RocketMouse:
A Game for Assessing ADHD

Hesham H. Salman Jonathon Kissinger Sean Mead Troy Johnson

October 29, 2014

1 Abstract

Studies have recently shown that attention deficit disorders may affect upwards of 5% of the population. In recent years, mobile games have also become extremely popular due to the increasing ubiquity of smart phones. By exploiting this trend of mobile game popularity, we hope to create new tools to measure the symptoms of ADHD in the form of a mobile game. In this paper, we explore the results of creating a mobile game application that is capable of testing response time, distractibility, and mobile application preference in users with and without ADHD. We gather basic information about the user, such as whether or not they have ADHD, through a survey the user is asked to fill out at the beginning of the game. We predict that users who have been diagnosed with ADHD type disorders will have a higher variability in reaction times, higher distractibility, and will also prefer mobile applications that do not focus on their reaction times as a measure of success. Previous studies have indicated that the inhibitory abilities of children with ADHD are normalized when playing video games. Due to this, our mobile game will need to alternate between levels that are more like a mobile game and more like a test. We also predict that children diagnosed with ADHD will have more mobile applications installed on their mobile devices as their short attention spans may cause them to jump from mobile application to mobile application quickly as they may get bored rather quickly, compared to a person without ADHD.

2 Introduction

With the recent explosion in the ubiquity of mobile devices and the mobile games that often accompany these mobile devices, a great opportunity is presented to researchers for reaching a large scale, geographically widespread audience through mobile applications, especially for those limited by their local geographic location. Studies show attention deficit disorders affects upwards of 5% of the population and unfortunately most users have to make a trip to the doctor's office to have their symptoms measured and have a diagnosis made on whether or not they may have ADHD.

We propose the development of a mobile application that can help collect data about users to help quantify the response time characteristics that may often accompany a person with ADHD. The mobile application would first survey the user about things such as their age, gender, ADHD diagnosis, if they're medicated for ADHD, and other basic questions. Once the user is done filling out the survey, the user is allowed the play the game. The game will record the response time characteristics of the user as they are playing and upload the response time data to a remote server once the user is done playing. From there, data mining is performed on the user's response time data to see if any patterns can be established. By mining the user's data, we hope to see if there exists any strong patterns in the response times that stick out between users with ADHD and users without ADHD. The game should also be fun and engaging to keep the user interested and playing and hopefully we can collect as much data on their response time characteristics as possible.

Our game implementation will be a two-dimensional side scrolling game in which the user controls a character that moves from left to right across the screen. As the user moves from left to right across the screen, they must jump over obstacles that appear very quickly, this is where the recording of response times comes into play. The time it takes for the user to tap the screen to make their character jump to the time the object first appears on screen is the time that we will use as a measure for user's reaction time. Once the user is done, their response time data will be uploaded to the server for further analysis. We predict that users with ADHD type disorders will have higher variability in these reaction times, higher distractibility, and will prefer mobile applications that do not focus on reaction time as a measure of success.

3 Background Studies

There have been many studies done recently that revolve around studying the connections between reaction time and ADHD such as in [1] and [2]. However, we have been unable to find any mobile applications that have been developed for the sole purpose of collecting data to help with ADHD diagnosis. In [3] the authors developed an adaptive intelligent game that to help diagnose ADHD with relative accuracy, but the game wasn't a mobile application. Many studies have indicated that there is in fact a link between the reaction time of people with and without ADHD. Below, we will discuss the results of some of the studies that have been performed on reaction time and ADHD diagnosis and their implications.

The authors in [4] created a computer game called Supermarket game that essentially made the game a test for the diagnosis of ADHD.Data mining techniques were used for determining whether or not a user had ADHD. Overall, the authors found there game was effective at identifying children who were classified as ADHD positive, but relatively weak at identifying ADHD disorder subtypes. It is worth noting that this game also was not a game that was developed for mobile devices.

In [5] the authors conducted a study on 151 children participants. Out of the 151 children, 104 had previously been diagnosed with ADHD, the other 47 were diagnosed as not having any form of ADHD. The children performed a variety of tasks, but the task that displayed the biggest difference between the two sets of children was the Go/No Go task. The Go/No-Go test is a visual reaction test. The p value was less than 0.01 with the null hypotheses being that there was no difference between the two groups of children. So, this test was pretty strong evidence that there existed a difference between reaction times for the ADHD diagnosed children and non-ADHD diagnosed children.

In [6] the authors showed that there appears to exist a relationship between the variability in reaction times and children diagnosed with ADHD. The study consisted of 144 participants, 60 of which weren't diagnosed with ADHD, and 84 of which had previously been diagnosed with ADHD. The participants all participated in four reaction time tasks in which they had incentive to do well because they earned prizes if they did well. Overall, the study concluded that ADHD was associated with slower reaction time and also a higher variability in reaction times, similarly to the results in [5].

CogCubed is a game that was developed for attempting to diagnose people with ADHD [7]. However, CogCubed required a patient to go into a doctor's office to use, so the user couldn't have access to the tool outside of a doctor's office. This is where our research contribution could be really strong. Having a mobile application that is fun could lead to the contribution of a lot of data that could be analyzed and mined, which could hopefully lead to stronger conclusions or maybe even new conclusions on the relationship between reaction times and ADHD diagnosis.

4 Methods

4.1 Planning of Game and Development Environment

We needed to come up with a game that was simple, but fun and engaging for the user, and was also capable of collecting data on reaction times from the user. We decided upon developing a 2d side scroller mobile game that requires the user to jump over objects that spontaneously pop up in front of the user on screen.

For the development of our mobile game, we decided to use the Unity game engine. With Unity we will be able to target mobile platforms (both iOS and Android), desktop platforms, and even the web environment, which will allow us to reach an even greater audience if our initial prototype proves to be successful. Unity allows us to easily create visually appealing environments and graphics. Unity also allows us to incorporate physics into our mobile game much easier with the use of libraries designed by Unity, so we believe that Unity we be an excellent choice and perfect fit for developing our game with.

4.2 Survey Development

We decided that we need to have a basic survey to gather information from the user to help give us more information about the user to help find correlations or patterns that may exist in the data. We designed a survey for gathering basic information about the user from the user. We don't ask for any information that could personally identify the user, rather we just ask some basic questions such as the ones listed below.

- What is your age?
- What is your gender?
- Have you ever been diagnosed with ADHD?
- If so, are you medicated for it?
- If not, do you think you may have ADHD?

4.3 Collecting Installed Application Information

We would also like to collection information on the other mobile applications that the user also has installed on their mobile device. Android provides native support for collecting this information, so we had to use native Android code in combination with Unity. By collecting this information, we hope to upload the application information to the server and see if there exists any potential relationships between the number of mobile applications the user has installed on their mobile device and their ADHD diagnosis. We would also like to see if their exists any relationships between the type of mobile applications that the user has installed on their phone and their ADHD diagnosis.

4.4 Collection of Game Assets

To gather images and other assets that were necessary for the creation of our game, we used images from open source locations. Open source locations let us use their images for free if they are for an educational purpose. Unity has a large set of open source assets that are available for our use, this greatly aids in our ability for easily create sceneries and other backgrounds since we don't have to worry about spending the time designing and creating all the images necessary for the mobile application.

4.5 Database Design

We will have one database and that database will be housed on the server. The database will consist of four tables; a reaction times table, a survey table, a user table, and an applications installed table. It is important to note that we don't store any identifying data about the user such as their name or anything else that could possibly trace the data back to the user who submitted the data. We assign each user a unique user id for the purpose of organizing the data and associating each user with their own submitted reaction time data, survey data, and mobile applications installed data. In the event that a user submits more data after they have already submitted some data, we can just add the new data to their data that already exists on the server instead of creating a new entry in the user table. The reaction time table will house the data on the reaction times the user had while they played the game. The survey table will contain the user's answers to the survey that they filled out before they were able to play the game. Finally, the user table will store data on the user such as their unique user id.

4.6 Privacy and Security of User's Data

Due to the sensitive nature of the data that we are attempting to collect, our mobile application must be secure and also must not store any data locally on the mobile device. We also will not collect any personally identifiable data about users to could be used to trace the data back to the user who submitted the data. However, if there is a circumstance where there are multiple users on the same device, we will need store some local data on the mobile device. In this case, an anonymous identifier will be generated locally to differentiate between the different users using the same mobile device to play the game and the anonymous identifier will be stored on the mobile device.

4.7 Server Set Up

A server had to be set up to accept data sent from the user's mobile device, which is uploading the response time data from the user for further processing along with the application data. We chose to utilize a LAMP set up for the server for easy and efficient set up of the server. The server should be capable of handling requests from multiple clients at once without experiencing any noticeable slow down and will be secure to prevent any possible theft or corruption of the data that will be stored on the server.

4.8 Development of Game and Game Mechanics

To gather data from user, we thought that the best way to get a large and varied pool of sufficient data would be to develop a mobile application that is a game. This way, it would be entertaining for the user to play and the user may be willing to submit more data by playing longer or more often. Since the game is mobile, we would be able to reach an audience all across the United States and possibly even globally.

To develop the game, we first needed to decide on the game type, we decided on a 2d side scroller. To gather response time data, we decided to have objects which spontaneously pop up in front of the user. The response time is then calculated by the amount of time it takes the user to press the screen to jump over the object from the time that the object first appeared. If the user is unable to dodge or jump over these items, then they will lose a health point. When the use runs out of health points, the game will be over. The user will have a score and they can increase their score by lasting a longer period of time in the game and possibly by collecting bonus items throughout game play. We will also like to add a leveling system so that we can make the levels become incrementally harder or just to give the user's a change up of scenery in the game so that it doesn't get as boring as fast. The implementation of the health system, score system, and level system aren't necessary, but we believe that by implementing them, we may be able to attract more users and also keep more users playing the game for long periods of time, which will allow us to gather a larger pool of data for each user.

In order to implement these we need to have a variety of mechanics in the game set up for all of these tasks. A big mechanic is collision detection. Thankfully, Unity provides some tools for collision detection which make it an easier task. Collision detection is used to detect whether or not the user successfully jumps over objects or collides with any other objects on screen. Using Unity's built in physics tools, we added are able to simulate the user jumping and hence are able to make the user jump over objects by having the user click on the screen.

4.9 Server Side Scripts for Analyzing Data

Server side scripts are the scripts that will reside on the server which are responsible for processing and analyzing the response time data that the users send to the server from their mobile devices. These scripts will be written in the Python programming language and will use the numpy and scipy libraries for performing the statistical analysis on the data.

These Python scripts will be responsible for accepting data, parsing the data, organizing the data, and storing the data into the SQL database located on the server. These Python scripts will also be responsible for pulling data from the server and mining the data for any possible patterns that may exist in the data. We hope to find some sort of relationship between reaction times and people who have been diagnosed with ADHD and haven't been diagnosed with ADHD. We also hope to find some sort of relationship between the number of mobile applications that the user has installed and the types of mobile applications that the user has installed and whether or not the user has been previously diagnosed with ADHD. The main statistics the scripts will be calculating and analyzing are the correlations between different aspects of the data and also confidence intervals. We hope to establish strong confidence intervals with our results so that we can effectively show how strong our conclusions are.

4.10 Testing

It is extremely important to have an application that is reliable and performs as expected under any scenario without experiencing any failures. To make sure our application and server are as reliable as possible we will need to perform testing on both the mobile application and the server.

4.11 User Interface

To test how intuitive the game is, we will ask those unfamiliar with the game to attempt to play the game and gather their feedback on the game and ways to improve the game. We need to ensure that the ease of use on touch devices is also good. For example, we need to make sure that the icons in the game aren't so small that a user with rather large fingers would have trouble playing the game.

4.12 Overall Systems

Overall, our game should be fast and responsive. The user shouldn't have to wait more than a second or two to upload their data. So, tests will have to be performed to ensure the speed of the application is fast, even when the server is experiencing a heavy load.

4.13 Stress Tests

Manual and automatic stress tests will be performed on the server and on the mobile application to ensure that they both can handle heavy loads without breaking or experiencing any other unexpected behavior. With the manual testing, a group of 10 users or more will simultaneously perform the same functions and attempt to upload their data to ensure that no wait times or ramp up times occur. We can also perform automatic tests using open source software testing tools to stress test the server by sending a bunch of requests to the server and seeing how the server handles them. To test the stability of the mobile application, we can use monkeyrunner or anteater to simulate hundreds of pseudo random presses in places on the game screen to see if the mobile application can handle it without crashing.

References

[1] S. L. Karalunas, H. M. Geurts, K. Konrad, S. Bender, and J. T. Nigg. Annual research review: Reaction time variability in ADHD and autism spectrum disorders: measurement and mechanisms of a proposed trans-diagnostic phenotype. *J Child Psychol Psychiatry*, 55(6):685–710, Jun 2014.

[2] Stewart H. Mostofsky and Daniel J. Simmonds. Response inhibition and response selection: Two sides of the same coin. *J. Cognitive Neuroscience*, 20(5):751–761, May 2008.

[3] Leila Cristina Vasconcelos de Andrade, Luis Alfredo Vidal Carvalho, Cabral Lima, Adriano Cruz, Paulo Mattos, Carlos Franco, Adriana Soares, and Bruno Grieco. Supermarket game: An adaptive intelligent computer game for attention deficit/hyperactivity disorder diagnosis. In *Proceedings of the Fifth Mexican International Conference on Artificial Intelligence*, MICAI '06, pages 359–368, Washington, DC, USA, 2006. IEEE Computer Society.

[4] Fabio E. G. Santos, Angela P. Z. Bastos, Leila C. V. Andrade, Kate Revoredo, and Paulo Mattos. Assessment of adhd through a computer game: An experiment with a sample of students. In *Proceedings of the 2011 Third International Conference on Games and Virtual Worlds for Serious Applications*, VS-GAMES '11, pages 104–111, Washington, DC, USA, 2011. IEEE Computer Society.

[5] J. N. Epstein, J. M. Langberg, P. J. Rosen, A. Graham, M. E. Narad, T. N. Antonini, W. B. Brinkman, T. Froehlich, J. O. Simon, and M. Altaye. Evidence for higher reaction time variability for children with ADHD on a range of cognitive tasks including reward and event rate manipulations. *Neuropsychology*, 25(4):427–441, Jul 2011.

[6] P. Andreou, B. M. Neale, W. Chen, H. Christiansen, I. Gabriels, A. Heise, S. Meidad, U. C. Muller, H. Uebel, T. Banaschewski, I. Manor, R. Oades, H. Roeyers, A. Rothenberger, P. Sham, H. C. Steinhausen, P. Asherson, and J. Kuntsi. Reaction time performance in ADHD: improvement under fast-incentive condition and familial effects. *Psychol Med*, 37(12):1703–1715, Dec 2007.

[7] Sanjana Srivastava Jennifer Schumann Jaideep Srivastava Monika D. Heller, Kurt Roots and T. Sigi Hale. A machine learning-based analysis of game data for attention deficit hyperactivity disorder assessment. *Games for Health Journal*, 2(5):291–298, 2013.