**Brain Games: An Improved Version of Dual N-Back**

**Tianmu Zhi**

OMSCS, Georgia Institute of

Technology

tzhi3@gatech.edu

**Abstract –** Brain Training and Brain Games have grown to be a multi-billion dollar industry in the last decade, and it all began with the Dual N-Back game that the Jaeggi study purported as an exercise that could increase anybody’s IQ. [1] This paper describes a modified version of the original Dual N-Back game. The game attempts to prevent the natural effect where the player grows accustomed to the particular game format which lessens the workout the game gives to the working memory.

**Index Terms -** Brain Training, N-Back, Brain Games.

**1 INTRODUCTION**

Brain training games has become all the rage in the past decade. Many firms such as Lumosity have claimed that by training specific functions of your brain, most notably the fluid intelligence/ memory, the effects will carry over in other cognitive functions. While years of psychological studies show that practice at tasks only make you better at those specific tasks, recent research has finally started to buck that trend. The seminal study that really put Brain Training on the map was a 2008 study by Jaeggi and Buschkuehl, two psychologists at the University of Maryland. They study posited that using a fairly simple game called N Back, subjects could improve their working memory which was highly correlated to IQ. In this paper, an augmented version of the Dual N Back game is proposed that dynamically adjusts the game grid so that the player doesn’t get used to the game and continues to tap his working memory in playing the game.

**2 MOTIVATION**

The idea of Brain Training has been wrought with controversy ever since the

idea became popular. Most notably, Lumosity, perhaps the singular platform that made Brain Training popular, was sued for 2 million dollars for its claims of protecting against cognitive decline.

For every paper advocating the potential of brain training, there is another one claiming that there is no scientific evidence for it.

To add some nuance to it, there are established positive results for Brain Training for older adults as well as those with mental maladies. The most controversial aspect of Brain Training research is whether it is effective on healthy youth. Naysayers posit that brain training only allows the subject to get better at brain training games and the results doesn’t apply to general cognitive activities. For example, there is the counter-study by Adrian Owen where he tested 11,430 children with brain training software over a 6 week period and did not discover any evidence of significant transfer effects.

**3 N-BACK MECHANICS**

First, let’s describe in detail the mechanics of the original N-Back game. The original game goes like this: the subject is given a sequence of letters shown in order. When the letter matches what was shown ‘n‘ steps before, the subject then presses a button. The number ‘n’ will increase, meaning that you need to remember more and more consecutive items as the game progresses in order to match the current item with the item ‘n’ sequences ago. For N = 2 then, the subject must press the button at the end of the below sequence:

L V S S T U N N V P O Q A T L M L

This game challenges the ‘active’ part of the working memory and forces the subject to constantly update a memory ‘buffer’ to compare new input against what you remember. This requires the maintenance and manipulation of the working memory.

The ‘Dual N-Back’ or ‘Dual Task N-Back’ version of the game makes things a little more difficult by introducing an auditory input as well as a visual one. Brain Workshop implements version in their game. Different boxes in a square grid will light up and a letter be sound every time a box lights up. The subject must remember the sequence of which boxes light up as well as the letter associated with the boxes. Another version of the Dual N-Back game will often get you to match items not only based on their position, but also on their color. So you might see letters appearing, but also changing color each time. Now you must press stop if the items match either in sound or in color. Another version exists by BrainHQ which comes in the form of playing cards. The subject must remember the sequence of cards as they are being dealt by the dealer. Difficulty increases as more cards are shown in the sequence and each card comes with more detail, i.e. one must remember not only the card number but the suit.

**4 PREMISE FOR GAME DESIGN**

**4.1 Weaknesses of the Current Studies**

One of the weaknesses in the studies done on Brain Training is that it’s hard to define a control as well as define a test to determine the pass through effects of Brain Training exercises to more general cognitive tasks. A major flaw of many studies was that the control group was the group that didn’t undergo any training, or any activity of any kind for that matter. [4] The difficulty with this is that it makes the study more susceptible to the Placebo effect. Another issue is that the appropriate task determining improvement in general cognitive ability is difficult to pin down and in my opinion, is the true crux of the argument. From what I’ve gleaned from reading numerous studies, the issue is philosophical in nature and should be a subject of its own study. While studies and reviews of studies have strongly indicated that Brain Training has little effect on one’s IQ as determined by standard IQ tests. However, much like how I highlighted in the previous assignment, this doesn’t necessarily mean that Brain Training has no positive effects on some aspect of cognitive aptitude.

**4.2 Why Brain Training Games May Still Be Useful**

For example, if Brain Training didn’t improve IQ but did improve one’s ability to drive safely, remember acquaintance’s names, as well as recall phone numbers, this might still qualify as improvement to one’s mental faculties

While Brain Training hasn’t conclusively shown that it can improve general cognitive abilities better than ordinary games, it does indeed improve general cognition just like activities such as Chess or Go. 1 This isn’t a very far-fetched concept, since the brain gets better the more you use it. 2

In fact, from a metacognitive perspective, the central idea of Brain Training is rather simple and almost self-evident. For example, a study of London taxi-drivers revealed that their hippocampi centers bulged from their years of navigating the labyrinthine London roads. Their spatial memory were more advanced than non-taxi drivers. 6 Having an advanced spatial memory is useful in many tasks beyond driving a cab, namely interior design as well as architecture. The same study however found that the taxi drivers performed below average in other types of memory tests, implying that the amount of time they spent honing their spatial memory came at the expense of training other types of memory. This, of course is nothing new, as it’s merely stating that the more one practices a particular task, the better he becomes at it. The core tenets of Brain Training attempts to overcome this by attempting to target specific core mental skills that are used in many tasks. This way, time spent in training at this skill would augment many types of cognitive abilities and wouldn’t mean one cognitive faculty is developed at the expense of another. However, this also means that not any particular mental cognitive ability is augmented significantly. The usual target for Brain Training games is improving working memory. Working memory is defined as one’s ability to juggle disparate concepts in one’s thinking and is analogous to a computer’s RAM. Additionally, working memory is exercised and required for much of the subject matter in education but is rarely specifically targeted for training. While the evidence is at best inconclusive for this thesis currently, the general idea is certainly self-evident. For example, it shouldn’t surprise anyone that sharpening one’s arithmetic skill will enable him to better grasp differential calculus down the line. Having a strong grasp of arithmetic will also allow one to grasp many of the concepts in accounting as well as many other disciplines much easier. This is easy to understand because it’s obvious arithmetic is a fundamental building block to differential equations as well as accounting. The current fallacy in many of the studies in Brain Training is that they are testing whether Brain Training is a panacea for general IQ and the evidence is pretty clear that it’s not. However, as research in this area gets more refined and metacognition becomes more well understood, I believe the studies will get more targeted and the tests more specific.

**4.2 Addressing N Back’s Main Weakness**

I’ve revamped the Dual Back Game to correct researcher’s the main concern with it, namely that users end up getting better at the game and gradually have to strain their working memory less and less. They often can utilize memory techniques such as chunking or mind palaces. I’ve remade that the Dual N-Back game so that it is dynamic in gameplay, requiring the user to adapt to different grids and different inputs the player has to remember.

**2.2 Development**

The game was developed in the Python language and packaged as an executable.

Development began with the creation of the wireframes and later translated into screens using the PyCharm IDE environment. The code borrows techniques from Erik Njare and his open source version of the N-Back game on Github. []

The game results are stored in a CSV file.

**2.2 Installation/Setup**

The Python code that drives the game is version 2.7 and the main driver of the game play is the PyGame module. The user needs to download the module at <http://www.pygame.org/download.shtml> and install the [pygame-1.9.1.win32-py2.7.msi](http://pygame.org/ftp/pygame-1.9.1.win32-py2.7.msi) if the Python the user is using is on Win32. Many might be on the 64 bit version of Python and need to follow the instructions at <https://www.webucator.com/blog/2015/03/installing-the-windows-64-bit-version-of-pygame/>.

**2.3 User Experience**

The application loads with the Main Menu. The player is informed that the Space Bar is the trigger button for when the player sees a box appearing in the location N steps ago and the “A” key is the analogous trigger for sound. There are three grids

The problem with these games is that playing the game for extended period of time, subjects will start learning heuristics that aid in progressing in the game but won’t actually increase the working memory of the subject. For example, one simple aid is speaking out loud the letters of the dual N back game which gives the memory another crutch to lean on to recall the sequences. The key to making the game more robust is giving it enough variation so the subject is never able to develop these heuristics. Not that much variation is needed to achieve this. I plan to make 3 variations of the basic Dual N-Back game that cycles through at some frequency, giving the subject just enough consistency where he can get comfortable with the game but not enough where he can develop memory tricks. Along with the standard form, I will create one version where the boxes are in an X shaped grid instead of a square grid. Also, instead of hearing a letter associated with a flashing box, the box will display a number and the player will have to remember both the location and the number. This interface focuses on visual memory instead of testing both visual and auditory memory. Another interface will have circles that rotate at a set speed and the player sees circles light up and hears an associated letter. With all these different variation of the Dual N-Back game, each version will repeat if the user passes the game, allowing him to progress to a higher “N” level. If he fails, another of the three possible interfaces will launch and the game reinitiate with the same N level.

Functional Requirements:

1. The basic functionalities of the standard Dual N Back game will have to be built first. The game will enable the user to play a round of the Dual N Back game in a standard square grid interface.

2. The game will progress and increase in “N”, i.e. the number of objects if the player scores at least an 80% on the game.

3. The game will switch to a different interface if the player fails to score at least an 80%. This will curtail the player’s ability increase their score in the game by merely getting better at the particular form of the game. The new interface will be selected randomly from the 3 possible interfaces I will code.

4. The player can choose to select a particular “N” level if they want to challenge themselves with a harder level. They will not be able to select a particular interface because it would allow them to learn a particular interface.

5. The game will keep a log of the player’s performance in a text file.

The default initial grid is the standard square grid with both positioning and sound input, requiring the player to remember both the position of the squares and the audio numbers utter at ever square.

Each grid will play for at least 6 slides. By slide, I mean one new display of a box. After the third slide, the game will keep a score of the win percentage. A win here is either a correct trigger of a repeat square from N steps ago, or a no trigger of a square that is not the positon of the square N steps ago. If the player’s accuracy goes above 75%, the game will change grids. The player has to keep a minimum 80% accuracy rate to advance to the next N level. The N level is kept in a .txt file in the game folder that is incremented if the game ends with a player maintaining an 80% accuracy. A player can override this by editing the .txt file directly.

**2.4 Unit Tests**

I enlisted the help of my wife who is a Director of an Elementary School perform the Unit Testing for the game. For each Unit Test, I set up the conditions as the test prescribed and was told to recreate certain game conditions (such as achieving a particular score) to test a desired outcome. My instructions to her did not reveal what the desired outcome would be.

* N = 1

Slides = 8

Check the grid switches after the 4th slide if the player achieves 75% accuracy for the first 4 slides.

* N = 1

Slides = 5

Check N goes up to 2 in the next game if 100% accuracy is achieved in the first game.

* N = 1

Slides = 8

Grid Switches = 1

Check that final accuracy of the game is maintained between the 2 grids after 1 grid switch.

* N = 1

Slides = 8

Grid Switches = 1

Check that triggering any key but ‘A’ or Spacebar will do nothing.

3)

N = 1

Slides = 8

Grid Switches = 1

Check that triggering the ‘A’ key will do nothing in the second grid since audio input is no longer needed.

1)

N = 2

Slides = 8

Check the grid switches after the 4th slide if the player achieves 75% accuracy for the first 4 slides.

2)

N = 2

Slides = 5

Check N goes up to 2 in the next game if 100% accuracy is achieved in the first game.

3)

N = 2

Slides = 8

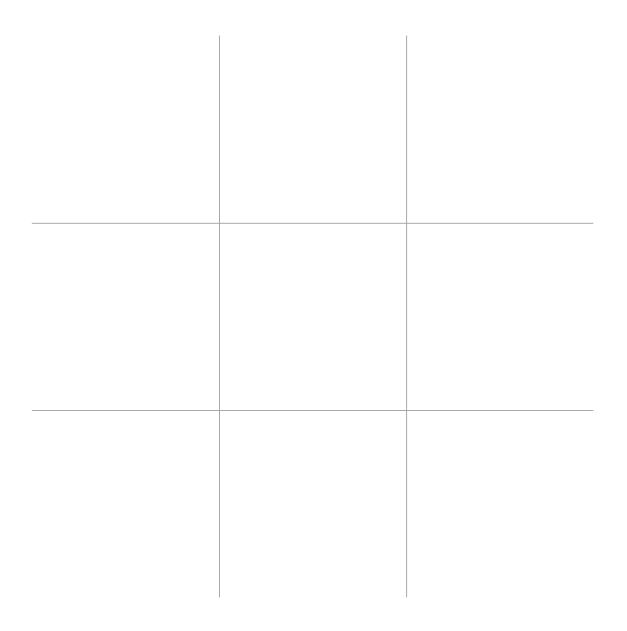
Grid Switches = 1

Check that final accuracy of the game is maintained between the 2 grids after 1 grid switch.

**2.4 Functional Enhancements**

The game remained in Python code format but should be expanded to be a web app so that it can be used more widely and possibly in a commercial setting. Heroku is a Cloud based application platform and possibly the most convenient way to do this.

Additionally, more dynamic grids should be added to challenge the player’s working memory. Initially, I envisioned grids where the boxes moved. While that’s not featured as a grid here, I’ve laid out the groundwork for it. Designs of new grids can be easily added on the current framework.



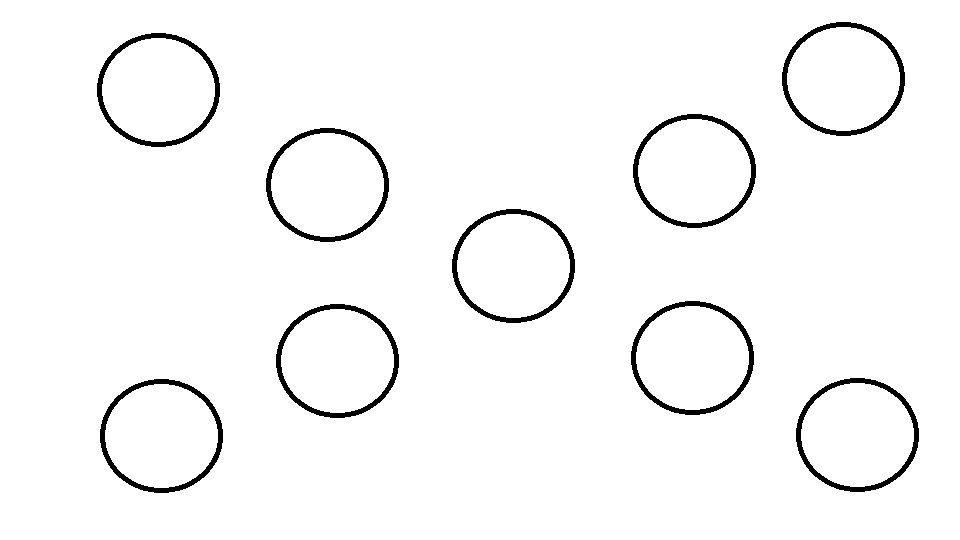


Fig. 3. Grid 3

Fig. 2. Grid 1

The Grid has an audio input as well. The player must trigger the space key if the position of the box matches the N back box as must also trigger the ‘A’ key if the sound heard matches the N back sound.

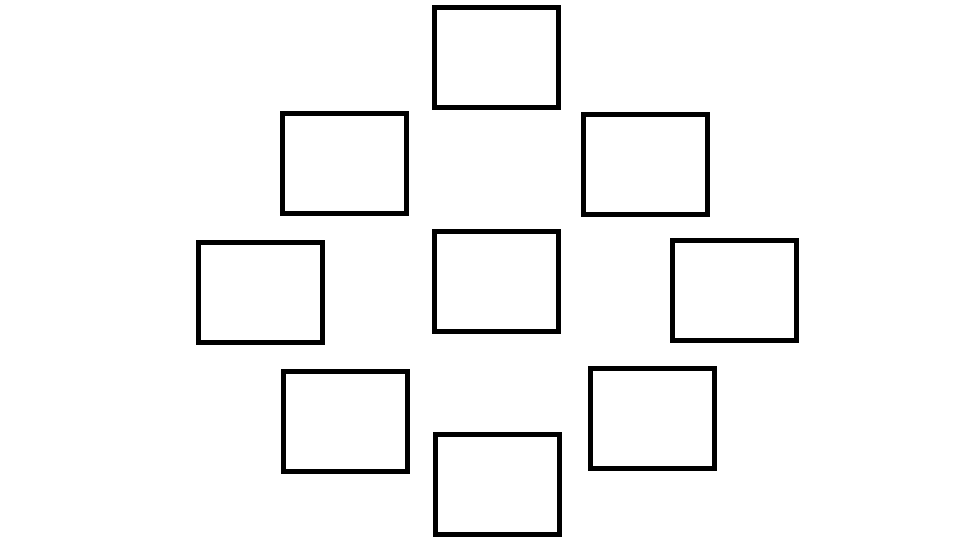


Fig. 2. Grid 2

**3 CONCLUSION**

The motivation behind the development of this application was to provide accurate and reliable information to high school girls about Computer Science through the use of mobile technology.

Science Education can also be built in the application, with prizes attached, to provide motivation for learning.

**ACKNOWLEDGEMENTS**

The author wishes to thank Dr. David Joyner, TAs and my mentor Gregory Weber, for their continued support. I also want thank all the students for their valuable peer feedback.

**REFERENCES**

1. Scragg, Greg and Jesse Smith. 1998. “A Study of Barriers to Women in Undergraduate

Computer Science.” SIGCSE 98. Atlanta, GA.

1. CISE Pathways to Revitalize Undergraduate Computing Education (CPATH). Request For Proposal, National Science Foundation, Directorate for Computer and Information Science and Engineering, 2006.
2. Trauth, E.M. & Howcroft, D. (2006). Critical Empirical Research in IS: An Example of Gender and IT. Information Technology and People. Special Issue on Critical Research in Information Systems
3. Trauth, E.M. & Howcroft, D. (2006). Critical Empirical Research in IS: An Example of Gender and IT. Information Technology and People. Special Issue on Critical Research in Information Systems
4. Jenson, J. & De Castell, S. (2010). Gender, Simulation, and Gaming: Research Review and Redirections. Simulation & Gaming 41, 1 (February 2010), 51-71.
5. Terlecki, M., Harner-Steciw, L, Irvin-Hannum, J., Marchetto-Ryan, N., Ruhl, L., Wiggins, (2011). Sex Differences and Similarities in Video Game Experience, Preferences, and Self-Efficacy: Implications for the Gaming Industry. Current Psychology, 30(1), 22-33.
6. Farenga, S. J., and Joyce, B. A. 1999. Intentions of Young Students to Enroll in Science Courses in the Future: An Examination of Gender Differences. Science Education, 83(1): 55-75.
7. Ellen Spertus. 1991. Why are there so few female computer scientists? Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA.
8. Ernest, John. "Mathematics and Sex." The

American Mathematics Monthly, October 1976, 83:595-615. A good survey of the factors discouraging females from mathematics with an excellent bibliography