Executive Summary

The following document is a proposal for the design of a game to improve speed and accuracy of memory recall, sponsored by Ohio University. First, the background of the client and the purpose behind the design will be presented, as well as any products or patents that exist which describe similar software to the one being designed. Their prices and shortcomings will also be discussed. Then a methods section will follow which will describe ideas and goals for the implementation of the software. This is just an overview of what to expect, but many details are liable to change as the design process continues. Finally, the proposal will close with a budget of expected costs for the design of the product, with some comparisons with other similar products, and a summary of the main points in the conclusion.

1. Introduction
1.1 Background (Client and Disability)

Many of the people who will make use of this software game have Alzheimer’s Disease or dementia and suffer from severe memory loss induced by the disease. Others do not have Alzheimer’s or dementia but struggle with short-term memory loss due to advanced age. For those who have Alzheimer’s, the cause of the onset of the disease is unclear. There are, however, a couple of main hypotheses which describe possible causes for the damage inflicted on brain cells in patients with the disease. One possible explanation is the deposition of beta-amyloid plaque, which interferes with communication between brain cells. Another possibility is the entanglement of an essential neuronal protein called tau. In people with Alzheimer’s this protein undergoes alterations and does not function properly, leading to damaged neurons. ¹

1.2 Purpose

Ohio University’s project to develop a software game to improve memory recall is intended to aid elderly patients with Alzheimer’s Disease or dementia, or those who suffer from short-term memory loss due to advanced age, by improving their ability to correctly recall the names of friends and loved ones. Patients who cannot remember their family’s and friends’ names often suffer frustration and embarrassment. Therefore, a speech pathologist at the university suggested that a way to improve memory recall is to repeatedly practice remembering names with pictures of important people in the patients’ lives. Therefore, the university’s request is for software to be designed which will act as a game for patients to play with pictures of people/places/things they want to remember, which will help to improve name recollection.

There have also been studies done, such as one entitled “Improvement of Picture Recall by Repetition in Patients with Dementia of Alzheimer Type” in the International Journal of Geriatric Psychiatry, volume 12, which show that the use of imagery that is repeatedly shown to

¹ Mayo Clinic at http://www.mayoclinic.com/health/alzheimers-disease/DS00161/DSECTION=causes
patients suffering from memory loss, while simultaneously requiring them to identify the image, is helpful in improving memory recall. Therefore, inspiration for this software game originates from this evidence.

1.3 Previous Work Done By Others

1.3.1 Products

There are several software memory recall products that are available on the market; however, they are not useful in fulfilling all the specifications determined by the university for a memory recall game. A game called Memory Magic at www.acceleratedlearningmethods.com includes several types of flashing memory games in which pictures are flashed on the screen and the user must either identify them by typing the name in or remember the order of certain images. This is done repeatedly to increase memory recall and memory capacity. The game sells for about $50 and includes several other types of memory games besides the one just described; however, it does not allow for customization of images or voice input. Also, it, like many other memory recall games on the market, is geared toward children and teenagers. Therefore, it will not suffice for the client’s usage.

Another example of memory recall software programs that are available on the market are SharperBrain and SharperMemory by SharperPrograms.\(^2\) They basically have the same functionality as the above described game, but it is geared toward children and adults, with the aim of treating ADD and ADHD. The two games sold together are priced at $740, and have the same shortcomings as the product described above.

1.3.2 Patent Search

Unlike marketed products, there were no patented memory recall softwares found; however, there were two related patents which describe systems or devices to aid in memory recall for people suffering from Alzheimer’s Disease and other neurological disabilities using images. US Patent 2008/01387783 A1, dated June 12, 2008, and filed by Microsoft Corporation for a system to implement Memory Training Via Visual Journal, describes a method which could improve a person’s cognitive ability by recording certain snapshots of events during the day and then playing them back to the person to enhance his/her recollection of the day’s events. The events can be recorded from the person’s perspective or from a third person’s perspective (e.g., by placing the recording device on top of a dashboard). Image capture is triggered by using physiological or environmental sensors to record stimuli associated, for example, with blood pressure, mood, body temperature, external temperature, location, motion, lighting as well as time-based mechanisms. The system would be comprised of a recording component and also a memory recall component which would play back the “memories” for the person who experienced the events. Figure 1 below is an image of the setup.

\(^2\) SharperPrograms: Break Free From ADHD at http://www.sharperprograms.com/
US Patent 2005/0203430 A1, dated September 15, 2005, and also filed by Microsoft Corporation for a Recall Device, is very similar to the device discussed above. The same components are involved, as well as the same type of triggering for image capture. The difference is that the physiological or environmental stimuli must be followed by a relatively stable period, measured by an accelerometer, before the image is captured. In that way, clearer images may be captured. An image of the device may be seen in Figure 2 below.

Figure 1: US Patent 2008/01387783 A1 system setup\(^3\)

Figure 2: US Patent 2005/0203430 A1 device\(^4\)

---

\(^3\) Free Patents Online : All the Inventions of Mankind at http://www.freepatentsonline.com/y2008/0138783.html

\(^4\) Free Patents Online : All the Inventions of Mankind at http://www.freepatentsonline.com/20050203430.pdf
While these patents support the notion of using images to enhance memory recall for people with memory impairments, they still do not meet the requirements of the given project, as they are not software games and they do not necessarily exercise the user’s ability to remember names of people through audio input.

2. Project Description

2.1 Objective

There are several requirements that the sponsor requires the name game software to fulfill. First, it must be installed on a PC. The patients use mostly Microsoft Windows with Vista and so the game must be installable on this operating system. The game must accept digital images and names, specific for each user and must also supply a package of images of celebrities and political figures, relevant to elderly people. Also, the user or care-taker must be able to select different lists of names to be worked on in different sets, the game must prompt the user to say a name into a microphone once the image is presented, the game must keep score and time, and it must respond to correct or incorrect answers with auditory and visual feedback. Finally, the software used for voice recognition must be supplied, along with the microphone.

Other requirements that the sponsor has have to do with ease of use for people who have few computer skills and are cognitively impaired. The user interface must be simple and clear. Any written instructions must be made relatively large for users with visual impairments, and any audio feedback must be loud enough for people with hearing loss.

Implementation of this software game will probably involve Java, Labview, and maybe both, depending on how the team finally decides to integrate the voice recognition component into the code. Visual Studio and languages like C# are also a possibility. Java is desirable because of its many libraries which give the ability to code for excellent graphical user interfaces (GUI), which would meet the specification for a simple, intuitive, easy-to-understand interface for users who are cognitively impaired or have few computer skills. Labview is useful in that it, too, can create intuitive user interfaces, and it has certain subvis which allow the programmer to work with audio and voice. Another advantage of Labview is that it is a very high level programming language, with drag and drop functions. C# is very similar to Java, and Visual Studio is a powerful tool for creating enterprise software and also includes drag and drop functions.

The actual details of the code for the game will not be presented here but an overall functioning of the program will be described. This is just a proposal so details are liable to change, but this should provide a good overview of what to expect.
2.2 Methods

Regardless of the type of language used to code the game, there are five main components, which must be written into the code that are essential to meeting all the specifications. They are the following and will be discussed in more detail in the subsequent paragraphs:

1) Code to create the graphical user interface—involves buttons, selection lists, text and image fields
2) Code to create lists of personal photos from images saved on the computer, which may be ordered by the user and stored for use in the future
3) Code to select a picture list to be used in the current game
4) Code to interface the voice recognition software and microphone with the rest of the game
5) Code to convert the software to an installable game on the users’ computers (e.g., .exe file)

Figure 3 below is an image which depicts what the team anticipates the GUI will look like once it is complete. The fields labeled “score” and “timer” are text fields which will continually update throughout the game, based on the answer that is received from the user through the microphone, as he/she progresses through each image in his/her tailored list of personal pictures. If the user identifies the picture incorrectly, the score will remain the same, the image will change to the next picture in the list, and the clock will continue counting until the user has reached the end of the game. Once the game is complete, another text field, which is not displayed in Figure 3, accompanied with audio feedback, will be visible, which will display game statistics, such as total score, total time, percentage correct, and which images were incorrectly identified, along with the correct identification. Text will also appear throughout the game to prompt the user to say the name of the person in the image. Both Java and Labview have timers and text fields, available in their libraries, which, with some modification, will work perfectly for the uses described above.
The buttons in Figure 1, labeled “Play Game”, “Pause Game”, “Create Picture List”, and “Select List” will be implemented using either push buttons or radio buttons, both of which are available in the Java libraries, and the former of which is available in Labview, as well. “Play Game” will either start the game from the beginning by calling the voice recognition software to perform and initiating the display of the images and the timer and score count, or it will resume the game from where the user has left off after pressing pause. The particular functions of the “Create Picture List” and “Select List” buttons will be discussed later. In Java, these buttons’ functions can be executed by creating appropriate listener classes to associate the object of the action with the button—the initiator of the action. In Labview, buttons and their functions are associated simply by connecting appropriate wires in the block diagram display. It should also be noted that certain buttons will not perform their particular function if other functions have not yet been performed. For instance, the user will not be able to press “Play Game” and have the game function, if he/she has not yet created and/or selected a list of pictures to be used. In these cases, a text message will appear that will indicate the error to the user and explain how to correct it.
Goal two calls for code which allows the user to create personal, ordered lists of photos to be stored for current and future use in the game. This function, as mentioned earlier, is associated with a button on the GUI. The software program, itself, will not supply a method for uploading and storing the images. Rather it will utilize the computer's own hard drive in order to store and work with the images. The user will either scan in the pictures or transfer them to the computer from a digital camera via a USB cord, using the programs associated with these devices. The user can then save the pictures to the directory of his/her choice. Then, the user will be able to press the “Create List Button” and another window will appear which will allow him/her to browse to the directory where the pictures are stored and select the images that should be placed in the list. Once these images are selected, the user will be presented with the list of pictures that he/she chose and will be prompted to order them according to the way they should be presented in the game. Once the images have been selected and ordered, their location paths will be stored by the program in a list-type data structure, such as a queue, and then the program will be prepared to present the images to the user.

As will be discussed in a little more detail below, the list creation will also involve providing the name of the person in the image (with perhaps a spoken pronunciation of the name into the microphone), so that not only does the program store the image but it also stores the person’s identification, along, perhaps with its spoken pronunciation for word matching once the game has begun. Also, within this process, the user will be presented with the option to save the list (and identification) for future use in a directory of his/her choice. Creating the list stores the images only temporarily for current use, so in order to use the same list in the future without re-creating it, the user will have to choose this save option. Also, creating a list does not require that that particular list be used in the current game. The list may be saved for another time and a different list may be chosen for use as described in the next paragraph.

Goal three involves the “Select List” button, which allows the user to use lists of images which have been previously created and saved. (The game, itself, will also come supplied with a default list of celebrities and important figures, which are especially relevant to people aged 60 and up. This list will also be a selection option). If the user has just created a list according to the process above, and wishes to use that list immediately, this button does not have to be pressed, since the program will automatically enter in that newly created list to be used. However, if he/she creates a list, saves it, and then wishes to use a different list than the one just created, pressing this “Select List” button and choosing another list will over-write the temporarily stored image location paths with the paths from the newly chose list. Once the user clicks on “Select List”, he/she will once again be presented with a window that will allow him/her to browse through the files to the directory where he/she stored the lists and select one with which to play the game. Then the image location paths will be loaded into the game and used to play, as described above.
Goal four presents probably the biggest challenge of the project. That is, interfacing voice recognition software with the game, itself. Vista will be used on most of the users’ computers, which has its own built-in voice recognition software which allows the user to speak what should be typed on the screen, rather than actually typing it. This is very helpful for patients with both physical and cognitive disabilities. There is the option to use Vista’s voice software, but it may be more beneficial to supply an external voice program in case a different operating system is ever used. That way, the game won’t become unusable if this change should happen. The voice recognition software will just have to be reinstalled.

Regardless of what software is used, the challenge of integrating both the game and the voice recognition software remains basically the same. In general, speech recognition works by taking in to a computer an analogue sound signal through a microphone, converting it to a digital signal which the computer can use, breaking down the word into phonemes, or small word segments, and then comparing these phonemes to a library of words to decide which word has been pronounced. There are several obstacles to accuracy in voice recognition programs, including varying pronunciations of words due to dialect or accents in different people, limited vocabulary libraries, and low signal-to-noise ratio, which requires distinct pronunciation of words and cannot tolerate a lot of background noise. Since the user of the game will be pronouncing proper nouns into the microphone, depending on how common the name is, it may present difficulties in speech recognition of the name. There aren’t any voice programs which will overcome all of these obstacles, even in the newest and most sophisticated programs; however, there are programs which do better jobs than others at recognition.

One of the best voice recognition softwares available is Nuance’s Dragon Naturally Speaking, which may be seen in Figure 4 below. It is one of the best general purpose, large vocabulary, continuous speech recognition products available, with over 95% accuracy. This is the product that the team is inclined toward using for this project. It allows for dictating text into most Microsoft applications and on the internet, as well as allowing for the control of the mouse and many desktop applications. It also has custom vocabulary features, which allow the user to add words, which he/she commonly uses, to the pre-existing libraries. This feature, especially, may be useful in exploiting for accurate recognition of proper nouns.
The code to integrate the voice recognition with the game to be designed can be explained, at this point, in a general sense. Once the voice software has been installed, the code for the game will have to have a method for calling the executable file for the voice software in order to begin its use for the game. This function will probably be included in the “Start Game” button’s responsibilities, as described above. The manufacturer’s specifications for Dragon Naturally Speaking, as well as many other voice recognition softwares, indicate that the program will work for most Microsoft and Internet applications, such as Word, Excel, Internet Explorer, and Mozilla Firefox; however, its functionality outside of these specific environments is not guaranteed. Therefore, the game’s code will have to find a method for “forcing” the voice software to function within the game environment, perhaps by creating document or text files in Word or Notepad into which the names are dictated by the voice software and then read by the game and compared with the name that was given to the image during the creation of the image list to determine if the user is correct. Also, perhaps as part of the responsibility of the “Create List” button, proper nouns can be added into the vocabulary library of the voice software for better word recognition.

The final goal of this project is to make the completed game installable on a computer operating with Microsoft Windows. In order to accomplish this, an installer, created by a Wizard, will copy the program executable and all of its dependents to the program files directory. Then the installation package will be transferred to the user’s computer on a CD-ROM. Shortcuts will be copied to the start menu and desktop. With this done, the game should be ready to be played.

3. Budget

The total cost for making this game is estimated to be at $360. Considering the prices and the capabilities of the already existing memory game products available, which were discussed above, and similar products which fall mostly in the same price range and have the same functionality, this is a reasonable price. No products were found which allowed for customizable lists of images and voice input from the user, and some of these products, such as SharperBrain and SharperMemory are priced at almost $740. Most high-end memory recall
games, especially those which attempt to treat cognitive disorders, are priced in this range. The marketed price for the proposed project, at 35% of the prototype costs, is $126. Therefore, the product proposed offers exactly what the client requests for less cost, compared to many other software games. For those games, such as Memory Magic, which are priced less than what is proposed in this budget, it must again be taken into consideration that those games do not fulfill the specifications, nor do they require the purchase and installation of additional, sophisticated software, such as the voice recognition programs discussed. Money will also be saved in that all of the software to be used to write the code for the game is either already available to the team or downloadable as freeware. The priciest item on the list is a high-end microphone, which cancels out noise and is especially used for Nuance voice recognition technology. Considering that the user will only be speaking proper nouns into the microphone, which could cause difficulty in recognition, and also that people with Alzheimer’s Disease can sometimes have speech difficulties, it’s important to the proper functioning of the game that the best microphone possible be used to pick up sounds.
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Approximate Cost (US Dollars)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nuance Dragon Naturally Speaking Voice Recognition Software</td>
<td>Large vocabulary including proper nouns\nCustomizable vocabulary libraries\nAble to deal with accents and dialects</td>
<td>$110</td>
</tr>
<tr>
<td>SpeechWare microphone (or other brand especially designed for use with voice recognition software)</td>
<td>Top rated for use with Nuance voice recognition technology\nSuperior digital sound quality for voice recording &amp; speech recognition\nNoise canceling technology with manual sensitivity control, filters out background noise\nFar away positioning</td>
<td>$250</td>
</tr>
<tr>
<td>Total approximate cost of product</td>
<td></td>
<td>$360</td>
</tr>
<tr>
<td>Total approximate cost of product including testing equipment (1 Nuance product and 1 microphone)</td>
<td></td>
<td>$720</td>
</tr>
<tr>
<td>Price of product at 35% of prototype value</td>
<td></td>
<td>$126</td>
</tr>
</tbody>
</table>

Table 1: Approximate budget for project

4. Conclusion
In conclusion, the software game to be designed meets all of the client’s specifications, including customizable image lists and voice input with a microphone. These latter two features distinguish this product from other products on the market. Compared to most higher-end memory recall software games, which still don’t meet all the client’s requirements, the proposed design is more cost-efficient. In terms of implementation of the software, Java, Labview, or some other Visual Studio languages, will most likely be used to implement a simple, intuitive GUI and to integrate the voice recognition software and microphone with the game.