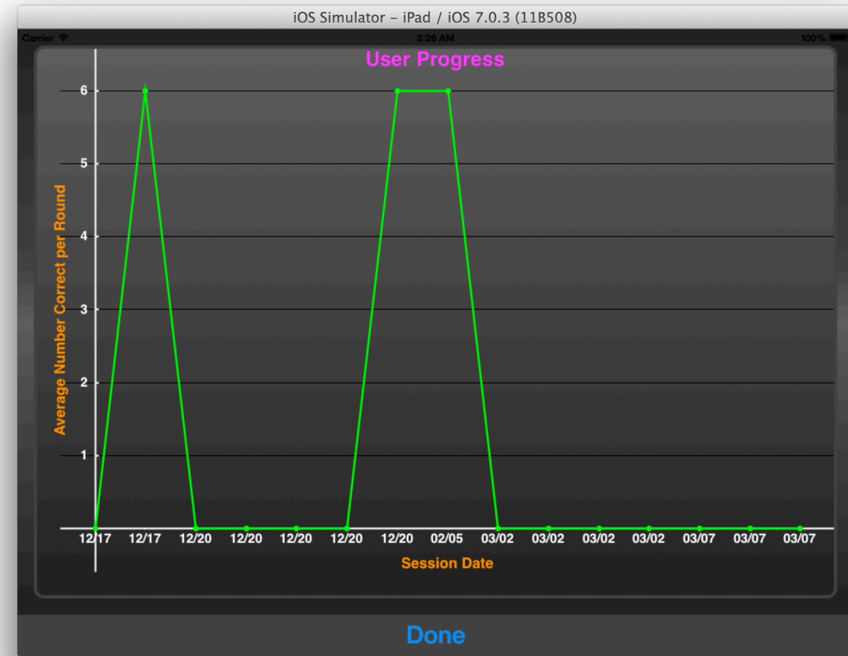


Figure 6. User Progress Graph

The user progress section displays a table of each of the training sessions the user has completed sorted by date as shown in Figure 4. Tapping on a session displays a table showing information about each round of training completed during that session as shown in Figure 5. Furthermore, a line graph of the user's progress over time can be accessed from the initial user progress screen as shown in Figure 6.

Settings

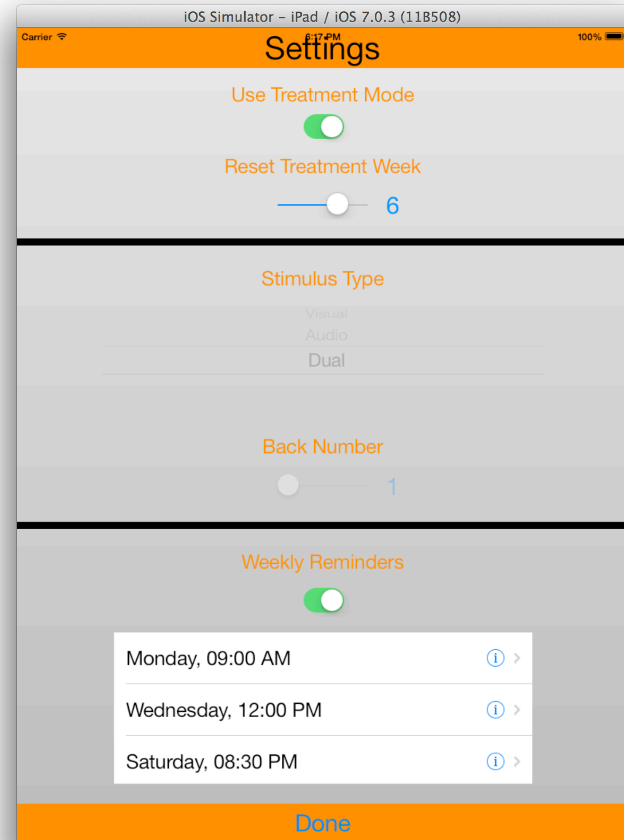


Figure 7. Settings Screen

The settings screen offers the user the ability to change a variety of settings related to the training portion of the app as shown in Figure 7. The first setting gives the user the ability to toggle using the app in treatment mode. When treatment mode is on, the user's training will follow the 8-week training progression described in Appendix A. The user may jump to a particular training week by moving the "Reset Training Week" slider. If treatment mode is off, the user may control settings for the stimulus type and back number for the next training session. Finally, the user may toggle weekly reminders, of which three may be set. When a reminder occurs, a local notification is delivered to the device regardless of whether or not it is currently in use, and tapping the notification automatically opens the app.

Chapter 4

Discussion of the Code

In general, the code for the app follows the standard iOS app design guidelines set forth by Apple. For the most part, the challenges faced in writing the code were what could be considered the normal challenges of making an app, but there were some more difficult challenges as well. Furthermore, some of the desired features led to the use of third-party frameworks and lower-level control.

Code Organization

The code is organized using the Model-View-Controller organization scheme. There are two models, one view, and nine controllers, with a variety of supporting files as well.

The models are used to keep track of user progress data. One model represents the entire storage of user data for a particular instance of the app, and the other model represents the data from an individual training session. The outer model is a singleton, while there are many instances of the inner model. Each time the user participates in a training session, the settings for that session and the number of correct responses and incorrect responses for each round are stored in variables as part of the model upon termination of the session.

The “view” is a single storyboard made up of a variety of views, some of which are full-sized and some of which are form sheets. Navigation starts with the start screen as described in Chapter 3 and all other views are presented modally. While all of the views are laid out in the storyboard, much of the final user interface is also manipulated in the controllers, including the background shading for the start screen and training screen, as well as most of the buttons and the training screen visual stimuli.

Each of the controllers controls a different screen in the app. Many of the controllers share information (objects) with one another before a transition to another view is made.

Challenges

The typical challenges encountered in writing the code were deciding how to organize the model, choosing appropriate transition types between screens, ensuring proper sharing of objects and data, making animations behave nicely, and developing a proper system for organizing and managing the unique settings used. The big challenges were writing the algorithm that generates the sequence of stimuli and controlling the timers used for the training.

Sequence of stimuli

The stimuli presented to the user are meant to be essentially random with a fixed number of matches (six) per round. In order to ensure this occurs and be able to verify its accuracy, a complex algorithm was required. First, it is important to note that in order to guarantee the fixed number of matches and be able to verify them, all stimuli for a training round must be selected before the round starts. Initially, a simple algorithm was used that randomly selected a stimulus for each of the 20 intervals in any given round, while ensuring no matches initially. Then, six intervals were randomly chosen to be intervals in which a match occurred, and the match interval numbers chosen were stored for verification later. Finally, the stimulus from the interval *n* prior to the selected interval (for *n*-back training) was copied to the selected interval, thus resulting in a match at the selected interval. However, this algorithm is not robust and results in the possibility of unintentional matches. For example, consider the sequence A-B-C-B-D. Assuming we want to do 1-back training and assume that the 3rd interval is chosen at random to be a match. Based on the algorithm, the B from the 2nd interval would be copied to the third interval, making the new sequence A-B-B-B-D. However, now we have a match on both the 3rd and 4th intervals, but we did not intend to have a match on the 4th interval. This is bad, as the user will hopefully identify this match, but the system won't have the 4th interval marked as a match interval. The problem is further compounded when considering dual n-back training. In this scenario, after random visual and auditory stimuli are generated, a fixed number of matches must spread randomly across visual only matches, auditory only matches, and dual matches. This means an algorithm is required that can fix all three types of matches independently without creating accidental dual matches.

The solution comes in the form of an extensive series of accidental match checks. For visual match only or auditory match only rounds, this means that after selecting the match intervals each time the stimulus sequence is modified a check is performed to make sure there was no accidental match created on either side of the modified interval. For instance, in our previous example, after modifying the sequence to be A-B-B-B-D a check would be done to make sure the modification of the 3rd interval did not create a match in the 4th interval. In this case, of course, it does. If this occurs, the interval where the accidental match was created will be changed to another random stimulus. Let's say that D gets randomly chosen, making the sequence A-B-B-D-D. Now an accidental match has been created in the 5th interval. However, the algorithm specifies that anytime the sequence is altered, a check for accidental matches is performed. Thus, after verifying that the newly selected stimulus for the 4th interval does not match the 3rd interval, it would check that the 4th interval stimulus does not match the 5th. If it does, it would continue to generate a random stimulus for the 4th interval until it does not match either. If this algorithm is executed in sequential order starting from the lowest interval match continuing up to the highest interval match, we are guaranteed not to have any accidental matches.

For dual n-back training, the algorithm gets more extensive. The additional complication occurs in selecting the match intervals. After randomly distributing the six matches amongst visual only, auditory only, and dual matches, the dual match intervals are chosen first. Then, visual match intervals are chosen such that none are the same as the dual match intervals. Finally, auditory match intervals are chosen such that none are the same as the dual or visual match intervals. The next complication arises because the dual matches require altering both the visual stimulus sequence and the auditory stimulus sequence. Since the algorithm requires all sequence altering must be done in sequential order, the matches for all three types must done in overall match interval order, rather than doing each match type individually. For instance, if dual matches should occur at intervals 3 and 9, visual matches at 4 and 6, and audio matches at 2 and 10, the sequence altering order should be: audio(2), both(3), visual(4), visual(6), both(9), audio(1). This requires storing a small amount of extra information when matches are chosen such that they can be properly sorted when it comes time to alter the sequences. After this, the same sequence-altering algorithm is used as described for the visual match only or audio match only case.

Controlling the timers

The research backing the settings for the training dictate a specific timing sequence be upheld for presenting the stimuli. To achieve this, the app uses a built-in timer object that can be set to go off at a regular interval. Each time the timer goes off, stimuli are presented. A more in-depth view of the timers is that they are based off of specific date-times, down to the millisecond. The timer object knows its timing interval and each time it goes off, it sets a reminder, so to speak, that far into the future. Once that date-time occurs, the timer goes off again and sets another reminder.

The problem lies in that when the user presses the quit button, it prompts the user to verify that they want to quit, or otherwise continue training. Thus it is necessary to pause the gameplay while they make a decision. However, there is no built-in pause method for the timer objects, and the timer objects are what control the gameplay. The solution is to save both the current date-time and the next reminder date-time when the timer should be paused, and then manually set the next reminder to be very far into the future. Then to resume the timer later, we calculate the time length since the timer was paused, add this time to the previous reminder date-time, and manually set the next reminder to be this new calculated date-time. After the manually set reminder goes off the next time, the design of the timer specifies that it will return back to its original timer interval.

Other Interesting Facets

Including some of the features that were desired and creating an aesthetically pleasing user interface required the use of a third-party framework and some lower-level code. We believed that viewing the user progress in graph form would be a nice additional feature to the table view of user progress, but there is no built-in way to create custom graphs in iOS. Therfore, the app utilizes the third-party framework CorePlot. CorePlot supports the creation of fully customizable bar graphs, scatter plots, and pie charts. Thus, user progress can be viewed in the form of a customized line-graph as shown in Figure 6.

Furthermore, the interface builder in Xcode, the application used to develop iOS apps, does not expose all of the properties of all objects. As such, certain visual effects must be programmed by accessing lower-level properties of certain objects. One example of this is the slight gradient background that can be seen in Figure 1 and Figure 2. These backgrounds are created as layers by specifying various colors and alpha levels at specific vertical levels on the screen and then blending them together. The layers must then be inserted on top of the original background. Another example is the visual stimulus effect, which is one of the rounded gray squares in Figure 2 flashing red momentarily. The red flash is actually a timed transition from gray to red accompanied by the addition of a white shadow around the perimeter to give the appearance that the square is glowing. Both of these parts required setting a variety of properties on the square objects, including the fade in and fade out lengths and the shadow radius, position, and brightness. A significant amount of time was dedicated to adjusting these settings to give the optimal, smooth flashing effect.

Chapter 5

Design Choices and Motivations

One of the most important things to keep in mind when designing any app is the target audience. Given that the majority of all cases of MCI occur in elderly people, the app has been geared towards that demographic. With this in mind, the design of this app had to accommodate for users who may have difficulty seeing and hearing, as well as a lack of experience with technology. Furthermore, given that the users will also be people with MCI, it is important to avoid causing any cognitive overload from the app experience. It is also imperative to minimize any frustration the user will have in using the app. Thus, the app incorporates the following key design concepts.

Simple interfaces

The interfaces are kept clean and simple, yet still aesthetically pleasing. Varying shades of gray are used for the backgrounds, and no unnecessary visuals or sounds are presented at any time. A variety of colors is also used for button text to better distinguish between different buttons. Keeping the design minimal yet distinctive eliminates any clutter that may cause cognitive overload in the user.

Large and simple fonts, buttons and icons

Clean, simple fonts in large typeface are used to make reading them easier and to avoid any unnecessary eyestrain to the user. Furthermore, large buttons are also used to ensure that interacting with the app is easy. Using large and simple icons provides clear feedback to the user and aids in the process of understanding the flow of the app. Overall, ensuring that the visuals are clear will reduce any frustration and difficulty experienced by the user.

High contrast

In general, it is much easier to distinguish visuals that have a high contrast with the background than those with low contrast. Thus, color schemes were chosen such that a high contrast is present wherever possible.

Clear audio

Audio recordings were done such that the audio is clear and distinct to minimize any difficult the user has in understanding it.

Clear instructions

Keeping the instructions clear and concise allows for the best chance that the user will understand them and be able to use the app properly.

Consistency

Maintaining consistency throughout the app creates a more pleasing user experience. Thus, the app maintains consistency in button style, text style, and navigation throughout the different portions.

Intuitive navigation

Navigating the app is simple and intuitive in order to make the user experience simple. Button text was chosen to be concise and clearly represent what action pressing each button will have. Furthermore, navigation buttons are placed in easily identifiable locations and consistency in placement extends throughout the app.

Chapter 6

Extending the App

The app as it stands, while it is fully functional and meets all of the specifications, has marked room for expansion and improvement. The development process does not need to end in the current state and the app can continue to grow and be developed into an even more powerful and useful tool for treating and tracking cases of MCI. Some of the most useful and immediately implementable feature and usability extensions follow.

Interactive Tutorials

Currently, there is a simple instructions screen that describes the usage of the app and the settings that can be changed. This screen uses text only. For a more effective learning experience, interactive tutorials can be designed that visually and audibly demonstrate the way training is completed. For instance, after reading about the basic rules a short demo can show what a visual match looks like and then require the user to respond to it by pressing the visual match button. The same could be done for an auditory match and a dual match. The same process can repeated for both 1-back training and 2-back training so that users understand the difference.

User Data Retrieval

The app currently does not support any methods for retrieving the user data that is collected. This user data could be valuable to the doctors and/or researchers who administer the app as a form of treatment or as part of a study. Currently, the user data can be viewed on each individual copy of the app, but support for transferring the data to another location could be added. The two main possibilities for transferring the data are direct, individual transfers from the user device to an administrator's device and transfers from each device independently to a single offsite database. Both of these solutions would require reorganizing the data structures holding the user data into a form more suitable for that solution.

Individual Transfers

With individual transfers, support would be added to the application such that two devices running the application can communicate back and forth via Bluetooth. The administrator would have special privileges allowing it to receive and view user data from multiple users. To receive the data, a Bluetooth connection would first be established between the patient/participant's device and the administrator's device. The patient/participant's device would then send the user data to the administrator's device, where it could then be viewed among other data from other users. This scenario would require an authentication procedure that would distinguish an administrator from a patient/participant in order to ensure privacy and protection of user data. An advantage with this solution is that data could be collected onsite for medical studies and no Internet connection would be required. As elderly people often do not have Wi-Fi access readily available, an Internet-free solution also eliminates problems with uses for the application that include at-home treatment. The disadvantage here is that the administrator would then have to meet with the patient/participant in person each time they want to retrieve new user data, which may be an inconvenience.

Transfers to Single Database

Transfers to an offsite database would require Internet connection. Once the device established an Internet connection, it would connect to a server hosting the database of user data. User data would be transferred to this database when possible, and administrators would have access to the database to review user data. A possible iteration of this scenario discussed in the development process would use REDCap (Research Electronic Data Capture), a web-based application designed for supporting data captured from research studies, in order to store the database of user data. Penn State University is one of over 300 institutional partners associated with REDCap, making it a realistic choice. The advantage with this solution is that the user data can be collected without needing to meet the patients/participants in person, and all of the user data will end up one place that can be accessed from anywhere with a secure Internet connection. The disadvantage here is that Wi-Fi access may not be readily available for all users of the app, meaning the user data simply could not be collected.

More In-depth User Data

Currently the app collects data on the number of correct matches identified, the number of incorrect responses, and the number matches missed for each round in each session and what the settings were for the session. However, another very useful statistic would be to collect timing information. The app could record the length of time between the stimuli and the response from the user for each response. This data could be used for further analysis into the mistakes users make and determine if any patterns exist. Furthermore, the app could record the exact sequence of stimuli for each round to determine if certain combinations or sequences of visual and auditory stimuli make matches harder to identify than others.

Chapter 7

Future Plans for Testing and Conducting a Study

Plans are already in place for beta testing to be done with app within the next month. After analyzing the usability of the app from the beta testing, improvements and adjustments will be made. Then, eventually a study will be conducted to test the effectiveness of the app as a treatment method for MCI. The long-term goal then is to have the app become a widely used and effective method for treating MCI.

Beta Testing

A description of the plans for beta testing can be found in the study proposal attached as Appendix B. In the proposal, the beta testing is described as follows:

> "We will purposively recruit older adults with different levels of mobile technology experience, ranging from individuals with their own iPad devices to those with no previous experience. A minimum of eight participants will be recruited from Centre County Senior Centers as well as weekly iPad training classes held at a local electronics store. Following screening and enrollment, a member of the study team will meet with each participant to facilitate testing of the ATA, take field notes pertinent to app and device usability, and to conduct a subsequent audio-recorded, semi-structured interview regarding use of the ATA. Additional measures will include a demographic questionnaire, brief cognitive screen, and a computer and technology experience questionnaire in order to contextualize user experiences with the ATA. Quantitative usability data will be captured as the app is accessed and used. Among participants with their own devices, the ATA will be installed using TestFlight, a mobile app beta testing service; all others will use a

study-provided iPad. The entire visit will take approximately two hours, which includes time to complete the baseline study measures, ATA testing, and post-test interview. A trained individual external to the research team will be hired to transcribe audio recordings into verbatim transcripts. Open-ended data from participant interviews and field notes will be analyzed to uncover common themes related to feasibility of the ATA. Usability data collected during the use of the ATA will be analyzed and used to generate a data-based report which will guide revisions of ATA content and protocols."

Conducting a Study

Beyond the beta testing, the long-term goal is for the app to be used in a formal study wherein participants will progress through the 8-week training session and data will be collected to track their progress throughout the training. In this scenario, the settings screen would be modified, disabled, or removed to ensure that the participants of the study complete the 8-week training as intended and as described in Appendix A. The results of the study will hopefully prove the app to be useful both as a treatment for MCI and as a data collection tool for further research into MCI and finding better ways to treat it.

Chapter 8

Conclusion

The success of this app cannot yet fully be measured as human testing has yet to take place. However, it is worth noting that the app as it stands meets the specifications requested by Dr. Hill in Appendix A, and furthermore meets the personal goals discussed in Chapter 2. The app successfully runs on various Apple iPad devices, can administer n-back task training with a variety of settings, tracks user progress, and has a clean, intuitive user interface. Further success will be measured by the results of the beta testing described in Chapter 6.

Appendix A

Attention Training App Specifications (*as provided by Dr. Hill*)

The CREATE for MCI *Attention Training App* (ATA) will use the Dual *n*-back task as a basis for attention training. In the *n*-back/dual *n*-back procedure, participants are presented with a 1) visual, 2) auditory, or 3) both visual and auditory stimuli (dual *n*-back). Participants must respond on the screen whenever the current stimulus matches the target stimulus presented *n* trials back. In the dual *n*-back task, the current stimuli could match the target visual or auditory stimuli or both. See Jaeggi et al. (2007,2008) or Lilienthal et al. (2013) for more details on the procedure. [1,2]

In order to accommodate the varying cognitive abilities of study participants, the ATA will be adaptive based on user success as well as provide incremental challenge with task mastery. The goal is for all participants to reach the dual 1-back task training throughout the study period, but that may not be feasible for some individuals. Training on individual visual and auditory components are considered preparation for the dual-task component. Functionality will be included that will provide for automatic adjustments to the difficulty level, tutorial reminders, etc. to facilitate success in using the app and avoiding frustration. Additionally, the app will accommodate higher levels of *n*-back tasks above the 1-back, but it is unknown whether any participants will achieve these higher levels.

DESCRIPTION OF ATA COMPONENTS

1. **Tutorials**
   - To be created after completion of other components
   - Will present instructions with simultaneous visual and audio components (show the screen, voiceover instructions). See "IQ Boost" app for an example.
   - Separate Tutorials for:
     - Visual 1-back
     - Visual 2-back
     - Audio 1-back

- Audio 2-back
- Dual 1-back
- Dual 2-back
- At this time, it is not anticipated that further levels will be needed (*SSRI pilot may help inform this*)
- Brief "Summaries" of Tutorials will also be created. These will be presented at the beginning of each session, based on the level achieved during the last session. This will serve as a reminder of where the participant left off in their progress. Once the session begins, the adaptive logic described in the next section (**Progression of ATA Components**), will determine if/when more detailed instruction should be provided or the participant should move back to an earlier level of the ATA.

2. Visual Stimulus

- Welcome/Introduction to Session
    - Consists of a welcome message for today's session, full Tutorial or Summary Tutorial presentation based on progress achieved to date, and a summary of today's session (e.g., "You will play 20 short game sets today, which will take about 20 minutes. When you are ready to begin, press the START button below.")
- A box is presented in 1 of 8 locations at random around the periphery of the screen (a 3 x 3 grid in which the middle location is not used)
- Each trial lasts 3 seconds:
    - 500-ms item presentation interval, followed by
    - 2500-ms inter-stimulus interval during which the participants should respond
- Each block consists of 20 trials
- Each session consists of 20 blocks
- Each block of 20 3-second trials will have 1/3 (6 or 7) of the presented stimuli be a positive match (matched according to *n*-back condition).[3] Matches to be determined at random.
- Participants respond by pressing the "Position Match" button on the screen when the current box location matches the box location presented *n* trials back
- Correct responses are identified by a green checkmark flashing on the screen
- Incorrect responses are identified by a red X when an incorrect response is given. If a match was missed, an icon of either an eye or an ear will be displayed to indicate the miss
- After completion of each block:
    - Brief positive feedback presented, e.g. "Good job! 1 of 20 games for today completed."
    - Based on the adaptive logic described in the next section (**Progression of ATA Components**), determination is made regarding:
        - Increase/decrease/maintenance of task difficulty
        - Repeat presentation of Tutorial components
- Each session consists of 20 blocks
    - Therefore, each session will consist of 20 minutes of active time on task, plus any tutorial time and brief time spent on feedback screens

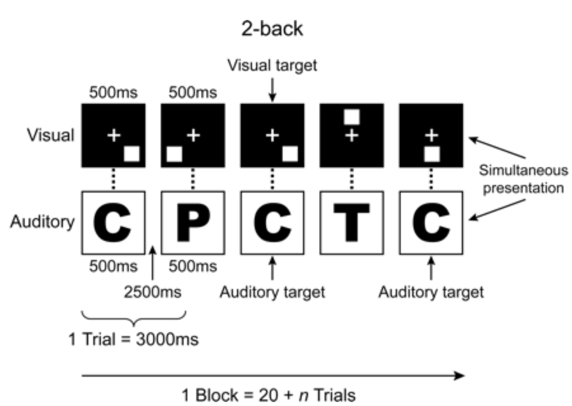
- After completion of the session, participant is presented with a screen giving something like the following:
    - Indication of end of daily session, e.g., “That is the end of your training session for today. Great job!”
    - Summary of any positive progress made, if any, e.g. “Today you made it to the next level”, or “Today you had 20% more correct responses than during your last session.”
    - Instruction/Pictures of how to end session, e.g. “You may now return to the computer’s main screen by pressing the button below, or the computer will return to the main screen on its own after several minutes. Please return the computer to its docking station if you removed it to complete this session. Thank you!”

3. Auditory Stimulus

- Welcome/Introduction to Session same as Visual Stimulus condition
- One of 8 letters is presented at random by audio stimulus: B, F, K, H, M, Q, R, X.[4]
- Participants respond by pressing the “Sound Match” button on the screen when the audio stimulus is the same as the audio stimulus presented *n* trials back
- The following are the same as the Visual Stimulus condition: timing, presentation, and randomization of stimuli; correct/incorrect response feedback; feedback after block completion and adaptive logic for determining next block; end of session feedback and instructions

4. Dual Task (Visual + Auditory)

- Welcome/Introduction to Session same as Visual and Audio Stimulus conditions
- On each trial, participants are presented simultaneously with a visual and an auditory stimulus; these follow the same conventions as described in their respective sections above
- The current stimuli could match the target visual or target audio stimulus or both
- Participants respond by pressing the “Visual Match” or “Sound Match” or “Both”
- The following are the same as the Visual Stimulus condition: timing, presentation, and randomization of stimuli; correct/incorrect response feedback; feedback after block completion and adaptive logic for determining next block; end of session feedback and instructions
- Visual representation of Dual Task from Jaeggi et al. (2008):



PROGRESSION OF ATA COMPONENTS

- Adaptive logic for ATA includes both within- and between- component decisions regarding increasing, decreasing, or maintaining task difficulty
- The Visual and Auditory Stimulus Components alone are considered training in preparation for the Dual Task Component; therefore, the initial presentation of these follows a fixed rather than adaptive between-component logic
  - Week 1 (3 sessions): Visual 1-back only
  - Week 2 (3 sessions): Visual *n*-back beginning with 1-back, adaptive within-component logic:
    - Fewer than 3 errors within each of the first 10 blocks: Increase *n*-back level by one for the duration of the session or until the decrease *n*-back criteria below is reached
    - More than 5 errors within one block = decrease *n*-back level by one (unless at 1-back, then remain the same)
    - Otherwise, *n*-back level remains the same
    - Subsequent sessions within this week should begin with the ending *n*-back level of the previous session
  - Week 3 (3 sessions): Auditory 1-back only
  - Week 4 (3 sessions): Auditory *n*-back beginning with 1-back, adaptive within-component logic:
    - Fewer than 3 errors within each of the first 10 blocks: Increase *n*-back level by one for the duration of the session or until the decrease *n*-back criteria below is reached
    - More than 5 errors within one block = decrease *n*-back level by one (unless at 1-back, then remain the same)
    - Otherwise, *n*-back level remains the same
    - Subsequent sessions within this week should begin with the ending *n*-back level of the previous session
  - Week 5 (3 sessions): Dual Task 1-back:
    - If more than 5 errors within one block, return to Visual *n*-back beginning with 1-back, adaptive within-component logic, until end of session
    - Each Week 5 session should begin with Dual Task 1-back, even if they returned to Visual *n*-back in the previous session
  - Week 6-8 (3 sessions per week): Dual Task *n*-back beginning with 1-back, adaptive within- and between-component logic:
    - Fewer than 3 errors within each of the first 10 blocks: Increase *n*-back level by one for the duration of the session or until the decrease *n*-back criteria below is reached
    - More than 5 errors within one block:
      - Decrease *n*-back level by one for anything other than 1-back
      - If 5 errors within one block for Dual Task 1-back:
        - Return to Visual 1-back, adaptive within-component logic, until end of session
        - At beginning of following session, start with Visual n-back:
          - Fewer than 3 errors for 10 consecutive blocks: Dual Task *n*-back beginning with 1-back, adaptive within- and between-component logic as above

        - Otherwise, remain at Visual n-back
    - Subsequent sessions within this week should begin with the ending level of the previous session

DATA COLLECTED BY ATA

- All data regarding correct/incorrect responses for each level, progression and timing of progression through components

Thoughts – if they exit the app early, will it only open with a new session?

References

1. Jaeggi SM, Buschkuehl M, Jonides J, Perrig WJ. Improving fluid intelligence with training on working memory. *Proc Natl Acad Sci U S A.* May 13 2008;105(19):6829-6833.

2. Lilienthal L, Tamez E, Shelton JT, Myerson J, Hale S. Dual n-back training increases the capacity of the focus of attention. *Psychon Bull Rev.* Feb 2013;20(1):135-141.

3. Jaeggi SM, Buschkuehl M, Etienne A, Ozdoba C, Perrig WJ, Nirkko AC. On how high performers keep cool brains in situations of cognitive overload. *Cognitive, affective & behavioral neuroscience.* Jun 2007;7(2):75-89.

4. Kane MJ, Conway AR, Miura TK, Colflesh GJ. Working memory, attention control, and the N-back task: a question of construct validity. *J Exp Psychol Learn Mem Cogn.* May 2007;33(3):615-622.

Appendix B

SSRI Proposal (Partial) (*as provided by Dr. Hill*)

1. Title of Proposal: Feasibility Study of an Attention Training Application for Older Adults using Mobile Technology in Community Settings; **PI:** Hannan, J.
2. Key Words: Attention; Cognitive Training; Older Adults; Technology
3. Purpose: Preserving healthy cognitive function is critical for maintaining quality of life and controlling cost of care among older adults. Recent research has demonstrated the potential of cognition-focused interventions to maintain or improve elders' cognitive function. Attention training, and specifically adaptive dual *n*-back training, is one approach which has demonstrated not only task-specific performance improvement, but also generalization to higher order cognitive processes. The proliferation of mobile technology may address several limitations to implementation of attention training interventions in community settings. Development of an attention training application for mobile devices could facilitate integration into daily schedules, real-time monitoring of adherence and performance, intervention standardization, potential reduction in implementation costs, and opportunity for wide scale dissemination.

The purpose of this proposal is to develop and test the feasibility of an Attention Training Application (ATA) for community-dwelling older adults using mobile technology (iPads). Findings regarding the usability of the ATA will inform revision and refinement of the app in preparation for future efficacy testing. The results of this study will be used to support an NIH small grant proposal (R03) to conduct a pilot clinical trial to evaluate the effectiveness of the ATA in improving attention among adult adults.
4. Background and Description of Activities: Computer-based cognitive training, which involves repeated practice on tasks specific to particular cognitive domains such as memory or attention, has demonstrated the ability to improve task performance among older adults including benefits above traditional paper-and-pencil approaches. A limitation of cognitive interventions has been the lack of generalizability beyond the specific tasks trained. One specific method, however, adaptive dual *n*-back training, has demonstrated not only the ability to improve performance on the task trained, but also transfer effects to improvements in mental flexibility, focus of attention, and episodic memory. The cognitive domain of attention, the primary area trained in the dual *n*-back task, is a prerequisite for higher order cognitive processing and is essential for successful cognitive performance. For example, an individual will be unable to remember details of a specific event unless he/she was paying attention when it occurred. Attention includes the mental states and operations needed to detect stimuli, select stimuli over 'noise,' and manage resources for the detection and processing of competing stimuli. Deficits of attention lead to difficulties in everyday activities; therefore, improving attention among older adults using an attention training program may lead to real-world cognitive benefits.
Aim 1: To develop the ATA for use with older adults using iPad devices. Older adults are the fastest growing users of computers and the internet; 53% of Americans age 65 or older use the internet or email and 69% own a mobile phone. The use of a mobile app to deliver computer- based interventions provides for customization important to an older adult population such as simple design, consistent screen navigation, and content limited to the task at hand. Furthermore, an app format supports the potential for future wider-scale implementation and dissemination of the intervention. To address this aim, we will translate the adaptive dual *n*-back procedure well- established in the scientific literature for use on desktop computers to an iPad application format. The ATA will include on-screen progressive training on use of the app as well as adaptive responses to task difficulty based on user performance. The goal of

these adaptations is to ensure users understand the task demands while limiting potential frustration in individuals with different cognitive capacities and familiarity with technology.
1""
Aim 2: To examine the usability of the ATA among older adults. To address this aim, we propose a mixed-methods pilot study to test the usability of the ATA among community- dwelling older adults. These findings will be used to: 1) guide revision and refinement of the ATA, specifically based on the needs of older adult users of this technology; and 2) provide critical usability data to support an R03 proposal to test efficacy of the ATA in improving attention among community-dwelling elders as well as generalization to everyday functional abilities. We will purposively recruit older adults with different levels of mobile technology experience, ranging from individuals with their own iPad devices to those with no previous experience. A minimum of eight participants will be recruited from Centre County Senior Centers as well as weekly iPad training classes held at a local electronics store. Following screening and enrollment, a member of the study team will meet with each participant to facilitate testing of the ATA, take field notes pertinent to app and device usability, and to conduct a subsequent audio-recorded, semi-structured interview regarding use of the ATA. Additional measures will include a demographic questionnaire, brief cognitive screen, and a computer and technology experience questionnaire in order to contextualize user experiences with the ATA. Quantitative usability data will be captured as the app is accessed and used. Among participants with their own devices, the ATA will be installed using TestFlight, a mobile app beta testing service; all others will use a study-provided iPad. The entire visit will take approximately two hours, which includes time to complete the baseline study measures, ATA testing, and post-test interview. A trained individual external to the research team will be hired to transcribe audio recordings into verbatim transcripts. Open-ended data from participant interviews and field notes will be analyzed to uncover common themes related to feasibility of the ATA. Usability data collected during the use of the ATA will be analyzed and used to generate a data-based report which will guide revisions of ATA content and protocols.

Our interdisciplinary project team includes researchers with the necessary expertise to develop and test the ATA with older adults: iOS (iPhone and iPad) app development, including designing and developing apps using Apple's iOS Developer Tools and advanced programming techniques (Hannan); cognitive aging and development of cognitive assessments (Sliwinski); intervention implementation, including cognition-focused interventions, for older adults across settings in both clinical and research capacities (Hill, Lin); and, the measurement of attention using mobile devices and the association of attention and other cognitive processes (Mogle).

5. Relevance to SSRI's Mission: Our proposal addresses a problem among older adults that is significant at individual as well as societal levels: the decline of cognitive abilities characteristic of the aging process and the resultant negative impact on daily life. This project brings together a new interdisciplinary research team which is addressing this problem in a novel way through the use of mobile technology.

6. Use of SSRI Services: None for this pilot study.

7. Budget and Justification: Total requested: $4608, See attached budget, page 8.
Department budget coordinator: Amy Hasan, alh31@psu.edu, 865-9189
Budget and fund number: 04-015-28, 10010 cost center SSRI-Hanna
Administrative area number: 015

8. Timeline:
November – December 2013: ATA Development; November 2013: Submit IRB application for human participant research; January – March 2014: Recruitment, enrollment, and data collection; March – May 2014: Data analysis and report preparation; May – July 2014: Revisions to ATA completed based on study findings; August 2014: Submit manuscript and prepare R03 Proposal.

BIBLIOGRAPHY

"Mild Cognitive Impairment." *alz.org.* Alzheimer's Association, n.d. Web. 21 Mar. 2014.

"USA." *quickfacts.census.gov.* United States Census Bureau, 27 Mar. 2014. Web. 28 Mar. 2014

ACADEMIC VITA

Robert Dick
21 Park Drive
Cheswick, PA 15024
rgd5032@psu.edu

Education:

The Pennsylvania State University Schreyer Honors College, University Park, PA
Bachelor of Science in Computer Engineering, minor in Spanish
Graduation: Spring 2014
Universidad de Sevilla, Seville, Spain (Spring 2012)
CIEE Language and Society program
Deer Lakes High School, Russellton, PA
Graduated Valedictorian June 11, 2010

Professional Experience:

Microsoft Corp. – Software Developer Engineer Intern: OneNote (May – August 2013)
Developed an interactive web page using C#, JavaScript, and HTML
Aerotech Inc. – Electrical Engineering Intern: Graphical Design Team (June – August 2012)
Developed and tested graphical user interfaces and APIs in C# under the .NET framework for multi-axis motion controller software applications
Zoll Medical Corporation – Engineering Intern (May 2011 – August 2011)
Debugged and developed C# code, wrote UNIX shell scripts, analyzed data, and wrote design test reports for an embedded system, medical defibrillation device

Honors and Awards:

The Lockheed Martin Corporation Scholarship recipient
The Evan Pugh Scholar Award
The President Sparks Award
The President’s Freshman Award
Dean’s List in all semesters

Association Memberships & Activities:

Apollo FTK – Independent THON Organization
Penn State Floor Hockey Club
Schreyer Scholars Orientation Group Leader 2011
Phi Eta Sigma National Honor Society