The 18th Winona Computer Science Undergraduate Research Symposium

April 25, 2018

4:00pm to 8:00pm

Pasteur 307

Winona State University
Winona, MN

Sponsored by the Department of Computer Science

at Winona State University

WINONA STATE UNIVERSITY

Computer Science Department

http://cs.winona.edu

Table of Contents

Bot or Not: Detecting Bots in Online Multiplayer Video Games through User Input

Alexander Boutelle
Winona State University: Department of Computer Science
1759 West 5th Street
Winona, Minnesota
(507) 458 3162
alexboutelle@hotmail.com

ABSTRACT
The main purpose of our project is to determine how a bot differs from a human and how that difference can be used for developing a metric that be used to detect botting in online multiplayer video games, specifically through user input. A bot is a program that automates the gameplay in a video game. Our stance is that modes of detecting botting are too intrusive on the player. For example, some video game developers choose to deploy software that runs concurrently with their video games to check for cheating programs on the user's device. Proponents to this type of mechanism consider process-monitoring software to be a form of spyware. We have found many different ways that we can monitor a user's input that can be helpful in creating a metric that will allow to detect botting in a video game that is non-intrusive or hinders gameplay in any way. To determine how bot and human behavior differ, we observed a user's mouse clicks and keystrokes that the game server received, and from their input, we could see if a user's behavior is similar to that of a bot. From the gathered input, we were able to observe key differences between bot and human behavior.

Keywords
Bots, Cheating, Detection, Key logging, MMORPG

1. INTRODUCTION
In online gaming, botting allows an illegitimate player to perform actions that may put legitimate players at a disadvantage. A bot may allow an illegitimate player may utilize a bot to harvest resources in the world; these resources would then become unavailable to a legitimate player who are not automating their game. Players that bot within games will begin to spoil the experience of legitimate players, as they will feel cheated and frustrated, at which point they may move on to a new game. Lack of prevention of botting will be damaging for a developers reputation as well, as rampant botting reveals security flaws and a lack of empathy towards players. As such, many developers choose to implement a way to prevent cheating within their games as way to maintain their reputation and legitimate customers.

Proceedings of the 18th Winonas Computer Science Undergraduate Research Seminar, April 25, 2018, Winona, MN, US.

The ability to bot in a video game usually presents security problems in the game itself. One way to enable bots in a video game is use legitimate API's to read game data and then using certain function calls to automate processes in the game [1].

Developers have many options when addressing the issue of cheating in there video games. One such way is to have a program scanning processes on the client's machine, and if the program detects a known cheat, it will forward it the server and halt execution of the video game on the client's machine. Blizzard Entertainment utilizes such a mechanism; however, it is a controversial method of preventing cheating as many consider it spyware.

An alternative proposed method is that instead of scanning the client's machine for processes, we gather information from user input. One could then compare the client's input to that of a bot and if the results are within a certain threshold, the client may be utilizing a bot. We have gathered input data, keystrokes and mouse clicks, from human and bot subjects to see if we can identify a significant difference between bots and humans.

I hypothesize that user input can help game developers determine if a player controlled character in a video game is a human or a bot.

2. METHODOLOGY
Botting is done through calling I/O APIs directly, rather than using a mouse and keyboard. Bots are used extensively in MMORPGs, as the nature of those video games are to progress a character, and progression is mainly inhibited by time and a bot removes this factor [1, 2].

The gaming industry has developed many different modes to prevent botting in their video games. Games, such as World of Warcraft, utilizes a cheat detection mechanism called Warden. Warden will scan background processes on a client's device as the video game is being played, and if it encounters a process that Warden recognizes as a cheating process, it will not allow the device to connect to the game server and will disconnect any active game client from the game server [3]. The author of another paper [3] claimed to develop an anti-cheat mechanism using anomaly based detection, this is where it checks how the client is behaving when it is being executed and then it compares it to a correctly operating client on the servers end, and if they are different a cheat is being used. In another study [4], the author attempted to combat botting through pattern recognition, as he observed how a bot made decisions while playing an online card game. Additional variables to consider in bot detection is resources generated, killing time, and pathing of the player controlled character [5].

In order to test our hypothesis, we used the video game Ashen Empires, shown in Figure 1, to perform testing on seven test subjects whom have prior experience with either role-playing games (RPGs) or massive multiplayer online roleplaying games (MMORPGs) as they will be familiar with the games concepts and will simulate a typical player in the game. All test subjects played the same character on the game and all attributes of this character remained consistent amongst all test subjects. For testing purposes, players used the same mouse, keyboard, and device to perform the testing and the latency between the network and the game server was consistent amongst test subjects. To ensure consistency, any unnecessary processes that were running on the machine were terminated.

Figure 1. ***Ashen Empires was chosen as the game of choice for testing due to its simplicity and lack of botting protection***

Our test subjects are to be chosen if they satisfy the following criteria; the test subject has prior experience with either RPG or MMORPG video games. Additional information to be gathered from the test subjects, such as age, gender, and the amount of time they spend each week playing an RPG or MMORPG video game.

2.1 The Keylogger

To collect the necessary data, we used a keylogger. The keylogger that we used was not an industry standard keylogger, as it is one we created ourselves using C++. The algorithm for the keylogger is shown in Figure 2. The keylogger recorded all keystrokes and mouse clicks done by the machine and it also recorded the timing between keystrokes and mouse clicks; note that these are two different timings, that is keystroke times were compared strictly amongst keystrokes and mouse click times were compared strictly amongst mouse clicks.

The program gathers the timing information by generating time since last epoch via the C++ chrono package. The program will then generate another time since last epoch when a keystroke and mouse click are detected and then to get the time between each stroke, it will subtract from the last recorded time. This does imply that the first keystroke and mouse click are performed prior to testing so that a large time is not recorded. The program will then detect the keystroke or mouse action the user has performed and if the user entered an ASCII character, this is done through the GetAsyncKeyState function. It will then write that character and the time since last keystroke into a log file. Likewise, if the user enters a mouse action, that will be recorded with the time since last mouse click and sent to a separate log file. It is worth noting that the keylogger translates the function keys (F1, F2, etc.) into an ASCII character, this is relevant to testing, as the keylogger will recognize F1 to be the letter "p" in the log file.

1. (Get Input) GetAsyncKeyState(key)
2. (Determine input) If VK_LButton is detected, input is a mouse click, else it is a s keystroke
3. (Computer time interval) Determine current time now and compare with the last input time, last, store now in a temporary variable, temp, to be used later
    a. temp = now
    b. now = now – last
    c. last = temp
4. (Write to log file) Store input and time interval in the appropriate log file
5. (Repeat) Continue until '{' is entered

Figure 2. ***Keylogger algorithm***

We also observed that the keylogger had an error rate of about 2.70% when testing our bot, so all times recorded were offset by about 2.70%. We accounted for this error in our results and conclusion.

2.2 The Bot

The bot that we utilized in testing was a simple macro that would perform keystrokes and mouse clicks on certain time intervals. The macro program that we used for this test was Jitbit Macro Recorder, shown in Figure 3. This macro software acted as a bot in the game by automating the testing process. For this project, the macro recorder pressed the key F1 and left mouse click in order to perform the given task. A delay is needed in between keystrokes because if the program clicks or presses a button too fast, the action will not be performed in the game due to latency between the client and the server. To obtain this delay, we pinged the game server 100 times to gather the maximum latency time and then we used the maximum time to be the delay between each keystroke and mouse click, since if it is set to the maximum time it minimize the chance of failure via clicking too fast. It also takes time to perform the action in the game, so we found the time between each action to be 6.5 seconds.

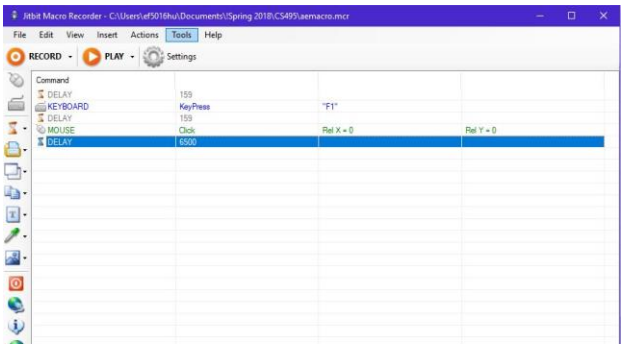


Figure 3. ***A macro was created using JitBit Macro Recorder to automate testing; the macro will continuously press F1 and Left Mouse Click with delays between each action***

2.3 Testing Methods

Players were introduced to Ashen Empires, that is they received an introduction to basic game mechanics such as how to craft items, use key-bindings, and were introduced to the user interface of the game. In order to become familiar with the crafting system, players were instructed to craft 10 string into 10 cloth. Players were then asked to craft 31 cloth, and while they are crafting their keystrokes and mouse clicks were recorded. The amount of items to be crafted is set to 31 for two reasons, first off it is a number low enough where the player will not be prompted with the in game crafting check; this checks to see if players are botting, every 50 crafting attempts a captcha is presented on screen. Second, we wanted to gather information on 30 crafting attempts and since the time since last keystroke or mouse click will be inconsistent amongst all test subjects, the first crafting attempt was thrown away. Each test subject and the bot performed this test three times.

2.4 Keylog Data

The keylogger wrote to a log file the recorded keystrokes and mouse clicks as well as the time between each keystroke and mouse click. From this, we were able to use another program to calculate the average time between each keystroke and mouse click as well as the total time it took to complete the test. We also obtained the minimum time between keystrokes and the maximum time between keystrokes as well as the total amount of mouse clicks and keystrokes. The average response time between the client and server was recorded for each test participant.

From the collected data, we have calculated the Euclidean Distance between the average times of each player and the bot, and from this, we can see if an observable metric is discoverable to see if a player is botting. We have also observed the efficiency of a test subjects keystrokes, that is how many strokes were expected compared to how many strokes it actually took to complete the task. We have also compared the range between the minimum time and maximum time between strokes for the bot and players.

3. RESULTS

From testing, we gathered that the bot executed 32 keystrokes and 32 mouse clicks on average, whereas the humans had 35 keystrokes and 41 mouse clicks on average. Figure 4 displays the average time between keystrokes and mouse clicks between the bot and the humans. The humans had, on average, 6,727 milliseconds between keystrokes and 6,530 milliseconds between mouse clicks, whereas the bot had an average of 6,824 milliseconds for both keystrokes and mouse clicks. From this, an observable difference is seen between the humans and the bot, as the bot is more consistent with its inputs than the human test subjects.

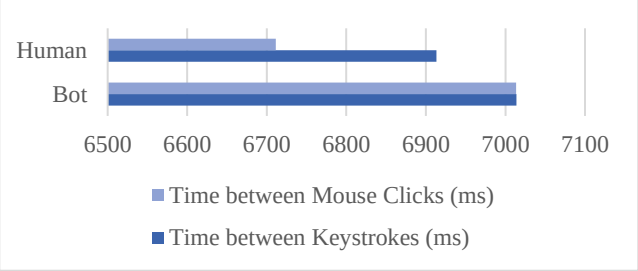


Figure 4. ***Average amount of time (milliseconds) between keystrokes and mouse clicks for humans and the bot***

We computed the range for mouse clicks and keystrokes timings, shown in Table 1. We found that the bot had a range of 21 milliseconds for keystrokes and 18 milliseconds for mouse clicks, whereas the human test subjects had much higher variance in their keystroke and mouse click timings. From this, we can infer that humans tend to be more sporadic with their keystrokes, as they may double click or press a key multiple times.

Table 1. ***Range between the longest and shortest response time (milliseconds) for a keystroke or mouse click for the bot and***

Test Subject	Keystroke Range (ms)	Mouse Click Range (ms)
Bot	21	18
Subject 1	8619	2352
Subject 2	7709	6472
Subject 3	8754	8057
Subject 4	9993	11184
Subject 5	7668	7535
Subject 6	8699	8592
Subject 7	9796	4197

In our testing, we anticipated 30 keystrokes and mouse clicks in each test case. With this information, it is possible to calculate the efficiency of each test subject's inputs. Figure 5 displays the efficiency of the bots and humans keystrokes and mouse clicks, and we found that the bot had a keystroke and mouse click efficiency of 94.16%, whereas the humans had a keystroke efficiency of 85.71% and a mouse click efficiency of 73.17%. The input efficiencies mirror the sporadic behavior found in Table 1, creating a relationship between input efficiency and the variance of input timings.

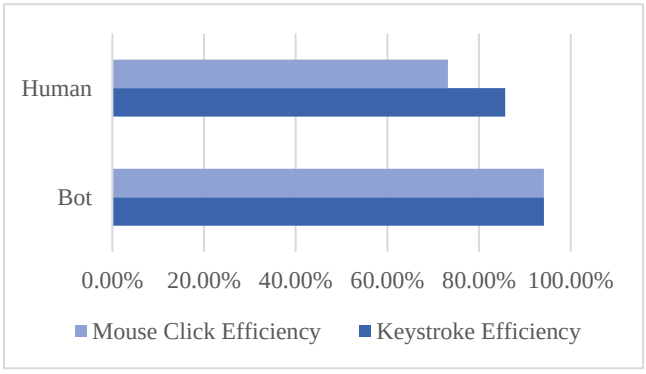


Figure 5. ***Efficiency of keystrokes and mouse clicks for humans and the bot, calculated by taking expected input and dividing by actual input***

4. ANALYSIS OF DATA

Given the variables listed in section III, it is possible to calculate the distance between a human and a bot. Table 2 displays the calculated distances for each test subject and we found that the average distance between the test subjects was 11,186.304. The

distance is a unit-less measurement as it is considering all variables shared between the bot and humans, so it is discovering how different the humans are from the bots.

Table 2. ***Calculated distance between the bot for each human test subject***

Test Subject	Distance from Bot
1	8914.3
2	10059.1
3	12848.3
4	14992.4
5	10835.1
6	12200.1
7	10692.1

From our distance metric, we can conclude that there is a difference between the behavior of a bot and a human in terms of their input to the video game. The human subject that had the smallest distance from the bot was 8,914.3, so this distance could hypothetically be utilized as a way to detect botting in a video game, however it is unrealistic given that it could easily lead to a false positive if another human happened to perform better. Instead, we can observe which areas the bot and human differed greatly and those metrics would be helpful in determining if a player is botting or not.

We can see that humans tend to be more sporadic with their inputs, which are observed by our efficiency metric, as their inputs were generally more inefficient than the bots. Because of this sporadic behavior, humans also had a much larger range between their longest and shortest period between a given input, whereas the bot was more consistent. Therefore, if we were to observe a player over a period time or over a set amount of actions sent to the game server, we could analyze the information sent by the player's client and see how consistent they are. If a player is very consistent in their actions, they could be botting.

5. CONCLUSION

From our analysis, we can see there is a distinct difference between the behavior of inputs between a bot and a human. Given our data, we believe it to be possible to utilize user input as a way to detect bots in a non-intrusive manner. The next step in our research is to perform further testing on multiple bots and humans and then develop an algorithm from their input, and this algorithm should be able to recognize every bot in the test set. We would also like to apply this research to more complex bots that are able to mimic human behavior and then determine if we can use the same methods to determine if we can develop an algorithm that would be able to identify more complex bots.

6. ACKNOWLEDGMENTS

Our thanks to Dr. Silas Bergen for assisting with statistical analysis of our results.

7. REFERENCES

[1] Wu-Chang Feng, Ed Kaiser, and Travis Schluessler. 2008. Stealth measurements for cheat detection in on-line games. Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games - NetGames 08(October 2008). DOI:http://dx.doi.org/10.1145/1517494.1517497

[2] Steven Gianvecchio, Zhenyu Wu, Mengjun Xie, and Haining Wang. 2009. Battle of Botcraft: Fighting Bots in Online Games with Human Observational Proofs. Proceedings of the 16th ACM conference on Computer and communications security - CCS 09(November 2009). DOI:http://dx.doi.org/10.1145/1653662.1653694

[3] Edward Kaiser, Wu-Chang Feng, and Travis Schluessler. 2009. Fides: Remote Anomaly-Based Cheat Detection Using Client Emulation. Proceedings of the 16th ACM conference on Computer and communications security - CCS 09(November 2009). DOI:http://dx.doi.org/10.1145/1653662.1653695

[4] Nguyen, B., Wolf, B. and Underdahl, B. (2013). Detecting and preventing bots and cheating in online gaming. US8708791B2.

[5] Sang-Hyun Park, Jee-Hyong Lee, Hye-Wuk Jung, and Sung-Woo Bang. 2010. Game behavior pattern modeling for game bots detection in MMORPG. Proceedings of the 4th International Conference on Uniquitous Information Management and Communication - ICUIMC 10(January 2010). DOI:http://dx.doi.org/10.1145/2108616.2108656

Load Balancing a 2D Cellular Automata Over a Cluster

Jared DuPont
Winona State University
Computer Science
jdupont13@winona.edu

ABSTRACT

Cellular automata can be used to simulate various physical situations such as population or bacteria growth, but can get very computationally intensive as they grow larger. This paper demonstrates an algorithm to distribute the workload of a 2D cellular automaton across a computing cluster. This is done by dynamically partitioning the regions that each computer is responsible for simulating such that each region has roughly the same work load. This algorithm utilizes all of the computational power of the cluster avoiding some of the computers wasting time being idle. An example cellular automaton rule set is used that purposefully creates concentrated areas of high work load and areas of low work load to better demonstrate the algorithm. Small simulations are not effectively distributed across the cluster due to network overhead.

KEYWORDS

Cluster, Cellular Automata, Load Balancing

1 INTRODUCTION

A cellular automaton (CA) is a method to computationally model a variety of problems such as population or bacteria growth, cryptology, and more. They work by taking as input an environment, an initial state, and a set of rules that dictate how the simulation will progress or update. The environment is typically a grid of cells represented by an array, but they can also be any number of dimensions. The initial state is usually decided by the person running the simulation and is created to test something, for example a single bacteria can be placed somewhere inside the simulation to test how it grows or spreads. In every update cycle or step all the cells look at their neighbors or adjacent cells and then update based on the rule set, an example rule would be that if a cell is surrounded by too many neighbors then it should die from over crowding. With the combination of the rules and initial state of the grid, a CA can generate anywhere from simple patterns to incredibly complex ones. For the purpose of this research a set of rules has been created for a 2D grid with the goal of generating pockets of high intensity areas that require a lot of computation. Area in this case refers to a rectangular subsection of the entire simulation. This will help demonstrate the algorithm by exaggerating how much work needs to be done.

WINONA'18, April 2018, Winona, Minnesota USA

A computing cluster is a group of discrete computers, or nodes, that a master computer assigns tasks to. Clusters are particularly useful in doing tasks who's steps aren't required to be executed in sequential order. Given that CA are generally required to be executed in sequential steps, calculating them on a cluster presents a challenge. If one particular computer's simulation happens to be particularly light, it has to wait for all the other computers to finish before continuing. The goal of this research is to find a method of distributing the computation of the CA in a way that all of the power of a computer cluster can be used. This will be done by splitting up the entire simulation into smaller areas that each computer is responsible for.

Spatial partitioning or subdivision is the act of splitting up an area into smaller chunks based on some rule, a simple way to partition a 2D area is to split it into equally sized rectangles that span the area vertically. In our simulation, the entire 2D grid is split into rectangular chunks that span the area vertically and who's width depends on how much computation or work is in their area. The effect that this will have is that the computers will eventually tend towards taking a roughly equal amount of time to calculate the area that they are responsible for.

In theory a cluster should be able to simulate the CA faster than a computer on its own, and the research has been done based on that assumption. One particular paper[5] covers CA parallelization pretty thoroughly including load balancing, however it does not use dynamic load balancing and only determines what the size of the regions is once at the start of the simulation. In contrast this algorithm continuously updates the size of regions without incurring any additional network overhead, although it does use a much less efficient networking protocol.

2 METHODS

2.1 Equipment Specifications

A computer cluster was built with 15 Dell Optiplex 3020s and one Optiplex 3010. The 3010 was used as the master and the 3020's were used as the nodes. All the computers will be running Ubuntu Server 16.04 with a JVM installed. All the child nodes are mostly homogeneous with 4GB of memory, Core I5 processor, and greater than 100gb of memory. The computers are all wired together over Ethernet and connected to a one gigabit switch. After java code is compiled into a jar, it was uploaded to the master and placed on a networked drive that all nodes have access to. The master's job will be to delegate tasks to the nodes, and the nodes will execute those tasks in parallel.

2.2 Computer Roles

The role of the master node in the simulation will be to handle all high level decisions about the simulation. The master node sets up

the initial state of the CA and distributes chunks of it evenly to all the nodes, after that it does not keep a copy of the current simulation or keep track of the current state of the CA. The master simply keeps track of how complete the simulation is and reports to the user important events such as initialization, percent done, and completion. In this way, the master node does not need to know the rules about the CA or really any information about it at all, just how each child node is operating. An advantage of this is that after the simulation starts, the master node contributes almost nothing to the network overhead.

The role of the child nodes is to simulate its area of the CA. The node computers will contain the rules of the CA as well as its current chunk and information about its neighbors. It will sequentially simulate steps of the CA for a certain number of steps before taking a snapshot of the current board as output. Because each cell of a CA needs the state of the cells around it, the cells at the borders of chunks will need information from neighboring chunks. The node computers will directly communicate with each other without using the master, drastically reducing the master's workload. In this setup, the nodes balance themselves without direction from the master.

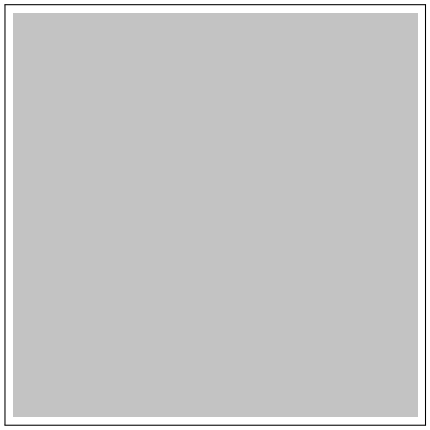
2.3 Nature of the Cellular Automata

An CA was created as an example for this research. The rules of the CA produce a simulation that can operate on a large scale, the cells only require the state of cells imminently surrounding it (the 8 adjacent cells), the amount of work across the board is not homogeneous, and that each cell holds only a small amount of data (A byte). The CA roughly mimics bacteria growth and the cells will be referred to as bacteria.

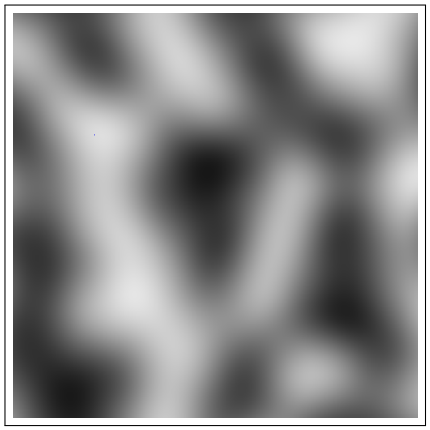
A problem with the bacteria growth that needed addressing was that it spread too uniformly and quickly. To address this a persistent background variable known as 'food' was created that varied across the board in random patterns. To achieve this inconsistent food source, the amount of food at any given location is a function of its position in simplex noise, passed through a filter. Simplex noise is a fast and easy way to generate a gradient noise filled plane, this means that while the pattern simplex noise generates is random, the values are continuous across a line. An example of this can be seen in figure 1, the amount of food at a location is given by the color, black is no food and white is max food. In the uniform version, the amount of food across the board is the same and never changes. In the simplex version, the amount of food is concentrated in some locations and sparse in others. Raw simplex noise was not enough to slow down the growth of the bacteria so the noise was passed through a filtering function to sharpen the edges and create more defined borders as seen in the last image of figure 1. To put it mathematically, the amount of food at any given location is:

$$food = \frac{1 + (\frac{2}{3}\arctan(2\pi * \text{simplex}(x, y) + 1))}{2}$$

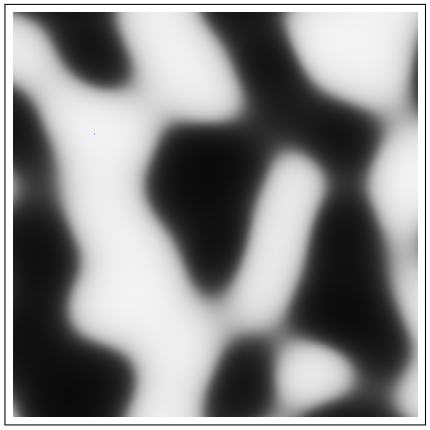
With the information about food, the basic function of the rule set is that a bacteria cell will try to eat as much food as it can to increase its health and then split once it is healthy enough. The amount of food a cell can eat is based on the simplex noise as well



Uniform



Simplex



Filtered Simplex

Figure 1. Various food densities.

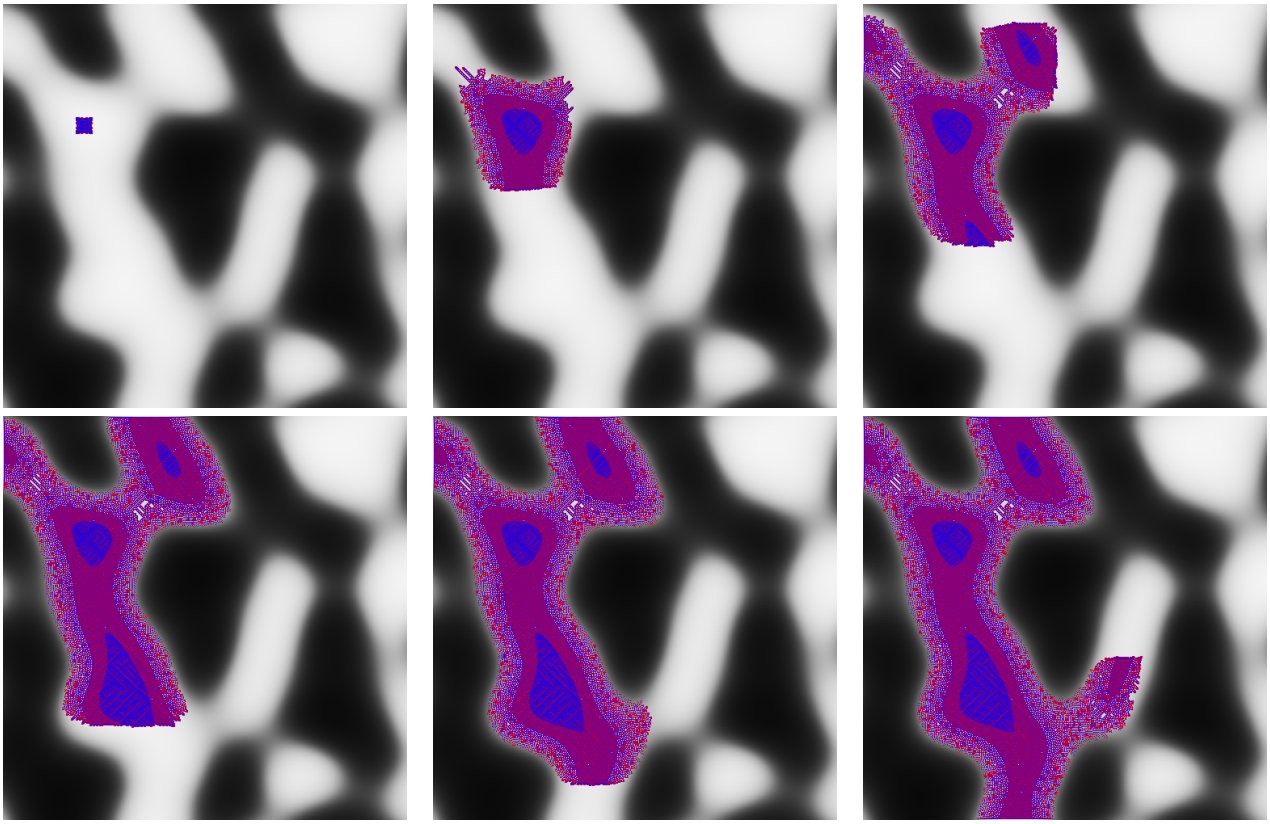


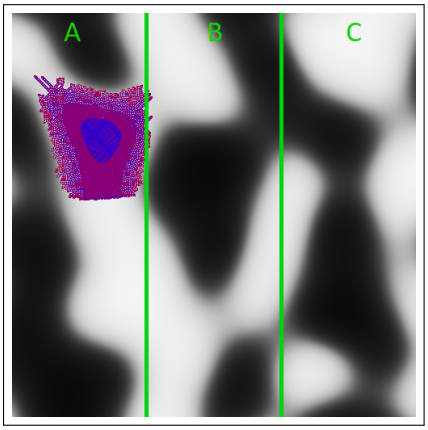
Figure 2. Step 250, 2250, 4250, 6500, 8750, and 11250 of a simulation.

as how many other cells are around it. The progression through time of an example simulation can be seen in figure 2 where the initial bacteria was placed in the top left corner and then left to spread. A colored pixel represents one bacteria cell, its color from red to blue represents how healthy it is from minimum to maximum. A more formal version of those rules is as follows:

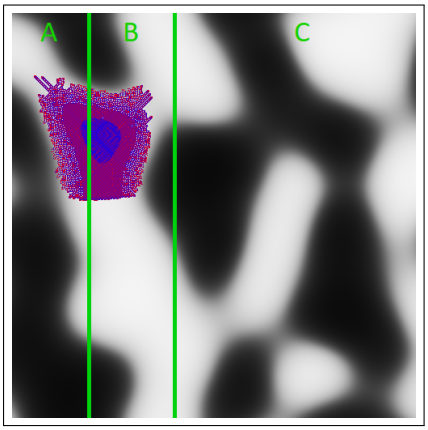
- If the current cell has bacteria:
  - Do some useless computational work to slow down simulation
  - Eat as much food as is available in the current location, given by the filtered food function in figure 1
  - If this cell is above 80% health, check each adjacent cell for a space to expand into and reduce 10% health for each spot it can expand to
  - Reduce health a fixed amount to simulate hunger
  - Reduce health a fixed amount for each neighbor to simulate crowding
  - If health is less than 1, die
- If the current cell has no bacteria:
  - Check each adjacent cell for a bacteria that can split and gain an amount of health for each bacteria that can split into this cell, thus making a new cell

2.4 Nature of the Subdivision

Two subdivision strategies were used, simple and adjusting. When the simulation is starting both the simple and adjusting algorithms divide the space into equal sized rectangles that span the height of the whole board. This means that when both simulations start, the two algorithms will initially look like simple partitioning in figure 3. The green lines represent the border between two nodes, so for example in figure 3 the simple version has three nodes, A, B, and C that are the same width. However, in the adjusting version the width of each node is changed dynamically over time, in figure 3 node A and B both grew smaller to concentrate computation on the bacteria and C grew larger to take over what the other nodes left behind. This approach greatly simplifies the algorithm by reducing the problem to one dimension and reduces the amount of data that needs to be swapped between nodes. One problem with this method is that if the work load is spread horizontally across the top of the simulation area uniformly, the master will not be able to allocate more nodes to the area of high work because it only has the ability to adjust width and not height. The algorithm to partition the space is simple in it's operation, as an example two nodes are next to each other, node 1 and node 2. Each pair of nodes keep an open TCP connection and upon completion of a step, send out a packet to their left neighbor containing a 64 bit long of how much time the node took to simulate in nanoseconds. So in the example node 2 sends the time packet to node 1 and node 1 then



Simple



Adjusting

Figure 3. Different partitioning types, images created as an example.

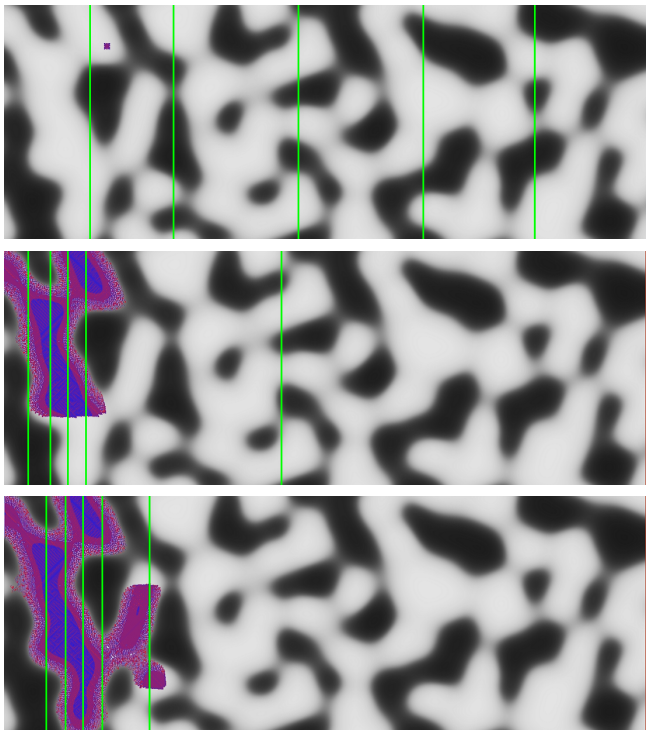


Figure 4. Boundaries of the nodes adjusting themselves as the simulation progresses. Images were generated from actual simulation.

compares how long node 2 took to finish to itself. If node 1 finished first that means that node 1 was wasting time waiting for node 2 to finish. In this case node 1 sends node 2 the rightmost column of the simulation, shrinking node 1 and growing node 2 by one pixel in width. If node 1 finished after node 2 then the opposite is true and node 1 requests a column from node 2. This exchange happens every step of the simulation which means that the nodes are always being adjusted. If any node takes a long time to finish it gets smaller and if it finishes quickly then it gets bigger. In this way the entire simulation slowly adjusts to have a homogeneous work load.

Two CA simulations were be used to evaluate the algorithm, one that adjusts dynamically and one that does not adjust. These two methods can be compared to determine which makes best use of the hardware. To compare the two different types of partitioning, time spent doing the calculation and time spent waiting for network exchanges were recorded. With that information it is possible to evaluate if the complex partitioning offers any advantages over typical partitioning. The actual effect of the load balancing can be seen in figure 4, the green bars represent the borders between nodes and the area between the bars is what each node is simulating. As the simulation is progressing, nodes on the left start to shrink while nodes on the right start to grow due to the increased work load of the bacteria.

2.5 Tests

The tests were conducted using 6 nodes. Four different sized CA were simulated with and without the balancing algorithm for a total of 8 simulations. The total height of the simulation remained constant at 1000 cells tall and the sizes describe how wide each node is at the start of the simulation, so a size 260 simulation is 1560 cells wide when put together. The four tested sizes were 260, 500, 1000, and 1500 because below a width of 250 it would be advantageous to run the simulation on a single computer and because above 1500 takes too long to reasonably measure.

The simulation time and the idle time were recorded for 50 step periods and printed for data collection. Additionally an image is rendered at these 50 step periods for data visualization purposes. At the end of a simulation, the images were downloaded from the cluster, stitched together, and then rendered into a video to help with debugging and ensure the system works.

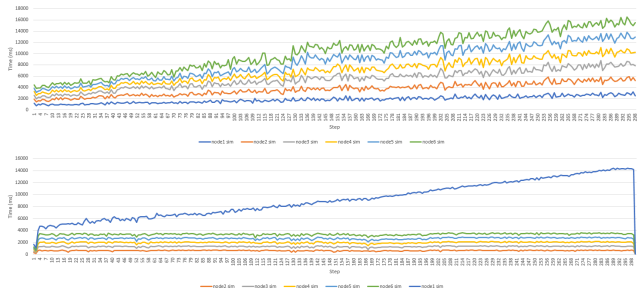


Figure 5. Stacked graph of time spent doing the CA simulation per 50 steps for each node. Top image is the load balanced version and the bottom is the non-load balanced version.

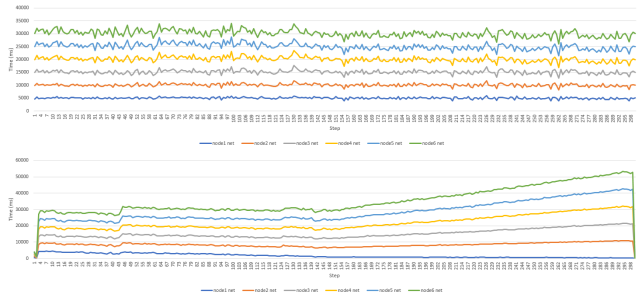


Figure 6. Stacked graph of time spent waiting for network transactions per 50 steps for each node. Top image is the load balanced version and the bottom is the non-load balanced version.

3 RESULTS AND ANALYSIS

Across all 4 sized simulations, the results were all the same, the amount of total time spent doing the simulation was the same between the load balanced and non load balanced, however the time spent waiting for network transactions was significantly less for the load balanced version.

The difference between the load balancing and non load balancing algorithms in reference to simulation time can be seen in figure 5. The graphs are stacked which means the topmost line also represents the total time spent in that frame between each node. Both graphs have the same profile which means that both simulations took nearly the exact same total time to simulate a frame. The difference though is that in the load balanced version all of the nodes simulated for a roughly equal amount of time where in the non load balancing version almost all of the simulation time is concentrated in node 1.

The advantage of load balancing can be seen in figure 6. In the load balancing simulation the time spent waiting for network remains constant throughout the entire simulation. In contrast the network idle time increases over the entire simulation.

Figure 5 and figure 6 are both size 500, but the other 3 sizes have similar results. This is shown in figure 7 which shows how much

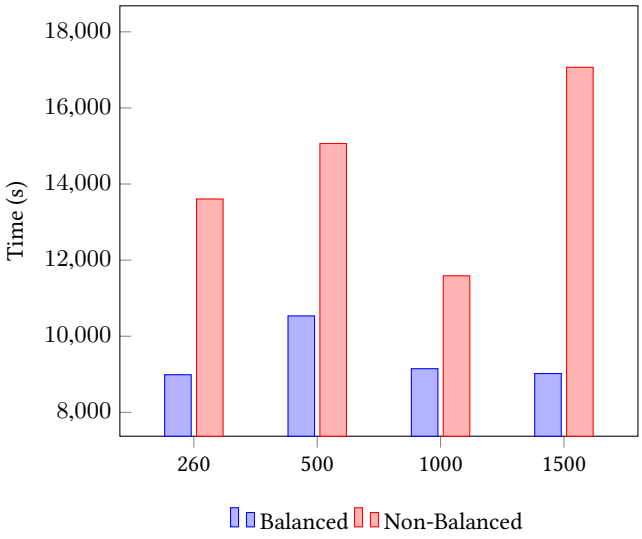


Figure 7. Total time spent idle for each of the simulations.

time was spent waiting for network transactions for each of the 8 simulations.

4 CONCLUSION

Cellular automata are tools that researchers use to simulate various physical phenomena such as bacteria or population growth. Due to their computational work load, it is advantageous to distribute the simulation across a cluster. It is important that the cluster does not waste time on tasks other than the simulation of the CA such as waiting for other nodes to finish, so a method to balance the work equally is important. A CA that is load balanced by nodes trading columns with their neighbors based on calculation time should reduce total time spent waiting for those neighbors to finish. When compared to a non load balancing version, the load balancing CA does in fact reduce idle time.

REFERENCES

[1] René Reitsma, Stanislav Trubin, Eric Mortensen *Weight-proportional Space Partitioning Using Adaptive Voronoi Diagrams.* Published online: 1 March 2007

[2] Robert Lubas, Jaroslaw Was, Jakub Porzycki. *Cellular Automata as the basis of effective and realistic agent-based models of crowd behavior.* Journal of Supercomputing. Jun2016, Vol. 72 Issue 6, p2170-2196. 27p.

[3] Tammy Dreznerl, Zvi Drezner. *Voronoi diagrams with overlapping regions.* Published online: 25 April 2012

[4] Xinli Ke, Lingyun Qi, and Chen Zeng. *A partitioned and asynchronous cellular automata model for urban growth simulation.* International Journal of Geographical Information Science. Apr2016, Vol. 30 Issue 4, p637-659. 23p.

[5] Xia Li, Xiaohu Zhang, Anthony Yeh, and Xiaoping Liu. *Parallel cellular automata for large-scale urban simulation using load-balancing techniques.* International Journal of Geographical Information Science Vol. 24, No. 6, June 2010, 803–820

Predicting Attrition in Children's Well-being Program Using a Neural Network.

Uzma Ghazanfar
Winona State University
Computer Science Department
ughazanfar14@winona.edu

ABSTRACT
An artificial neural network is trained to predict attrition in Whole Kids Outreach (WKO) - a children's well-being program. The model uses backpropagation - a supervised learning mechanism - to learn and is then tested. This paper presents the findings of the study that indicate that attrition can be predicted with reasonable accuracy and that neural networks can be trained to anticipate future refusal-of-services using the organization's data. This paper also lists the factors that influence the prediction of attrition.

Keywords
Artificial neural network, classification, multilayer perceptron, churn analysis, child welfare, machine learning

1. INTRODUCTION
There are a number of children and family organizations in the United States that work to improve lives. One of these organizations, Whole Kids Outreach (WKO), offers home visits and center-based programs that support children, women, and their families in isolated, low-income, rural areas (specifically in Missouri) to help children live holistic, productive lives free from preventable abuse, neglect, and illness. To be successful in their mission, it is important that WKO be able to identify and predict client attrition so that an intervention process can be initiated, if necessary.

In this paper, a prediction mechanism is proposed to accurately predict families that are prone to withdrawing from the program in early stages of their involvement so that they can be targeted and helped [2][4][5]. The proposed method, the multilayer perceptron (MLP) classifier, a member of the artificial neural network (ANN) family, uses the longitudinal data collected by the organization for over a decade to recognize patterns displayed by the various features that could be affecting refusal-of-services by families. The MLP is chosen for this study because they are capable of non-linear, multi-class classifications.

The research question was: can a multi-layer perceptron with back-propagation be trained to predict refusal-of-services by a client at WKO with reasonable accuracy? The results demonstrate that, once trained on the plethora of data, the network can predict attrition due to various reasons, with reasonable accuracy.

Proceedings of the 18th Winona Computer Science Undergraduate Research Seminar, April 23 & 25, 2018, Winona, MN, US.

2. METHODOLOGY
2.1 Data Description
Since this is a research project based on a real organization, the data being used in this study is longitudinal and has been collected by WKO over the span of its existence. Funded by Healthy Families America, there is information that WKO collects that includes answers to a wide range of questions. This information could prove useful when identifying potential refusal-of-services.

This data has been collected by a number of different social workers over the years using an HTML form with text fields. However, a significant amount of the information recorded, had not been standardized.

2.1.1 Data Cleaning
For the real-world data to be utilized successfully, it first had to be cleaned. OpenRefine, a powerful open-source tool by Google, was used to clean the messy data and make large-volume transformations required to manipulate the data into shape. Using this tool, the data was edited, cleaned, and adjusted. The tool helped uncover a number of issues because it allowed a holistic view of the data. Some of these issues were the number of missing values and inconsistencies in the data records. A technique called faceting was also used to check the different values in a column. This revealed that a number of values used were misspelt, which were then transformed and condensed into smaller, more relevant categories. Overall, the tool allowed us to filter and divest the data we did not need, retaining only those rows that could be used.

2.1.2 Data Selection
In preparing for the MLP, the data also had to be balanced. This meant that there needed to be a balance between clients who had refused services, and those who had either completed the program or left for other reasons since there were a lot more people who completed requirements that those who didn't. Balanced data gives the system an unskewed data set to learn from and, as previous studies have shown [2], it results in a better fitted model, and a more accurate result instead of one in which most data are incorrectly classified in favor of the weighted data.

Data from the first three visits of 423 families was used for this study. Out of these clients, 274 had refused services. Some of the variables that were collected from each group included general information of the adult, their medical information, information regarding their employment skills, and their high school education.

2.1.3 Issues
One of the issues faced was that the lack of meta-data on the data. Because of a number of anomalies, the data had to be manipulated.

An example is the Psychiatric History of a client that has been recorded once to be true but had been recorded as false in other visits. Considerable back and forth between us and the clients resulted in this issue being resolved.

There was also no information on when some factors were first recorded. This became an obstacle since an enormous amount of our data was discarded due to missing information.

TypeOfAid	PsychiatricHistory	DS1	DS2	DS3	DS4	DS5	DS6	DS7	DS8	DS9	DS10	NumberOfLiveBirths
	0											0
TANF | WIC | Fc	0											
	0											0
	0											0
	0											0
	0											0
	0											0
TANF | Food St	0											0

Figure 1 - Columns with over 70% of their rows empty such as Depression Scores

Figure 1 shows an example of a subset of rows that had over 70% of their cells empty. This could potentially have an adverse effect on the classifier training, since the result would be meaningless. Hence, variables with over 70% empty cells, were discarded. These variables included some of the variables the social workers at WKO had earlier identified as important factors in influencing attrition, such as Circle of Support, Depression, and Client Level. This left us with a data set of 2,000 rows with 423 unique clients to train the MLP on.

Lastly, categorical data had to be converted to numerical. Ordinal data can be dealt with because there is a natural order to them. However, nominal variables have labels and categories without having any kind of natural order. MLP's require all inputs to be numerical to learn from, which can be handled if a variable is ordinal but is harder to do with nominal variables. A python package called "Pandas" allows mapping of all values and labels to integers, including missing values.

2.2 Prediction Model

A neural network is an attempt to model the human brain, so it can perform some of the same tasks that humans are capable of, such as pattern recognition and classification. Neurons are assigned to 3 layers: the input layer, the hidden layer, or the output layer so that they can perform complex operations. Figure 2 demonstrates how inputs interact with one perceptron.

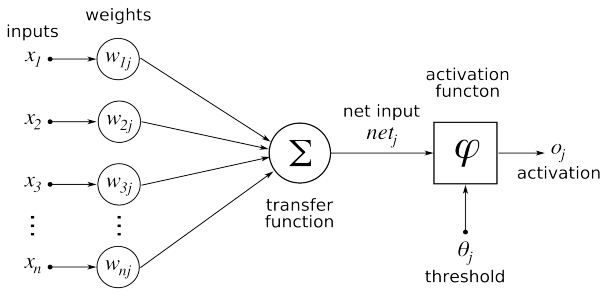


Figure 2 – Inputs interacting with one perceptron

Each variable from the data set forms an input for a perceptron and is assigned an arbitrary weight initially. Each input is then multiplied with the weight of its connection. Each neuron receives all the inputs are summed and transmitted to the next neuron. Each neuron in the layer is an activation function and this is what differentiates one neuron from another. If the summation value is above this activation threshold, the neuron will fire and relay the output as input to the next neuron and activation will propagate the feed forward into the system. The activation function used is the "Rectifier," which has strong biological motivation and results in better training, that is used for training deep neural networks. The initial learning rate was set to 0.001 and regularization was set to 0.0001.

Backpropagation is the mechanism by which the model learns. MLP is a form of supervised learning technique, therefore, it needs to be provided with expected outputs that it can compare its results to. Since information on clients that refused services and those who completed the program was recorded in the data, the MLP was provided with accurate information for the desired outputs. Given this, the MLP computes its output, compares it with the desired output, calculates its error gradient, and propagates it backwards into the network to adjust the strength of the connections. The network runs its computations thousands of times before it comes to the correct conclusion that matches the desired output. The model is then tested.

2.2.1 Overfitting

Overfitting is a modeling error that occurs when a function is too closely fitted to a limited set of data points. It results in a model that learns very well in the training, however, performs poorly during testing. 3 hidden layers with 25 perceptrons each are used to avoid overfitting in the model. Since, the goal is to estimate the accuracy of the model in practice, k-fold cross-validation is also used to ensure the model is not over-fitted. k-fold cross validation is a technique used for assessing how the results will generalize to an independent data set. It does this by partitioning the original into k equal sized subsamples, of which a single subsample is retained as the validation data for testing the model, and the remaining k − 1 subsamples are used as training data.

3. RESULTS

3.1 Confusion Matrix

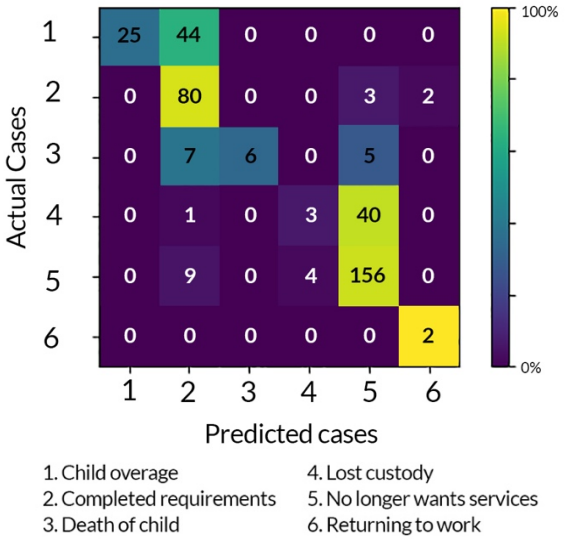


Figure 3 – Confusion matrix with columns depicting the predicted cases and rows depicting the actual cases

A confusion matrix is a specific table layout that allows visualization of the performance of an algorithm. Each row of the

matrix represents the instance in an actual class while each column represents the instance in a predicted class. The name stems from the ease with which it can be seen when and how the system is confusing two classes (i.e. commonly misclassifying one as another). Figure 3 shows our MLP's confusion matrix.

The MLP accurately classified 156 cases out of 169 as clients who no longer wanted services, only confusing 9 of the 169 as people who completed requirements and 4 of the 169 as having lost custody of their child. The model also correctly classified 80 of 85 cases as completed requirements. It misclassified 3 as those who no longer wanted services and 2 as those who were returning to work. The MLP misclassified almost all the cases that resulted in termination due to the death of a child and was confused about cases in which parents had lost custody as well. This could be because there are not enough data points to balance the cases where the client left the program because of the death of a child or because they lost custody.

3.2 Scores: Precision and Recall

One of the metrics for testing the model was precision. Precision is defined as the number of true positives over the number of true positives plus the number of false positives. The overall precision score of the model when testing our data was 73%. Another metric was the recall of the model. Recall is defined as the number of true positives over the number of true positives plus false negatives. Our model's recall score was 70%. This score is far from perfect. However, the breakdown of the scores in Table 1 demonstrates that the model misclassifies "death of child" and "lost custody" cases and this is why the overall results of recall are low.

Table 1. Table captions should be placed above the table

	Precision	Recall
Child Overage	100%	36%
Completed Requirements	57%	94%
Death of Child	100%	33%
Lost Custody	43%	7%
No Longer Wants Services	76%	92%
Returning to Work	50%	100%

In fact, the system's recall for clients who "no longer want services" is 92% and the recall for clients who have "completed requirements" is 94%. The overall score of precisions and recall are an average of each individual score and a few of them are contributing in bringing the overall score down.

4. ANALYSIS

Our model does not give a perfect prediction; however, it is still reasonably successful in predicting attrition. Negative attrition, due to refusal-of-services, is being predicted by the model with a precision of 76% and a recall of 92%, which tells us that certain factors, if present, will make the clients more likely to refuse services. The model gives us a means to predict the tendency of a client to leave the program and will allow the organization to be alerted so that they can divert resources towards the client when they are in the early stages of involvement in the program.

4.1 Coefficients

One important finding of this model is the relative importance of the features that affect attrition. Some of the most important features in the system are defined by the final weights that the MLP assigns each of them in its perceptrons.

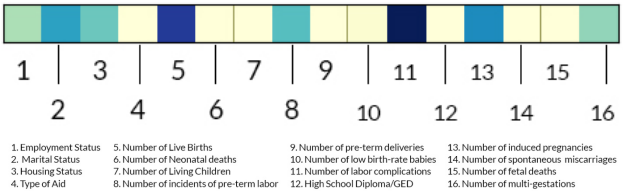


Figure 4 – Example perceptron with weights/coefficients displayed as a heat map

Figure 4 shows an example of the visualized weights of one perceptron as a form of heat map. The darkest panels are those features that have the highest weight, and the highest effect on the system. The output of all coefficient of each perceptron between the 0th and the 1st layer is shown in Figure 5. Summing the weights of these layers illustrates that features 5, 8, 11, and 13 are weighted the most.

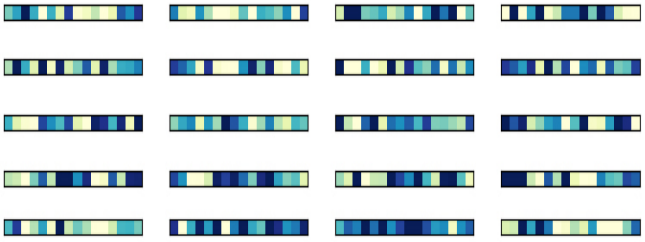


Figure 5 – Coefficients of all perceptrons from 0th to 1st layer

These correspond to number of live births, number of incidents of pre-term labor, number of labor complications, and number of induced pregnancies. A sensitivity analysis on the factors was not done. It cannot be said what nature of effect these factors have on attrition; however, an effect is detected.

Lastly, as successful as the model is at predicting attrition, it is also unable to capture the interaction between these features and treats each of them independently of each other.

5. CONCLUSION

The hypothesis has been proven to be viable. The model does not give us a perfect prediction regarding clients, however, it does predict the tendency. The model can be employed in the beginning of a client's journey in the program and if the model predicts attrition due to refusal-of-services, the specialist should be more vigilant. Currently, we would recommend that there be a change in the data collection procedures at WKO so that the data collected is complete. Missing values in the data disrupt the learning process and become an obstacle in the pattern recognition abilities of the MLP. It is also recommended that there be some standardization of the form that the specialists use, so that there is consistency and conformity across the data being collected.

Since the model indicates that number of live births, number of incidents of pre-term labor, number of labor complications, and number of induced pregnancies are factors that affect attrition the

most, specialists should take consideration of these when inducting clients into the program.

6. ACKNOWLEDGMENTS

This study was supported by data from the organization, Whole Kids Outreach. The author thanks Sister Anne Francioni and the social workers from WKO for their support and patience. The author also thanks Dr. Joan Francioni, Dr. Nina Marhamati, Dr. Mingrui Zhang, and Dr. Sudharsan Iyengar for their careful supervision.

7. REFERENCES

[1] Ya-Han Hu, Chia-Lun Lo, Sheng-Pao Shih, Developing early warning systems to predict students' online learning performance, Computers in Human Behavior, Volume 36, 2014, Pages 469-478, ISSN 0747-5632, https://doi.org/10.1016/j.chb.2014.04.002

[2] Dursun Delen, A comparative analysis of machine learning techniques for student retention management, Decision Support Systems, Volume 49, Issue 4, 2010, Pages 498-506, ISSN 0167-9236, https://doi.org/10.1016/j.dss.2010.06.003.

[3] Ioa Lykourentzou, Iois Gioukos, Vassilis Nikolopoulos, George Mpardis, Vassili Loumos, Dropout prediction in e-learning courses through the combination of machine learning techniques, Computers & Education, Volume 53, Issue 3, 2009, Pages 950-965, ISSN 0360-1315, https://doi.org/10.1016/j.compedu.2009.05.010

[4] Ira M. Schwartz, Peter York, Eva Nowakowski-Sims, Ana Ramos-Hernandez, Predictive and prescriptive analytics, machine learning and child welfare risk assessment: The Broward County experience, Children and Youth Services Review, Volume 81, 2017, Pages 309-320, ISSN 0190-7409, https://doi.org/10.1016/j.childyouth.2017.08.020

[5] Chintan Amrit, Tim Paauw, Robin Aly, Miha Lavric, Identifying child abuse through text mining and machine learning, Expert Systems with Applications, Volume 88, 2017, Pages 402-418, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2017.06.035.

[6] M.A.H. Farquad, Vadlamani Ravi, S. Bapi Raju, Churn prediction using comprehensible support vector machine: An analytical CRM application, Applied Soft Computing, Volume 19, 2014, Pages 31-40, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2014.01.031.

Usability of Web Applications Developed in Angular and Firebase

Kieran Gordon
Winona State University
175 West Mark Street
Winona, MN 55987
(507) 457-5385
Kgordon15@winona.edu

ABSTRACT

Web application development has a variety of tools and technologies that can be used to create applications that are efficient and user friendly. At times it can be difficult to determine what combination of technologies is best for a particular project. Two technologies that are gaining in popularity for web application development are Angular and Google Firebase. Angular is a JavaScript framework and Google Firebase is a cloud document database service. This research examines the usability and user experience of a specific web application developed using the aforementioned technologies. A content management application was developed and a usability test was designed to determine if the application possesses an adequate level of usability. The usability test, along with a demographic survey, was administered to 23 participants. The usability of the application was judged on three criteria: number of mis-clicks, time to complete each task, and whether that task was a success or failure. Results of the test confirm that a content management system made with Angular and Firebase can be both effective and usable.

1. INTRODUCTION

Web applications have become increasingly popular in the last few decades primarily because they are platform independent software. This useful feature of web applications makes them more available to a greater number of people without having to make multiple versions of the same software to ensure cross platform usability. There are many technologies used to design and build web applications. Two such technologies are Angular and Google Firebase.

1.2 Angular

Angular is a framework for front-end web application design that uses typescript, a superset of JavaScript, to facilitate the declaration of static types. Angular was designed by Google with the intention of providing developers with a tool to design single page applications.

Proceedings of the 18th Winona Computer Science Undergraduate Research Seminar, April 23 & 25, 2018, Winona, MN, US.

A single page application is a web application that has only one dynamic html file. Rather than changing the pages of the application as a user interacts with it, the Angular modules simply change which components are rendered on the page at any given time. In other words, the page dynamically rewrites its HTML on the detection of changes within the page [1]. Angular provides this feature by breaking the application up into separate components, which are given change detectors that show where in the DOM the change occurred[4]. Angular is always used in conjunction with a library called *Reactive Extensions*, which equips Angular with *observables*. Observables are functions whose return values arrive asynchronously over time. They are used most frequently in applications to populate a user interface with data that is being passed from a back-end service asynchronously. The object on the client side of the application that is receiving the data is referred to as the subscriber, and as the observable detects new values it pushes them out to the subscriber[5]. An observable is like a yield function in python, or to an iterable in Java.

1.3 Google Firebase

Google Firebase is a document model database that stores information as collections of small documents. Firebase is a cloud database that also offers cloud file storage[2]. Using Firebase allows a developer to create web applications without having to worry about server-side infrastructure. It also does not matter where the application is hosted, since data stored in Firebase can be accessed from any host and is queried directly from the client side using JavaScript. JavaScript can also be used to access the Google cloud file storage, a service associated with Firebase, allowing an application to upload and download files.

1.4 Usability

The combination of Angular and Firebase can be used to make web applications relatively quickly and easily. This is demonstrated by the fact that a developer with little experience was able to create an administration application for a website in a short amount of time. However, functionality isn't the only important part of building an application. The application must also have a design that makes it intuitive for a new user to operate[6]. Although Angular and Firebase are powerful tools to use, they don't guarantee that an application will have adequate usability.

It is the job of a usability test to ensure that an application is intuitive enough to use that new users will be able to operate it

the first time they use it. A usability test is a series of task that participants are asked to carry out while using an application. Each task is focused on testing a certain operation that a user would have to carry out when using the application. Usability tests should be carried out with certain goals in mind, and these goals can be characterized by three questions[7]:

- What aspect of the application is being tested?
- Why is this aspect being tested?
- How will the information that is gained from the test lead to implementations to improve the application?

Designing a usability test while keeping these three questions in mind will allow the testers to gain valuable insights as to what improvements can be made to the design of an application.

2. DETAILS OF THE APPLICATION AND USABILITY TEST

2.1 Whole Kids Outreach

Whole Kids Outreach (WKO) is a non-profit organization based out of Ellington, Missouri, that seeks to address some of the negative influences that children experience in isolated, rural, low-income areas. WKO engages in programs that are designed to relieve the stress put on children growing up in difficult circumstances by offering physical and emotional support to their familial unit. For example, WKO's Healthy Families of America Program sends outreach specialists and registered nurses to the homes of participating families to ensure that the children are receiving nurturing care, and to teach and assist young parents. As a non-profit organization, one of WKO's main sources of financing is through donations, and their website, which can be visited at www.wholekidsoutreach.org, plays an important role in getting the word out about WKO. This website is also a fundamental tool for the organization to collect donations from individuals who are interested in supporting their mission. Currently, the website does not have a content management system, and thus the website information is relatively static. It would be much better if the information on the website is dynamic and up to date, so as to present the most accurate and current representation of the organization to potential donors.

2.2 Building the Content Management System

A content management application was created using Angular and Google Firebase. The management application focuses on a set of core functionalities designed to meet the requests of Whole Kids Outreach (WKO) administration. To meet these needs the application must allow the staff to manage their website's photos and to change certain content that changes frequently. The user interface and business logic of the application is written in Angular and data is stored using Firebase and Google storage. Files will be stored in Google storage and content will be stored in Google Firebase, this content will include references to the files stored in Google Storage. The website was completely static at the outset of this project, and it was written in AngularJS, the precursor to Angular, so the first step was to convert the existing website into Angular. This was relatively easy to accomplish because none of the of original HTML had to be modified to be reused in the Angular website. This made the task of converting the website as simple as creating a new Angular project, creating Angular components for each of the websites pages, and then pasting the existing HTML into its respective component. The content management system was designed as an interface that would allow the user to access information about programs, events, board members, and staff members. The ability to view, edit, create, and delete information of any of those four categories was the main request of the organization. Thus, the components that were created through the conversion of the original website were a home page component, programs component, programs sub-page component, and staff page component. The information about these entities was stored in Firebase, and the application makes use of forms to create new objects in the database and to edit existing objects. The application also allows the user to view existing objects on the same pages where they are created.

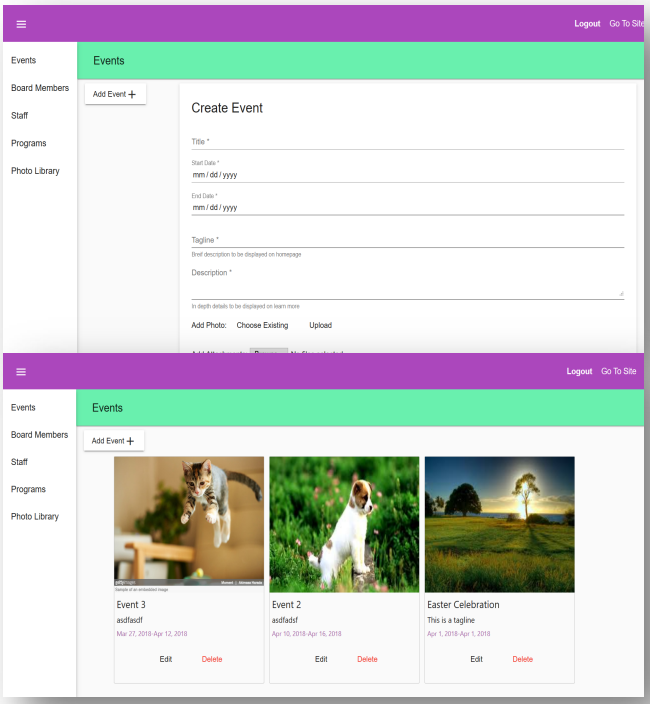


Figure 1. Event object creation form and event view.

Fig. 1 Shows examples of the form used to create a new object, as well as the interface to view existing objects. This example is from the events object page, but the pages for the other objects are similar enough for this to represent them as well. Both the website and the administrator application maintain a connection to the same database hosted by Google Firebase. This allows for the website to pull the event, program, board member, and staff member objects out of the database every time it loads. As the objects in the database change due to user activity on the administrator side, so do their representations on the website.

2.3 Usability Test

A demographic questionnaire and a usability test were used to test the usability and user experience of this application. First The

demographic test ascertained the participants age and computer experience. This information was used to test whether the background of the participants effects how they score the usability of the application. Next, a usability test was administered to the participants. The usability test was composed of 6 separate tasks, and all of the tasks were related to the functionality requests of the organization. The tasks were as follows:

- Upload 5 pictures to the photo library, so that they can be used for the remainder of the test.
- Create a new event by filling in all of the fields of the create event form and add a photo to the event by selecting one of the uploads.
- Create another event, like in task 2 but this time add an attachment to the event.
- Create 2 new board members and 2 new staff members. No pictures are required for this, but you still must fill in all information in the creation forms.
- Create a new program using the photos that you uploaded to the library for the image selection choices on the create new program form. Make the program a featured program.
- Edit some of the things you created, and make sure those changes took effect.

Scoring of the usability test was achieved by using a monitoring software called *Full Story* to record the actions of the participant as they complete the tasks. This allowed the scorer to monitor whether or not the task was completed and the time it took them to complete each task. The monitoring software was also used to track keystrokes, which were used to track the number of times the user committed a mis-click, or clicked a component not directly involved in completing a task. Lastly, the time it takes a participant to locate the necessary inputs on each page was tracked.

3. RESULTS

The administrator application for the Whole Kids Outreach website was successfully created using Angular and Google Firebase. The application offers the user five main functionalities, and each functionality is working as expected. The Whole Kids Outreach website has pages for displaying the organizations employees and board members, displaying the programs the organization offers, and displaying events hosted by the organization. The administrator application offers the user the ability to add, delete, and edit employees, board members, programs, and events. A photo gallery was also successfully implemented that allows the user to upload photos and PDF files and attach them to the events and programs.

Demographic information was collected from the usability test by means of a pre-test survey. The two demographic categories that were considered were computer experience and age. Fig. 2 below shows the break down in age of the participants who took the test. The graph shows that there was a concentration of participants between the ages of 20 and 40.

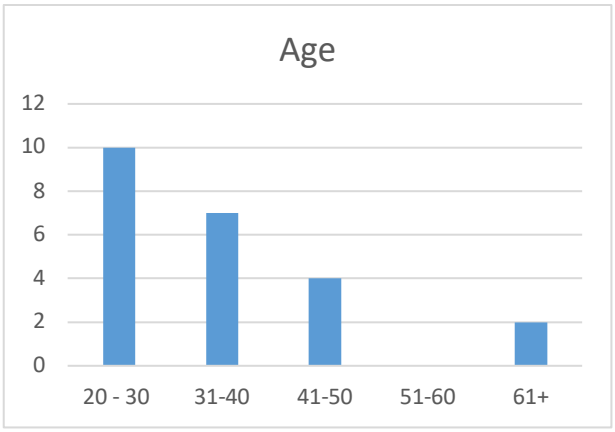


Figure 2. Demographic results for age out of 23 participants.

Fig 3. Shows the second piece of demographic information. As you can see from the char a majority of participants landed in the "some experaince" category. It would have been ideal to have a more diverse sampling of computer experience, but it is important to note that a vast majority of people these days have at least some experaince with computers.

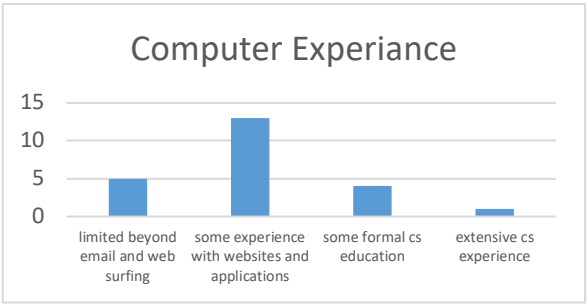


Figure 3. Demographic results for computer experience out of 23 participants.

After completing the demographics assessment, the users then moved on to the usability test tasks. The usability test consisted of 6 tasks, and the results of the test from each user were compiled into averages. 23 users were tested, and the results of the usability test are shown in Table 1 below, where the testing criteria is listed at the top, and the tasks are on the left.

Table 1. Compiled usability test results.

Task Name	Mis-Clicks	Time Taken	Passes	Fails
Photo Upload	3	11.65	20	3
Create Event	5	21.26	19	4
Event With PDF	2	22.16	22	1
Create Board Members	3	16.83	23	0
Create New Program	5	24.35	20	3
Edit Items	7	27.57	23	0

4. ANALYSIS

The data in Table 1 shows that, of 161 total task attempts, there were only 11 fails, which is a failure rate of 6.8%. The data also shows, however, that there were quite a few missed-clicks for the edit task. Observations of the users during the usability test indicated that it was not clear enough that you could edit a component from the same location in the application that you created the component. The plan to reconcile this deficiency is to add the word "Manage" in front of each tab of the side bar shown in Figure 1, i.e., instead of just saying "Events" it will say "Manage Events." Another main concern from the usability test was that there were failures due to errors in the software causing the improper managing of information between the application and the database. All the problems that the application was having involved the handling of images. Images that the user was trying to select from the image gallery and add to events and programs were not be correctly added. This was because the image information had to be passed asynchronously between two separate Angular Components, and the user was pressing the submit button before that asynchronous call could be completed. This problem was fixed by setting a java script timeout function between the initiation of the asynchronous call and its completion. This timeout function forces the user to wait until the asynchronous call is completed before they can press the submit button. Lastly, located on the homepage of the website are event tabs that show all the events that were created using the content management application Fig. 4 show an example of what this event ticker looks like.

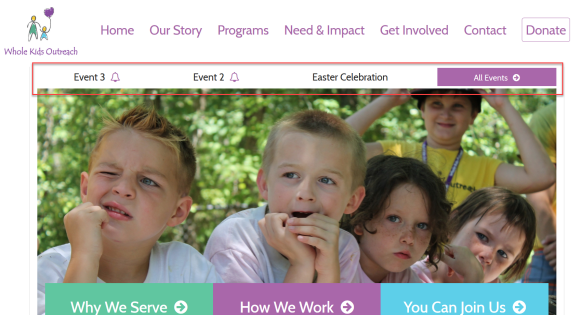


Figure 4. The event ticker on the homepage of the website.

It was hard to decide whether these tabs should be located at the top or bottom of the home screen image. Based on observations during the usability testing it was apparent that placing them at the top of the image allowed the user to locate them quickly when asked to go view their events.

5. CONCLUSION

Angular and Google Firebase exceeded our expectations as tools that can be used to create an effective content management application for the Whole Kids Outreach Organization. Google Firebase is exceptionally easy to connect to using Angular as a front-end framework. The connection can be established in only 5 lines of code because Angular offers an API that is specifically designed for Google Firebase called AngularFire2. The end result was an application that could successfully complete all the tasks required of it by the organization, and an effective user interface based built with Angular's material design components and based on the material design standards of Google Material. By using Angular and Google Firebase, a novice programmer was able to create a decent looking and highly functional content management application in a matter of 2 months. This qualifies Angular and Google Firebase as great tools for a low budget organization to use in order to create a cheap but effective content management system built for their website. The usability results indicate that the application was intuitive enough to enable most people to sit down and successfully complete important tasks that test the functionality of the application. To further ensure that the application is as functional as it can be the next step is to deploy the application onto a server so that it can be accessed by the personnel at Whole Kids Outreach. This will allow them to make judgments and give feedback regarding the usability of the application. They are the ones that will be using it, so even though 23 people have already participated in a usability test, it is important to have the application tested by target users.

6. ACKNOWLEDGMENTS

Thank you to the Winona State Computer Science Department for everything I have learned throughout my time in the program. I would specifically like to thank Dr. Joan Francioni, Dr. Mingrui Zhang, and Dr. Suharsan Iyengar for their support and assistance through the research process.

7. REFERENCES

[1] Codd, E. F. (1970). "A Relational Model of Data for Large Shared Data Banks". Communications of the ACM. 13 (6): 377–387.

[2] Leavitt, Neal (2010). "Will NoSQL Databases Live Up to Their Promise?"). IEEE Computer.

[3] "MariaDB versus MySQL – Compatibility". *AskMonty Knowledgebase. Monty Program. 30 July 2010*. Retrieved 16 September 2012

[4] Flanagan, David, "JavaScript - The Definitive Guide", 5th ed., O'Reilly, Sebastopol, CA, 2006, p.497

[5] Jesse James Garrett (18 February 2005). "Ajax: A New Approach to Web Applications". AdaptivePath.com. Retrieved 19 June 2008.

[6] Tomiuk, D. (2005). Companies' Ability to Foster Loyalty in Traditional Versus Web-Based Service Environments: A Relational Perspective. InternationalJournal on WWW/Internet, 582-586.

[7] James Hom, The Usability Methods Toolbox, available online: http://www.best.com/~jthom/usability (accessed January 6, 2000).

Machine Learning Algorithms to Adjust Images for People with Color Blindness

Jimmy Hickey
Winona State University
Jhickey14@winona.edu

ABSTRACT

Many people suffer from color deficient vision; though they learn to cope, they still have many issues distinguishing between colors. In this research, machine learning algorithms are combinted to help to eliminate some of the problems faced by these individuals. As a prototype, specialized data with intentional color blind issues is generated, transformed, and used to train the system. The colors in these images are then segmented using k-means clustering. The image is then analyzed and corrected if neighboring clusters are found to be conflicting. After correcting, the image is more color blind friendly. Omitted from the prototype is a supervised neural network that will be trained to predict these clusters, allowing this process to be applied on any image.

Keywords

Computer Vision, Machine Learning, Accessibility, Color Correction

1. INTRODUCTION

Color blindness is a deficiency in a person's color vision. A person affected will often confuse different colors that a person with normal vision would not have trouble distinguishing. The most common type of color blindness is red-green, followed by blue-yellow, and the rare full lack of color vision. Red-green color blindness affects "as many as 8 percent of men and 0.5 percent of women with Northern European ancestry." [3] The effects of color blindness range from minor disruptions such as choosing a mismatched outfit to more major difficulties such as discerning between the colors of traffic lights. There is no cure for color blindness, however there exist some methods to alleviate symptoms.

One popular solution to color blindness is specialized glasses. [1] The company enchroma sells a variety of these glasses. Color vision starts in the cones of the eye, of which people have three varieties: green, red, and blue. These detect and distinguish between different wavelengths of light; however, in the eye of a person affected by colorblindness, this discrimination is not always made properly. The enchroma glasses filter out the overlapping wavelengths, making distinctions (particularly between red and green) clearer. These glasses range from $349 to $429 without prescriptions and are not “intended to help pass color blindness tests for occupational purposes." [2] Additionally, they (enchroma brand glasses) are not useful to people with blue-yellow color blindness. A more permanent solution is currently being tested as well.

There is experimental gene therapy that has produced promising results in animals. [5] In one study squirrel monkeys, which are naturally red-green color blind, were made able to perceive differences in the two colors after treatment. These animals were monitored and “retained their new tricolor sensory capacity for more than two years."[5] Additionally, there have been no detected harmful side effects from the treatment; four more animals have been successfully treated since. These auspicious trials suggest that there may be a permanent fix to color blindness in the near future; however, gene therapy is often prohibitively expensive, costings hundreds of thousands of dollars per patient. A short-term, cheap solution to some of the issues faced by the color blind is needed.

The burgeoning field of computer vision lends itself well to this problem. Much work has been done in the area of feature recognition. Projects like Google's Deep Dream and work in robotic vision are some of the wide applications of computer vision. Computers' abilities to analyze images is greatly enhancing. These methods can be used as the small, cheap fix to some color-blind issues. A first step would be to examine images.

Instead of recognizing complex features, a machine needs only to examine the colors in an image. With appropriate knowledge of the problem domain, it should be able to diagnose whether there is areas of the image that may cause issues for the color blind. For example, given a set of color blind tests, it would be able to tell which slides a person affected would have trouble with. This process can then be incrementally furthered. The system can:

1. mark all problem areas.
2. fix the problem areas without losing too much of the context of the image.
3. perform the preceding actions on a video.
4. perform the preceding actions on a live video feed.

For the scope of this research, the first and second objectives will be addressed. Unlike the glasses, the machine is not dealing with the light itself, but how it is displayed. Thus, this could be easily expanded to support all of the common types of color blindness.

The overall goal of this research is to create a prototype to prove that a system can be trained to identify and resolve areas of potential color blind confusion in images.

2. METHODS

2.1 Data Creation

2.1.1 Image Creation

Initially, it was planned to use standard Ishihara Color Test plates, such as those used by eye doctors, as the data for this experiment. See figure 1 for an example. However, this afforded little control of the data. The learning process would rely on only the Ishihara samples publicly available online.

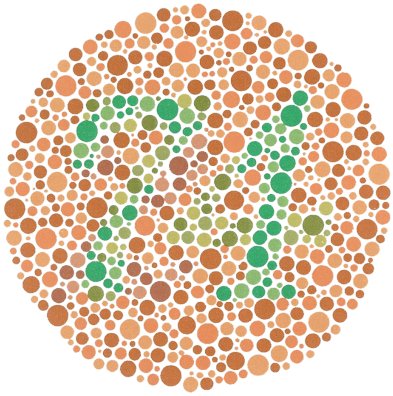


Figure 1. A red-green Ishihara color blind plate. https://en.wikipedia.org/wiki/Ishihara_test

Instead of this, new data was generated for the sake of this experiment. This allows for a wider range of data to be created and manipulated to help the system learn more color combinations. The data created using Adobe Illustrator. A background was chosen and then a foreground image was drawn. The colors of the background and foreground were tweaked until a red-green person with red-green color blindness started having trouble distinguishing between the two. Figure 2 shows pink sailboat on a gray background.

Figure 2. A sample image created in Adobe Photoshop.

For a person with color blindness (particularly red-green color blind), it may be difficult to discriminate between the background and foreground (at least for myself, a person with red-green color-blindness). This process will be continued to create a plethora of color combinations such as blue-purple, brown-red, orange-yellow, red-green, etc. These color combinations will then be altered and repeated. For example, dark blue and dark purple will be used in one image a lighter tint of each color will be used in another. Issues with the devised data creation system will be explored in the Future Improvements section.

2.1.2 From Image to Data

The image data needs to be represented in a more quantitative format while still maintaining the information stored in the picture. Namely, the information to retain is the color and location of each pixel. Thus, the x and y position will be stored as well as the red (R), green (G), and blue (B) values for each pixel. Table 1 shows a few rows of the data table representing the sailboat image from figure 2.

Table 1: Data representation of sail boat picture.

x	y	R	G	B
0	0	205	205	205
1	0	205	205	205
2	0	205	205	205
		⋮		
57	116	233	184	217
		⋮		
300	300	205	205	205

With the primitive creation method used to form this data, some editing needs to be done yet. One may notice that all of the background pixels may have the exact same red, green, and blue values. At that, the foreground also only include a narrow range of colors. This is not an issue immediately, but looking forward to the color detection and generalization methods that are be used, this will cause problems. These issues and solutions to them are be expounded upon in future sections.

2.2 Color Detection

2.2.1 k-means

An unsupervised learning algorithm for pattern recognition is used to segment the image. Differences in color are detected in each image. K-means clustering [4] is being used to find distinct trends in the pixel data. Since each image is created to depict a combination of two colors, the k-means algorithm is instructed to dichotomize the data.

The Python SKLearn package was used to implement k-means clustering. The algorithm works as follows:

1. Determine how many clusters the data will be separated into. For this research, two clusters are be used.
2. Randomly select a mean for each cluster.
3. Classify each piece of data according to each mean.
4. Use these new classifications to recompute the mean of each cluster.
5. Repeat 3-4 until the clusters converge (that is, they do not change more than a threshold between iterations).

Using this algorithm on the sail boat image shown in figure 2 successfully bisects the data into clusters. The clusters have been changed to black and white to clearly illustrate the difference:

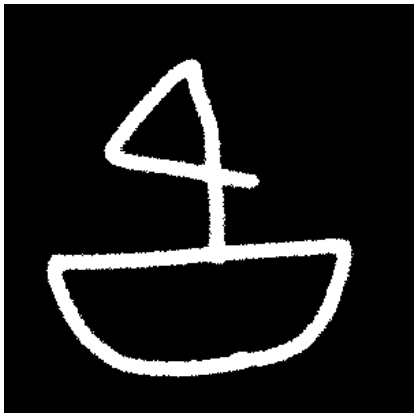


Figure 3. Image after segmentation by k-means algorithm.

It is expected that the system will identify issues with the red and green more so than the blue since it will be given data for a red-green person with red-green color blindness. Though k-means is sensitive to outliers, since multiple images are be created for each color combination, local maxima convergence will, hopefully, be avoided.

2.3 Generalization

2.3.1 Adding Background Noise

As previously mentioned, each image's foreground and background are very distinct, but in fact only have a few different RGB in each of them. For example, the background cluster of the sailboard image contains only (205, 205, 205) values in the RGB columns. If a single pixel's colors were changed, the k-means algorithm may have a hard time classifying it. However, if the colors were less uniform, the k-means algorithm could train itself on more varied, yet still distinct data. Thus random normal noise is added to each pixel's B, G, and R values. The parameters for this noise were arbitrarily chosen to be a mean of 0 and a standard deviation of 3. This these changes to the data, the image that the algorithm trains on looks as follows:

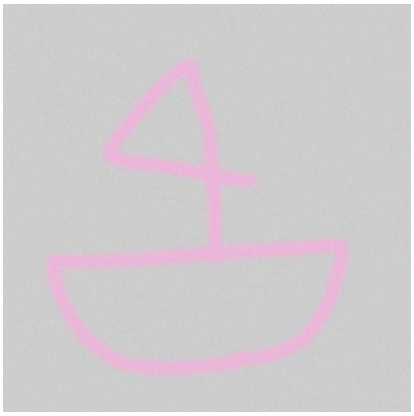


Figure 4. Normal noise added to each pixel of an image.

After running the k-means algorithm on this new data, a few cases were noticeably misclassified in this example; however, overall it did about as well.

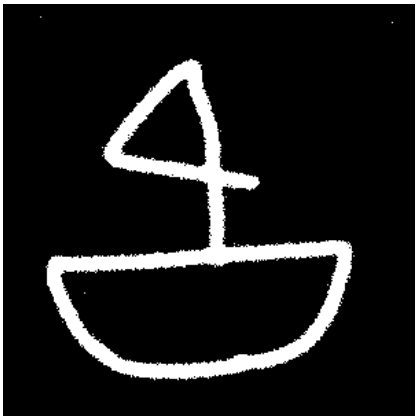


Figure 5. Clustering on data with normal noise.

2.3.2 Across Image Generalization

This idea of generalization is then brought further. Say there are two images, one with a gray background and a pink foreground and the other with a pink background and a gray foreground. In the long run, it would be desired for both of these pinks to be in recognized as the same cluster and the grays to be combined as another single cluster.

This can be achieved by stacking the data together. Only the color data of each pixel is used as input to the k-means algorithm. Thus, the data from many images can be combined together and trained on simultaneously. This offers the benefit of controlling the amount of clusters across multiple images. Furthermore, data with different clusters can be stacked.

Data with reds and greens can be combined with data with grays and pinks and the algorithm can be configured to look for 4 clusters. For the most general case, the data from all images could be stacked, and clusters can be found and combined across the whole test data set.

2.3.3 Clustering Generalization

With our current methodology, to categorize a new image, we would need to stack its data with the data from all of the other images. This is extremely computationally intensive and inefficient. Instead, we use a supervised learning algorithm to learn how to cluster.

With the clusters already known for the training data, a neural network will be trained to learn how these categories were formed. This is particularly useful for images without a binary color scheme. If the image passed has red, blue, and pink in it, the supervised neural network will categorize all of these. This is a necessary step in making the system work for real images.

2.4 Color Correction

2.4.1 Drawing a Border

One simple correction method that could be employed is through the use of edge detection. Once the system is properly trained to determine problem areas in images, an edge could be detected. Tracing this area with a thin black line would offer a visible separation between the two competing colors, while trying not to lose the information stored in the image.

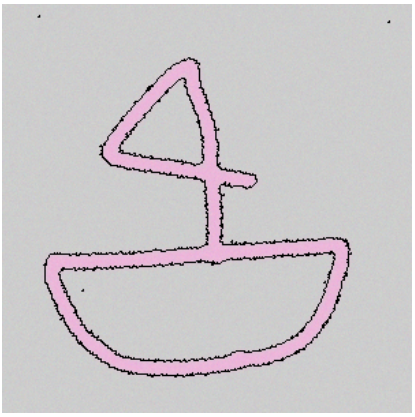


Figure 6. A thin black border is drawn between clusters.

2.4.2 Color Adjustment

Another correction procedure is to change the colors or contrast of the image itself. For example, in the sailboat data point in figure 2, the pink or the gray could be darkened. This method would leave the picture more intact as it wouldn't draw black lines, which could make the image more cumbersome to look at. However, this color changing would have to be very precise as to not completely alter the image. Changing the boat to black would assuredly make it more visible, but the image is no longer of a pink boat and a gray background. Subtly changing the color values or contrast would ensure that the image is still intact while being more accessible. With improvements to the system such as user testing this mechanism could be fine-tuned to translate an image into a user and color blind friendly picture. In this research, a very primitive approach is taken. In one cluster, the red values are increased and the greens decreased; the opposite happens in the other cluster.

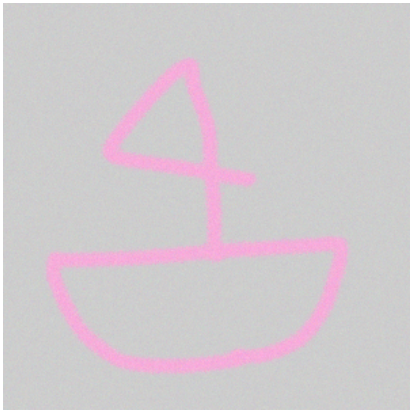


Figure 7. Cluster colors slightly adjusted.

3. Results Analysis

The output of this system is the corrected images. For the scope of this research, wide, comprehensive results are not provided. Only one person evaluates whether the objects are easier to distinguish; and indeed, they are. Peers were consulted as to how destructive the algorithm is, that is how much it alters the colors. A proper analysis would involve both groups of subjects with and without color blindness. They would examine the image before and after processing and comment on the effectiveness and destructiveness.

4. Conclusion

Overall, this research has outlined the prototype to a system for assisting people with color blindness. Images are ubiquitous on the Internet. This system provides a method for those affected to enjoy and appreciate this information as anyone else could. An ensemble of machine learning mechanisms provides color clustering; these clusters are then used to determine if there will be any potential confusion in the image and corrects accordingly. Importantly, this correction process does not destroy the image, so the message being communicated is not lost to the viewer. The process can be followed in full through the in figure 8. Further additions have been suggested throughout this paper, but there are yet more that would bring this research to greater heights.

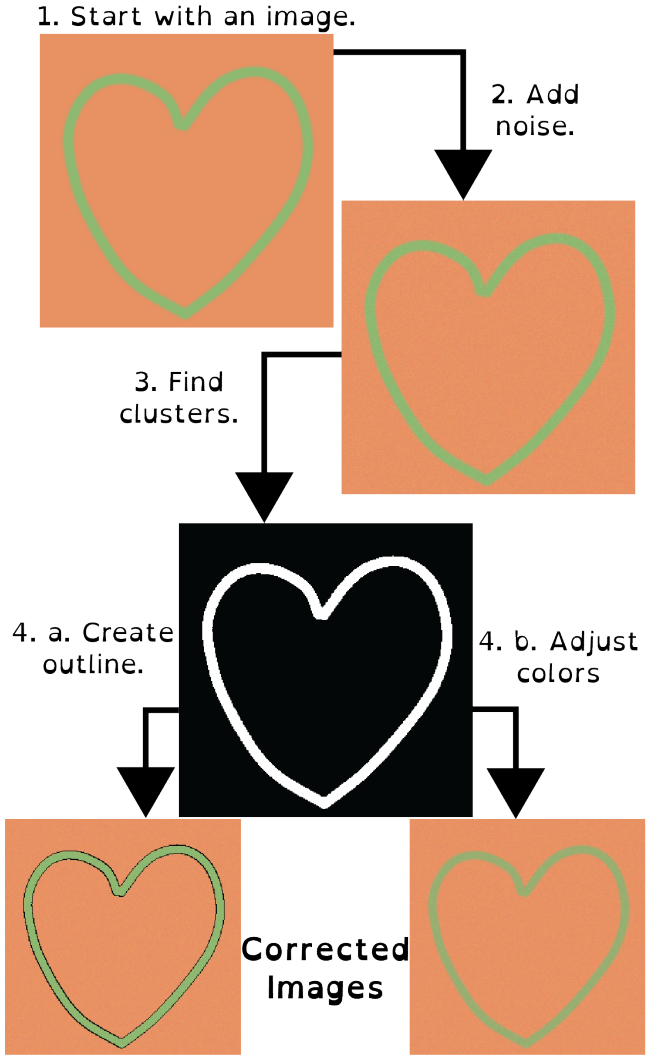


Figure 8. The entire correction process.

5. Future Improvements

5.1 Improved Data

The data is currently created by one person that is red-green color blind. This is a humble start, but everyone's color blindness is a little different. Thus, creating data that caters to only one person may leave out issues experienced by many other people with red-green and blue-yellow color deficiencies.

To solve this problem, the National Eye Institute or even an eye doctor could be inquired about the data. They could judge the data set created to see if there are any glaring cases missing from examination. Alternatively, they could provide the data themselves. This expert analysis of the problem would make the system more reliable.

5.2 Including Users in the Development

The users could help refine both the data and the correction functionality. Similar to an expert's opinion, getting feedback from many users will help create reliable data. This will ensure that no color combinations go un-learned by the system.

After the system has learned with a more inclusive data set, the users could also help judge the color correction. They could be given before and after pictures; offering their input as to whether the system actually accomplished its goal of helping them distinguish the colors in the image. Additionally, a non-color blind group of people could be brought in, again given the before and after pictures. These participants would be able to determine how drastic a change the system made to the image. Combining feedback from these two groups of people would help reach an equilibrium between making an image more accessible while not altering its meaning.

6. REFERENCES

[1] Ryan Cross. 2016. How EnChroma's Glasses Correct Color-Blindness. (June 2016). https://www.technologyreview.com/s/601782/how-enchromas-glasses-correct-color-blindness

[2] enchroma. [n. d.]. Frequently Asked Questions. ([n. d.]). http://enchroma.com/contact-us/

[3] National Eye Institute. 2015. Facts About Color Blindness. (February 2015). https://nei.nih.gov/health/color_blindness/facts_about

[4] Thales Sehn Kűrting. 2013. How K-Means algorithm works. (July 2013). https:/www.youtube.com/watch?v=IuRb3y8qKX4

[5] Emily Singer. 2009. Color-Blind Monkeys Get Full Color Vision. (September 2009). https://www.technologyreview.com/s/415339/color-blind-monkeys-get-full-color-vision/

Chaotic Random Number Generation.

Michael Holmblad
Winona State University
175 W Mark St, Winona MN 55987
Winona State University
mholmblad13@winona.edu

ABSTRACT

This paper focuses on creating and testing a pseudo random number generator (pRNG) using a branch of mathematics known as chaos theory. Generators are created based off of several chaotic systems. This paper goes through the process of how the tent map was chosen and modified to create a good rng. There has been research on the tent map as a generator, but not in this kind of sense. The tent map has had natural problems with computer hardware in the past, but those deficiencies were taken care of. This paper analyzes that creation process and tries to categorize the generator that was created.

1. INTRODUCTION

Random number generation (RNG) has been around for a very long time. Ever since a person spoke the words "heads or tails", people have been randomly generating numbers. One of the most used random number generators on the planet is a die or the usage of multiple dice. Items such as coins, dice, and similar random chance methods are known as true random number generators. (tRNG) These generators are used to pull a truly random number, and if made correctly, perform that task perfectly. However, humans have been trying to replicate this process on a computer ever since they started making video games, protecting information, and any other task that involves the usage of a random number. These generators are often known as pseudo random number generators. (pRNG) It must be asked, what is are random numbers?

Definition 1. Random numbers are numbers that occur in a sequence such that two conditions are met: (1) the values are uniformly distributed over a defined interval or set, and (2) it is impossible to predict future values based on past or present ones.

The goal of this paper is to educate about random number generators and to create a computerized tRNG, or at least come as close as possible to creating one.

2. BACKGROUND RESEARCH

2.1 Theory

Pseudo random number generation has been a discussion topic in mathematical computer science for a very long time. One book that people often refer to for information on pseudo random number generation is "The Art of Computer Programming" which is written by Donald Knuth. [3] He describes the creation of one of the most popular original pRNGs known as the Linear Congruential method, which is discussed a little later from now.
One thing about pRNGs that is talked about are their period length.

Definition 2. A *period* is the length of the string generated before it repeats a generated number.

The longer a period, the better the generator. Long is arbitrary though in this case, as in, there must be a long enough period to satisfy the needs of the generator's purpose.
Another topic that Donal Knuth talks about is seeds.

Definition 3. A *seed* is a number or vector used to initialize a pseudo random number generator.

Seeds can have an impact on how well pRNGs work. Every seed gives its own form of output. To abuse a pRNG correctly, we must choose a seed that generates the longest period, and similar seeds like it. However, in the realm of chaos theory, this isn't a problem. No seed should have a distinct advantage over another. Thus, generators must have seed choice irrelevance.

2.1.1 Linear Congruential Method

This is the most flushed out RNG in the research of RNGs. It has a very simple formula,

$$X_{n+1} = (aX_n + c) \mod m.$$

What should be noted is that X is the sequence, a is some multiplier, c is an increment, and m is a modulus to keep the numbers in a certain range. This method showed the importance of long periods and picking the correct seed. It proved that choosing a good seed can make or break the generator. It was also decided that seed choice must be avoided at this point.

3. CHAOS THEORY AND ITS ROLE

Chaos theory plays a very important role in the discovery of this new method to generate pseudo random numbers. A Chaotic Dynamical System has two major properties. According to Robert Devaney,

Definition 4. A dynamical system is *chaotic* if it satisfies three conditions: transitivity, have a dense set of periodic points, and sensitive dependence on initial conditions.

This definition is the widely accepted definition on what it means to be chaotic. The funny thing, is that this definition also defines a random number generator in some odd

way. These three conditions can be compressed into two well defined requirements.
The first of these two requirements is *sensitive dependence on initial conditions*; each point in a chaotic system is arbitrarily closely approximated by other points with significantly different future paths, or trajectories. Thus, an arbitrarily small change, or perturbation, of the current trajectory may lead to significantly different future behavior. A more simple definition is that arbitrarily close seeds create widely diverging sequences. This is key to generating random numbers because as long as the seeds are not exactly equal, they should be extremely different. Thus establishing a requirement denoted as *practical unpredictability*, given any sequence without the equation for the generator, it should be nearly impossible to predict the next term no matter how long the sequence is. Thus, if the system is chaotic, then it has sensitivity to initial conditions, and therefore has practical unpredictability.
The second of these two major requirements is that a chaotic system must have a *dense orbit*; every point in the space is approached arbitrarily closely by periodic orbits. A *periodic orbit* is a point which the system returns to after a certain number of function iterations or a certain amount of time. The dense orbit contains many different periodic orbits, thus creating long periods, seed choice irrelevance, and possibly a uniform distribution.

4. METHODOLOGY

This project is designed to take the chaotic system known as the tent map and turn it into a generator. The steps are to modify it so that the computer can preserve the chaotic properties of this mapping. Then once the generator is coded, it must be tested. There are many testing suites out there to test random number generators, however the most recognized testing suite is the dieharder test suite. The dieharder test suite is made up of three other test suites, plus more tests made by the creator of the suite.

4.1 Coding

There was a major challenge that came with producing all of the code for this project. This project utilized two different languages throughout the process. For all the preliminary setup work, python was chosen for its ability to create histograms in as little code as possible. For the generator creation and testing, C was chosen because of how close it relates to the hardware.

4.1.1 Python Theoretical Portion

During the python portion there were a couple things that needed to be done and led to python being chosen for these tasks. Python has a plethora of packages for doing simple mathematical calculations with the numpy library. On the other side of that coin, there is the robust plotting library known as seaborn, which was used to make all of the histograms in this paper.
The first system that was analyzed is known as the Logistic map. To be even more specfic, it was the logistic map of degree 4, $4x(1-x)$. It is one of the most recognized chaotic systems. After plotting a distribution for it on the interval $[0, 1]$, the logistic map was abandoned as the ulam distribution pictured below appeared.

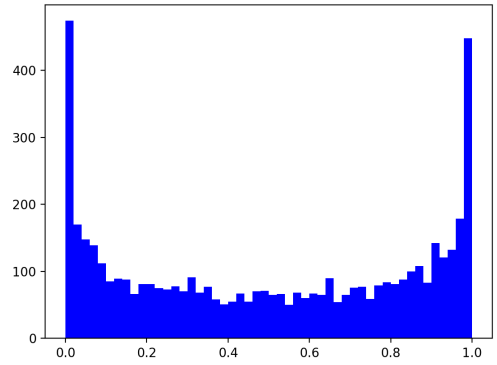


Figure 1: Ulam distribution for the logistic map

Seeing as that distribution isn't uniform, it was decided that the system must be more linear. Thus, the tent map was chosen as the next system to examine.
The tent map is as follows,

$$f(x) = \begin{cases} 2x & 0 \leq x \leq \frac{1}{2} \\ 2 - 2x & \frac{1}{2} < x \leq 1 \end{cases}.$$

Then a histogram was plotted to check to see if this distribution was uniform as well.

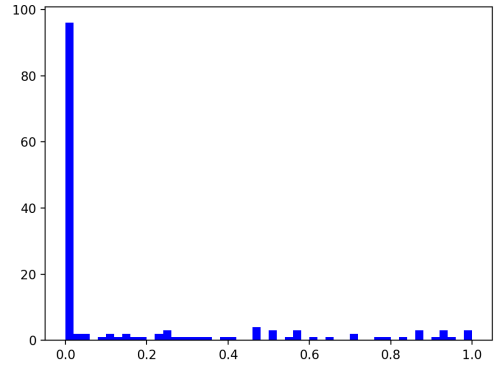


Figure 2: Distribution for the tent map after plotting 150 iterates

Seeing as this distribution had a ton of zeros, and a very balanced amount spread throughout the rest of the chart, something seemed off. It was then discovered that IEEE released a floating point standard in 1985. This standard says that floating point words are made up of 52 fraction bits, 11 exponent bits, and 1 sign bit. It just so happens that the tent map is equivalent to a bit shift map on the interval $[0, \frac{1}{2}]$. The map is also simultaneously a bit shift and bit flip map on the interval $[\frac{1}{2}, 1]$, since the map is base two, it will zero out guaranteed at the 54^{th} iterate. The standard did not change how $64 - bit$ words were organized since then. Note, that the tent map will zero out once the number fraction bits is exceeded. Thus, this is a hardware problem.

This became dubbed as the bit-loss problem. Proofs are in the appendix to show the mathematics behind this problem. The trick became solving the bit-loss problem while also preserving the properties of the tent map. The answer was simple, instead of two sections, break the tent map into three sections.

$$f(x) = \begin{cases} 3x & 0 \leq x \leq \frac{1}{3} \\ 2 - 3x & \frac{1}{3} < x \leq \frac{2}{3} \\ 3x - 2 & \frac{2}{3} < x \leq 1 \end{cases}.$$

A proof was then written to show two things, the tent 3 map is chaotic, and the tent 3 map is a ternary shift. Once these proofs were written, it the tent 3 map had to be plotted on a computer to verify the proofs.

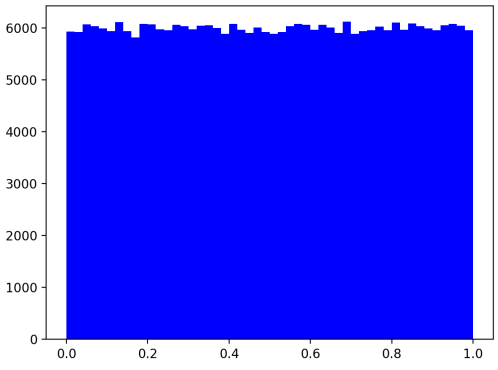


Figure 3: Tent 3 map after 300000 iterates

As can be seen by the above distribution, the system is uniform. To put everything together, since the tent map is chaotic and creates a uniform distribution all requirements are theoretically met. The map has sensitivity to initial conditions, dense orbits, and it is linear. Thus, it has practical unpredictability, seed choice irrelevance, seemingly infinite periods, and a uniform distribution. But, of course, it must be put to the test.

4.2 C Application Portion

The chaotic tent 3 map doesn't get destroyed by a computer like the original tent map. The tent 3 map hasn't been programmed as a generator before, so that must be figured out next. Thankfully, there is the general scientific library (gnu scientific library) that was written in C. This library supplies many different methods for the initialization, seed generation, tear down, and output capabilities for making a good generator.

The next thing that had to be solved is how does it get tested? The best way to test a generator is to pick a good statistical test suite. Dieharder is a statistical test suite that was created by Robert G. Brown, who is a professor at Duke University. [1] This test suite is comprised of several different test suites, which are the original diehard tests, the NIST test suite, the statistical test suite (sts), and tests made by Dr. Brown himself. Of course, the usefullness of all these different tests overlap with one another, but it is extremely useful to have all 114 different tests. To add to the thoroughness, dieharder will the tests multiple times to generate a large pool of results. The one complication here is the data type. The dieharder test suite requires a text file or a binary file full of numbers from 0 to MAX_INT. Of course, it wants $UINT\_MAX$, and the integers have to be 32-bit integers The complication then becomes a matter of converting from an integer(seed) to a float(calculations) and then back to an integer(output) again. This matter became trivial after thinking about how the architecture works. If we want to have a file full of 32-bit signed integers and the float data types are 32-bit, then seven bits of exponent are lost when switching from float back to int. The fix is to do all the calculations in 64-bit words, bits will still be "lost" converting from int to float to int, but they will be lost anyway going from 64-bit to 32-bit words. The key was to trick the computer into only returning the signed 32-bit portion of the 64-bit, then the output would be all 32-bit integers that are on the range $[0, Max\_Int]$

Needless to say, that silly fix solved all the precision problems. Just like changing the mod to solve the zero out problem, moving up an order of bit magnitude solved the storage and type conversion problem. Thus, the generator was finally able to be tested.

5. RESULTS AND ANALYSIS

Needless to say, all the dieharder tests were passed by both the mod 3 map, and by the mod 3 tent map. The results are comparable to the Mersenne Twister known as mt19937, and the AES_OFB generator. The dieharder tests outputs a bunch of p-values for each statistical test. Then they are categorized into three broad categories of passed, weak, and failed. Passed gives a value, failed returns zero, and weak returns a ridiculously high or a ridiculously low p-value

Table 1: Some Dieharder output

```
lb5304iwm71:~ lb5304iw$ dieharder -f number.txt -a -g 202
#=============================================================================#
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown          #
#=============================================================================#
   rng_name    |           filename             |rands/second|
     file_input|                     number.txt|  4.23e+06  |
#=============================================================================#
        test_name   |ntup| tsamples |psamples|  p-value |Assessment
#=============================================================================#
   diehard_birthdays|   0|       100|     100|0.22795542|  PASSED
      diehard_operm5|   0|   1000000|     100|0.83121714|  PASSED
  diehard_rank_32x32|   0|     40000|     100|0.46841587|  PASSED
    diehard_rank_6x8|   0|    100000|     100|0.72507778|  PASSED
   diehard_bitstream|   0|   2097152|     100|0.63052055|  PASSED
        diehard_opso|   0|   2097152|     100|0.98883135|  PASSED
        diehard_oqso|   0|   2097152|     100|0.97506602|  PASSED
         diehard_dna|   0|   2097152|     100|0.05787122|  PASSED
diehard_count_1s_str|   0|    256000|     100|0.93326328|  PASSED
diehard_count_1s_byt|   0|    256000|     100|0.99201710|  PASSED
 diehard_parking_lot|   0|     12000|     100|0.22401317|  PASSED
    diehard_2dsphere|   2|      8000|     100|0.65292427|  PASSED
    diehard_3dsphere|   3|      4000|     100|0.95080745|  PASSED
# The file file_input was rewound 1 times
     diehard_squeeze|   0|    100000|     100|0.88117218|  PASSED
# The file file_input was rewound 1 times
        diehard_sums|   0|       100|     100|0.41671846|  PASSED
# The file file_input was rewound 1 times
        diehard_runs|   0|    100000|     100|0.20719890|  PASSED
        diehard_runs|   0|    100000|     100|0.79383033|  PASSED
# The file file_input was rewound 1 times
       diehard_craps|   0|    200000|     100|0.29001659|  PASSED
       diehard_craps|   0|    200000|     100|0.35941143|  PASSED
# The file file_input was rewound 3 times
 marsaglia_tsang_gcd|   0|  10000000|     100|0.01464936|  PASSED
 marsaglia_tsang_gcd|   0|  10000000|     100|0.32903040|  PASSED
# The file file_input was rewound 3 times
         sts_monobit|   1|    100000|     100|0.21156827|  PASSED
# The file file_input was rewound 3 times
            sts_runs|   2|    100000|     100|0.16301673|  PASSED
# The file file_input was rewound 3 times
          sts_serial|   1|    100000|     100|0.12112171|  PASSED
          sts_serial|   2|    100000|     100|0.81128755|  PASSED
          sts_serial|   3|    100000|     100|0.98892782|  PASSED
          sts_serial|   3|    100000|     100|0.72848827|  PASSED
          sts_serial|   4|    100000|     100|0.10567349|  PASSED
```

The important thing to look at with the table on the next page, are the p-values. Not specifically analyzing how they directly compare, because they are going to come out different every time. The reason for that is the numbers are sampled from the generator and then sampled by the test, and those elements are going to change every time. The important thing to note is that a weak result every so often, and a failed every so often are a good thing. The problem is when they happen all the time. The different generators that were tested return a wide range of values, but they are widespread enough to where it isn't a serious problem. In fact, these results are assumed to be correct behavior. The results show an assortment of the average p-values wanted which is in the range of .4 to .6, outliers around .05 and .95 which are wanted a very small amount of the time, but still wanted and then every other kind of value in between.
Thus, one could argue at the moment, that this chaotic system full of mathematical bit shifts and taking the complement, is comparable to AES and the most respected Mersenne Twister. However, these generators are more well respected because of how long they have been used in industry. The AES generator and the Mersenne twister generator have been around for 10 - 20 years each. This has given them plenty of time to be repeatedly tested and repeatedly used, thus building its reputation.

6. CONCLUSION

Throughout this project there have been several discoveries that cannot go unnoticed. The first one is how data type conversions are more precise when 32 bit words are getting stored into 64 bit words when converting between floats and integers. The next major key here is how a theoretically perfect map can "zero out" on a computer due to its nature, but somehow be modified to achieve the overall goal. The last lesson is that theoretically perfect only works if there is a simple way to preserve the purity of the system in practice. Otherwise, theoretically perfect will never equal possible, until a practical method is discovered. It is still amazing that such a simple function design works this well as a generator at the time this paper is written. The final result is that for now, this generator is successful, but it must be continually tested and keep doing well.

7. ACKNOWLEDGMENTS

Special thank you to the main advisor of this project Dr. Peratt who guided this research process through to the end. Special thank you to Dr. Iyengar and Dr. Zhang, who guided and helped question the methodology and practices in this research. Special thank you to all Winona State faculty and colleagues that would listen to babbling about this project.

8. REFERENCES

[1] Brown, Robert G. *dieharder* Duke University Physics Department, 19 June 2017. Web.

[2] Devaney, Robert L. *A first course in chaotic dynamical systems: theory and experiment* Addison-Wesley, 1999. Print.

[3] Knuth, Donald *The Art of Computer Programming Volume 2: Seminumerical Algorithms* Addison-Wesley, 1968. Print.

[4] Phatak, S. C., and S. Suresh Rao. *Logistic Map: A Possible Random-Number Generator* Physical Review E, vol. 51, no. 4, 1995. Journal.

[5] Haahr, Mads *RANDOM.org* RANDOM.org, October 1998. Web.

APPENDIX

Proposition 1. The mod 2 map is a shift.

PROOF. Suppose we have a mapping function $f(x)$, where

$$f(x) = \begin{cases} 2x & 0 \leq x \leq \frac{1}{2} \\ 2x - 1 & \frac{1}{2} < x \leq 1 \end{cases}$$

where $0 \leq x \leq 1$ implies that x is always some decimal number. Therefore, since computers do there calculations in binary we may represent x in the form $\frac{d_1}{b^1} + \frac{d_2}{b^2} + \frac{d_3}{b^3} + ...$ So, all x values are of the form $\frac{d_1}{2^1} + \frac{d_2}{2^2} + \frac{d_3}{2^3} + ...$

Now suppose that $x \leq \frac{1}{2}$, then by $f(x)$ we must multiply x by 2, which looks like, $2 \cdot (\frac{d_1}{2^1} + \frac{d_2}{2^2} + \frac{d_3}{2^3} + ...) = d_1 + \frac{d_2}{2^1} + \frac{d_3}{2^2} + \frac{d_4}{2^3} + ...$ This may be called a left shift, since all of the components of x simply lost a power of 2 in the denominator. and therefore they all have a greater value.

Next, suppose that $\frac{1}{2} < x \leq 1$, then by $f(x)$ we must multiply x by 2, which looks like, $2 \cdot (\frac{d_1}{2^1} + \frac{d_2}{2^2} + \frac{d_3}{2^3} + ...) = d_1 + \frac{d_2}{2^1} + \frac{d_3}{2^2} + \frac{d_4}{2^3} + ...$ But, this time we subtract 1 as well, because $d_1 + \frac{d_2}{2^1} + \frac{d_3}{2^2} + \frac{d_4}{2^3} + ... > 1$. Which obtains, $\frac{d_2}{2^1} + \frac{d_3}{2^2} + \frac{d_4}{2^3} + ...$ This may be called a left shift, since all of the components of x simply lost a power of 2 in the denominator. and therefore they all have a greater value. However, we also cut off the leading term so that we stay inside the range of the map, but it is still a left shift.

Thus, the mod 2 map is a shift map.

Corollary 1. The mod 2 map and the tent map will always zero out after 54 iterations.

PROOF. In 1985, the Institute of Electrical Electronic Engineers published IEEE 754 (Standard of Floating Point Arithmetic) in 1985. Thus, on all modern computers every 64-bit floating point number is divided up into 53 coefficient bits, 11 exponent bits, and 1 sign bit.

By Proposition 1, we know that the mod 2 map and the tent map knick a term off the front, then recalculates. Since, its in binary, that bit. After 54 iterations of this we only return 0, since the floating point coefficient bits have overflowed with zeros.

Thus, the mod 2 map and the tent map zero out on computers.

CSS Grid for Responsive Web Applications

Lucas Credie Nogueira de Lima
Winona State University
460 W Mark Street, Winona, MN, 55987
Rua Florida, 71 apt 104. São Paulo, Brasil
+1 (507) 858-5128
LCredieNoguei13@winona.edu

ABSTRACT

This project involves analyzing the current technologies that are used to create responsive applications in the web and aims to test the newest option, CSS Grid, as a potentially more powerful and efficient option for building layouts. In order to analyze both technologies from a developer's point of view, two equal landing pages were created, one using Bootstrap and another one Grid. To avoid biases, the original landing page design was created using pen and paper and then digitally illustrated. After having developed both versions, CSS Grid has proven to be the best layout and responsive design tool by having more features and being able to separate content from style more efficiently than Bootstrap, which improves maintainability and allows for more flexibility when dealing with user interface changes over the years.

Keywords

Web Development, Responsive, Design,

1. INTRODUCTION

When the world wide web first came to be, no one expected that it would grow to be one of the most important environments in software development. This growth forced the initial technologies used to adapt to the challenges we faced over the years. The main challenge discussed in this document is the creation of responsive applications, which consists of creating applications with layouts that adapt to different screen sizes without the need to build multiple versions of the same application.

To understand the nature of the subject discussed in this document, it is necessary to understand that HTML (Hypertext Markup Language) is what composes the structure of a web page and content that it holds while CSS (Cascading Style Sheets) define the stylistic properties of that structure. A framework (such as Bootstrap) is a collection of pre-made styles and properties that can be applied to HTML.

Proceedings of the 18th Winona Computer Science Undergraduate Research Seminar, April 23 & 25, 2018, Winona, MN, US.

Before Responsiveness was a concern, the first challenge was to build layouts and one of the first ways used to create layouts on the web was by the use of HTML tables in the 90s [1]. There the whole page would be one table where specific cells would be used for specific content or component and although it worked, the extensive markup, slow load time and visual inconsistencies [1] made tables a problematic option.

By the year 2001 Internet Explorer 6 was released [2] and that allowed for CSS to start gaining momentum. CSS is mainly used to give a web page it's appearance, making possible for content and style to be separate and by then the use of relative and absolute positioning [2] was used as a primitive grid that served as foundation for many features we use today.

The next revolution that influenced web development was Smartphones. Since Steve Jobs unveiled the iPhone in 2007, smartphones grew to be the main way people use the web and in 2016 it is estimated that 77% of Americans own one [3]. The amount of different devices with different screen sizes, operating systems and browsers made it so responsive design was complicated and daunting process that many developers dreaded.

Many tactics and workarounds were used to build responsive design and along the years many CSS frameworks, like Bootstrap and Foundation, were created to accomplish responsive design. Although frameworks work, they offer new challenges in maintainability and flexibility to layout changes because they add unnecessary code to the structure and content. It is good to note that those frameworks are not only used for responsive layouts but to give developers the ability to create a decent looking application with no graphic design skills, so they offer value in many different ways and might still be a good option today.

The newest option available is CSS Grid Layout, which is a new way to layout and build responsive application with pure CSS, which is a different approach to development compared to bootstrap and foundation because it separates style from content and markup. This is of extreme importance because it affects how easy the web application is to maintain and how flexible it is when changes need to be done over the years. For those reasons, I hypothesize that CSS Grid is an easier and more flexible way to build responsive web applications compared to Bootstrap's grid system.

2. METHODOLOGY

To test the hypothesis, a landing page for a fictional organization was designed and developed. What was being tested in this experiment is how CSS Grid keeps content and markup in the HTML code separate from style and how it's features can allow developers to create layouts that are not possible in contrast with Bootstrap.

2.1 Design

To avoid biases, the landing page was created using pen and paper and later recreated digitally so it could be better presented.

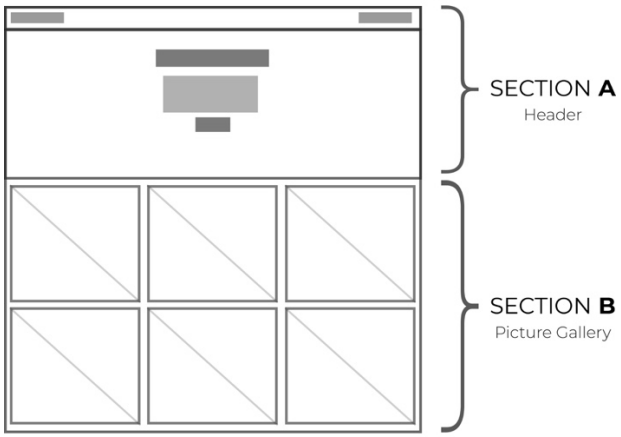


Figure 1. Digital Wireframe created from original drawing.

The page was divided in two sections, header and picture gallery, which can be viewed in Figure 1. The list of components for each section is displayed in table 1 bellow.

Table 1. Components necessary for the creation of the layout. Subcomponents are in this scenario components that don't influence layout or its responsiveness.

Component	Section	Subcomponents
Navigation Bar	A	Logo, Back Link
Header	A	Title, Paragraph, Button
Pictures	B	none

The section used to test the responsive design features is Section B, the picture gallery. At full size the gallery will display six pictures in a 3x2 fashion, at medium size 2x3 and at small size 1x6. This will be discussed with more detail in section 4.3.

2.2 Technology Background: Grid Systems

Grid systems are used to specify where items in a page will live and responsiveness is where the items live at a specific screen width.

The usual way this is done is by having the page divided into a number of equal columns and then specifying how many of those columns should an items length take, having its height defined separately and the order of appearance defined by the order they appear in HTML. The way responsiveness is achieved is done by specifying how many columns the item takes at a specific screen size, and this principle remains the same for both Grid and Bootstrap.

Since Bootstrap is a framework, it is necessary to download the specific CSS files in order to use them. After importing the files, the premade classes are ready to be used with the HTML. The grid system in Bootstrap is made up of twelve columns and to place an element in the grid it is necessary to add extra HTML tags to define a container of rows and more html for each row as well as the classes for the items.

With CSS Grid, there is no files to be downloaded and all you need is a parent HTML element and set it's display property to "grid". Grid allows the developers to set any number of columns and rows and the placement of items in the grid is done in the same CSS file without the need of any extra html markup.

2.3 Software Development

Before the implementation of the landing page I chose that the website was going to be a landing page for a fictional pizzeria called "Amore Mio" and after selecting a few free to use pictures the development process could start.

The first step was to set the folder structure for both versions. Each folder being named correctly and containing an HTML and CSS file each. After the set up was done the next step was to implement the CSS Grid version.

2.3.1 Software Development: CSS Grid

When using CSS Grid, the developer has the freedom to create any number of columns but in this specific implementation, I decided to create twelve columns so it would be the same as Bootstrap and better compared.

This was done by simply setting the parent element, which in this case was the body, to "display: grid;" and then adding "grid-template-columns: repeat(12, 1fr);" as a property. This property creates 12 columns of 1 equal fraction of width each. Note that CSS Grid's columns don't need to be the same width. I also included the "grid-template-rows: 60px 450px;" being the first row the navigation bar with 60px of height and the 450px the header component (see Table 1). This could also be accomplished by going to the specific components and setting their height there.

Setting items turned out to be easier than setting up the grid. To place the navigation bar, that spans across all columns, all that needed to be done was add "grid-column-start: 1;" and "grid-column-end: 13;" which could also be written with only one line "grid-column: 1/13;". The reason why that is 13 in a 12 column grid

is because, differently from Bootstrap, CSS Grid uses lines of the columns as references instead of the columns themselves.

Given those concepts, the rest of the CSS Grid development involved setting up another grid for the Section B on a HTML main tag and placing the pictures inside. This was done by setting the "grid-column-start" property of every picture to "span 4" meaning that every picture should take 4 columns out of 12, giving us an 3x2 grid by default.

To achieve responsiveness with CSS Grid, you can use all of those features and others, discussed later in this document, and combine them with media queries. For this project all I needed to do was set up 2 media queries. One where the pictures would "span 6" when the browser's screen size is smaller than 750px and another where pictures would "span 12" when screen size is lower than 500px.

To finish this implementation, the gutter was created by adding "grid-gap: 15px;" in the Section B parent, which is a feature that allows you to define gaps between rows and columns in only one line of code. Refer to Figure 2 for the final result of this implementation.

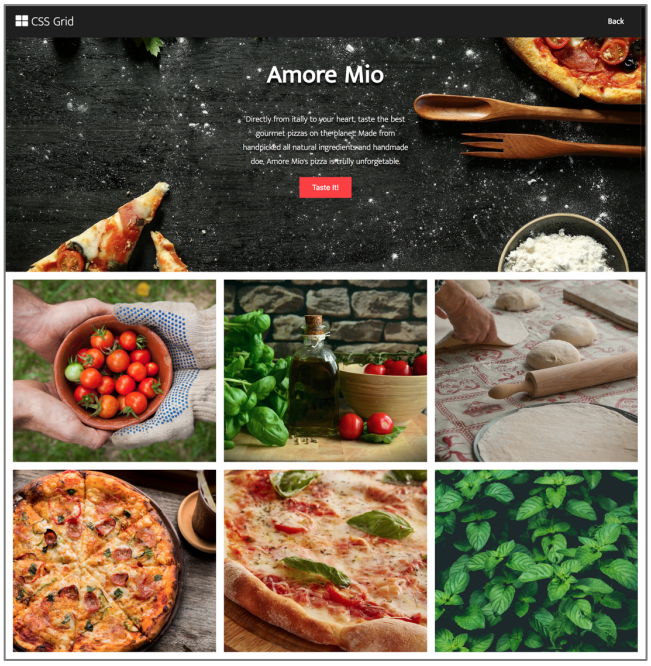


Figure 2. Final CSS Grid implementation

2.3.2 Software Development: Bootstrap

Differently than with CSS Grid, Bootstrap requires the developer to shift his focus to HTML instead of CSS and development is fairly quick once you have all the Bootstrap files imported.

Since the grid in Bootstrap is already defined to be 12 columns, all you need to do is add specific classes to HTML elements to position and make elements responsive. This process is fairly quick for developers that are familiar with the Framework and its speed of implementation is one of the main reasons why Bootstrap is so popular.

To position the header component so it would span all 12 columns, the class "col-lg-12" is added into the header HTML tag. That specifies that the element is a "column" element and it takes 12 columns when large which is really convenient. In Bootstrap the sizes are predetermined and large means 992px and up [4].

A common thing when using Bootstrap is that extra HTML Markup needs to be added in order for the functionalities to work. When creating Section B, which is the main responsive component, the main parent takes in a "container" class and an extra division tag needs to be added for every row in the grid. After that every item inside the row requires 3 classes in order to be responsive in 3 different screen sizes. In the case of the pictures the classes used were "col-lg-4 col-md-6 col-sm-12" to create the responsiveness. This is very quick to set up but also very redundant.

The main challenge that I ended up having with this implementation was the gutter system between columns and rows. Adding paddings, margins or borders to would cause the columns to have double gap. So to be able to create an even grid gap that is responsive with Bootstrap you would have to add media queries to change to settings on individual pictures in their different screen sizes. But what if the pictures are being read from a database? There is simply no work around. The solution to that was to add borders, which made the layout look similar to what was originally intended in the wireframe design but not perfect. Notice in Figure 3 how the gutter is uneven on the sides and top.

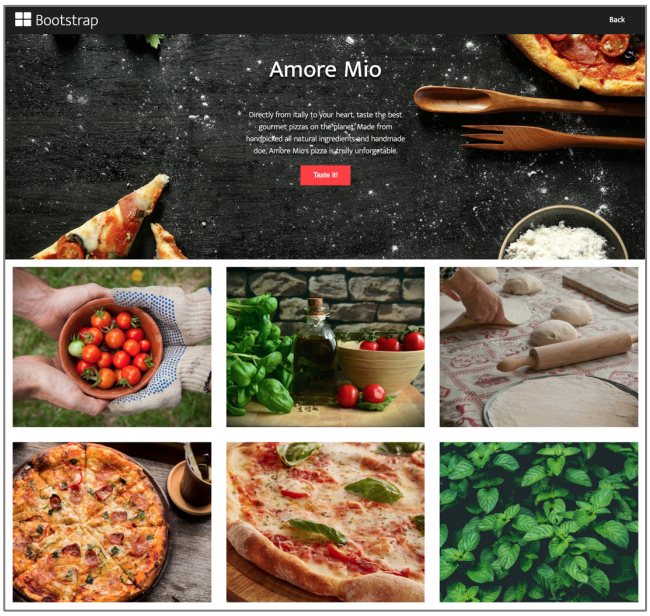


Figure 3. Final Bootstrap implementation

3. ANALYSIS

After both versions of the page were built, there was no significant differences in development time (Both took around 2 hours and 30 minutes). That said, it is significantly easier to get a grid project started since there is no need to import any files.

For the summarized comparison of main features and benefits refer to Table 2.

Table 2. Features and Benefits Comparison showing that CSS Grid allows for more customization and flexibility to create complex layouts

Feature	CSS Grid	Bootstrap
Media Queries Independent		✓
In built Gutter	✓	
Automatic rows	✓	
Custom column count	✓	
Vertical Span	✓	
Custom column widths	✓	
Dependency free	✓	
Content reordering	✓	

When comparing both technologies, the main advantage of using the Bootstrap Grid system is that you don't need to think about setting media queries and break points. With this freedom comes the price of less customization but that is one of the reasons why Bootstrap is so popular. As far as CSS Grid goes, there are many benefits of using it as a layout tool compared to bootstrap.

The gutter system I found to be the most obvious one. It was nearly impossible to recreate a gutter system that is responsive with Bootstrap alone and having it inbuilt in grid within one line of code is really convenient.

My second point in favor of Grid is that compared to Bootstrap, CSS Grid is highly customizable. If you wanted to create a 15 column layout in Bootstrap you could as a developer go into its CSS file and edit its classes but the extra functionality you would get out if it is not worth going through around 9 thousand lines of CSS Code (Bootstrap V4) and risk creating unexpected or broken behavior of other features. Grid also allows the developer to create different sizes columns and rows, which is a feature that doesn't exist anywhere else.

The most important feature I found it was the way CSS Grid handles rows. Not only can you specify row sizes but also have it as automatic handling. This at first sight doesn't seem that important but it plays an important role in how it reduces the amount of extra HTML tags and classes needed to create responsive layouts. In a big project this feature can help improve maintainability.

4. CONCLUSION

As a layout tool, CSS Grid is superior than Bootstrap. The amount of features allow for very customizable and easily changeable layouts and combined with media queries it can make designing responsive applications very fun and efficient, making possible to create layouts and user experiences that were not possible before.

The only downside of CSS Grid is that changing the order of content with CSS can cause accessibility problems for users using screen readers, so that is something to consider when improving upon this technology.

When it comes to making a decision on which one to use, the conclusion is that both Grid and Bootstrap have a place in development, but those places are different. CSS Grid might be superior when setting up complex layouts but Bootstrap still offers developers a quick way to set styles and allows for developers to create a fairly good looking application quickly. So if the project doesn't require unique design and the objective is to build something really fast, bootstrap might be a good option but for serious web development where a design team is involved with development, CSS Grid is the best bet for creating any layout imaginable.

5. ACKNOWLEDGMENTS

I thank Dr. Mingrui Zhang, Dr. Sudharsan Iyengar and Dr. Joan Francioni for all the patience and guidance and all my Computer Science Research Seminar classmates for the help and inspiration to keep moving forward.

6. REFERENCES

[1] Myia Kelly. A Look Back at 20+ Years of Website Design. Published date: July 10, 2013. https://blog.hubspot.com/marketing/look-back-20-years-website-design. Accessed on: April 2, 2018.

[2] Jason Hoffman. "A Look Back at the History of CSS". Published date: October 18, 2017. https://css-tricks.com/look-back-history-css/. Accessed on: April 2, 2018.

[3] Rachel Sandler. "The IPhone's Smartphone revolution in 4 graphs". Published date: June 28, 2017. https://www.usatoday.com/story/tech/news/2017/06/28/iphones-smartphone-revolution-4-graphs/103216746/. Accessed on: April 2, 2018.

[4] Grid System. https://getbootstrap.com/docs/4.0/layout/grid/, Accessed on: 16 January 2018.

Computer Security Documentation for a Non-Technical Audience

Salman Quraishi
SQuraishi13@winona.edu
Computer Science Department
Winona State University
4/30/18

ABSTRACT

This research project investigates the development of computer security documentation. Computer security includes the protection of hardware, software, and/or digital information from theft and/or damage, along with preventing disruption or misdirection of the services a computer may provide. The scope of the research was to develop a document detailing the basic fundamental concepts of computer security that individuals of all backgrounds can use without requiring prerequisite knowledge of computer security. The core concepts of the document encompass phishing, social engineering, password security, trusted/untrusted networks, viruses, malware, and antivirus software. Both technical and non-technical individuals can utilize the document to learn how to be safe online and make informed decisions on the internet. The evaluation of the efficacy of the document relies on two surveys. The methodology involved participants taking a pre-survey before reading the document. Followed by reading the developed document, and afterward taking a post-survey. The results of the first survey in comparison to the second survey are used to determine the efficacy. The participants are selected on a voluntary basis, with the focus being on non-technical individuals of varying backgrounds.

General Terms & Keywords

Computer Security, Phishing, Social Engineering, Document, Documentation, Survey, Passwords, Virus, Malware, Antivirus.

1. INTRODUCTION

In a modern world where technology plays a significant part in our daily lives and individuals from all occupations are using computers, computer security is becoming a growing issue. Considering the impact of good security decisions and the repercussions of bad security decisions, it is important that all individuals, technical and non-technical individuals alike, have an understanding of the core concepts behind computer security. Staying safe in a connected digital world requires a basic fundamental understanding of a variety of computer security concepts.

The dissemination of such information would assist in enabling individuals to make safer decisions with computer security in mind are not available as one comprehensive document. Much of the information regarding the topic are scattered throughout a plethora of documents and often use language that is difficult to understand without a considerable prior knowledge of the topic. As such, the need for a document written in colloquial, simple language that is understandable without prerequisite knowledge of computer security is necessary. Therefore, the primary motivation for this research is due to a lack of a single comprehensive document geared toward a non-technical audience detailing the fundamental concepts of computer security

The primary objective of the research was to develop such a document covering the fundamental concepts of computer security as comprehensively as possible. Understandably, not all aspects of computer security can be covered in a single document. Hence, this document aims at covering the most of the basic concepts related to computer security. This document acts as a primer to computer security briefing the reader on several concepts of computer security, rather than as a complete textbook on computer security. This document was created with the intention of disseminating the concepts as briefly and succinctly as possible without losing the value of the information itself. This also creates an opportunity for future revisions and additions to the document.

For a non-technical community, information on the basics of computer security will enable individuals to take certain precautions, make well-informed decisions, and prevent potentially disastrous damages. The scope of this research is for a non-technical audience and therefore does not include technical strategies for developing security, rather it is to inform about the basics of computer security and making safer decisions regarding it.

2. HYPOTHESIS:

The developed document detailing the basics of computer security increased how informed individuals were regarding computer security based on a pre-survey followed by a post-questionnaire.

3. BACKGROUND RESEARCH

The topics covered in the document are those that have been determined to be the most important and of widespread concern today. The determining as to which concepts may be of importance has been done via consultation with Dr. Gerald Cichanowski. The topics of widespread concern have been determined by considering their impact (cost and damage) and the common consensus of security organizations. The organization of the developed document follows how US-CERT organizes their tips for computer security [4]. The main areas of focus of the document are phishing, social engineering, identity theft, password security, trusted/untrusted networks, and viruses/malware. These security concerns are widespread and the victims are often those with less technical or technological experience.

3.1: PHISHING

Phishing is one of the security issues that has only grown in number over the years. According to research by the Friedrich-Alexander University, 76% of people reported that are aware of the risks of clicking on an unknown link but clicked it anyway [1]. While this university is in Germany and one might think not is representative of the United States. The situation is no different in the US, as the Computer Crime and Intellectual Property Section (CCIPS) reported that have been “more than 4,000 ransomware attacks [that] occurred every day since the beginning of 2016” [2]. This is in comparison to roughly 1000 attacks that occurred every day in 2015

[2]. While CCIPS stats for 2017 are not yet available, the trend seems to be continuing to grow. According to Verizon's 2016 Data Breach Investigations Report, "30% of phishing emails are opened. And about 12% of targets go on to click the link or attachment." [3]. Therefore phishing is of consequential importance and will continue to increase, which is why it is more important than ever for individuals to be informed about phishing. Phishing in general, different types of phishing, and how to avoid being phished is the crux of what the document covers regarding phishing.

3.2: PASSWORD SECURITY

Additionally, passwords and password security is covered in the document, followed by the importance of creating strong passwords and what constitutes a strong password. The aforementioned Data Breach Investigations Report by Verizon in 2016 also found that "63% of confirmed data breaches leverage a weak, default, or stolen password" [3]. This is unacceptable, it is very important for individuals to be serious about password security, which is why the developed document discusses password security and strategies for developing a strong password.

3.3: TRUSTED & UNTRUSTED NETWORKS

The basics of trusted and untrusted networks, how to identify the credibility of a network, and the dos and don'ts of public Wi-Fi are also covered in the developed document. The motivation for discussing untrusted networks is due to the security concerns it poses. This does not just include public Wi-Fi, like coffee shops and restaurants but also a number of other unsecured networks. Even with hotel Wi-Fi, while there may be a password, the password is shared and anyone staying in that hotel has access to the network. This opens up a number of other security issues that could potentially occur.

3.4: ANTIVIRUS

The developed document also briefly compares antiviruses. The primary reference for comparing the antiviruses was AV-Comparatives. AV-Comparatives is one of the most reliable and independent sources that do periodic testing of antiviruses and publishes the findings online. They are also unaffiliated with any particular antivirus. AV-Comparatives conducts a variety of tests ranging from real-world protection, malware protection, and performance. The only test that the developed document encompasses is AV-Comparatives' Real-World Protection Test Results for July-November of 2017 [9]. Their research involved 1769 test cases tested across several antivirus software programs. According to their findings for that test period and test cases, Panda was the only software that had a 100% protection rate. Bitdefender, Trend-Micro, Symantec, and Kaspersky were not far behind but only Panda provided 100% protection rate. However, it is important to note that this may vary in future tests, and no one antivirus is perfect. The recommendations were given accordingly in the developed document, taking into consideration price, performance, and availability.

3.5: CONCEPT SELECTION

The document is 10 pages long and includes all of the aforementioned topics. The document covers what the reader needs to know, why, and where to find additional information. The document also includes tips regarding computer security in general.

4. METHODOLOGY

The methods used to assess the validity of my hypotheses included surveying the internet, academic journals, and past expert research. The research consisted of 10 individuals, who were given a pre-survey, a security document, and a post-questionnaire. The core of the research revolved around the development of a document detailing the fundamental concepts of computer security. Additionally, the research involved the development of a pre-survey to gauge the level of information the reader had prior to the security document. Lastly, the research involved the development of a post-questionnaire to gauge the effectiveness of the security document and if there was an increase in the individual's information on the subject. First individuals completed the pre-survey, subsequently individuals read the developed security document, and lastly, individuals completed a questionnaire after reading the security document [*refer to Figure 0]*. This pre-survey compared to the final questionnaire determines the effectiveness of the main document.

THREE-PART METHODOLOGY

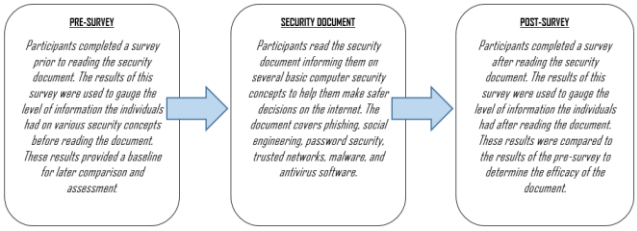


Figure 0: Three-Part Methodology

The assessment, meaning the actual comparison between the performances of the participants, was a combination of assessing correct answers as well as the opinion of participants. While opinion may vary, it is vital that the value such a document may pose to the target audience takes into consideration their opinion. However, opinion alone cannot satisfactorily assess the efficacy of the document. Therefore, short-answer questions were used to gauge the information of the participants. A combination of two types of results were used to determine the results. As a baseline, if 60% (6/10) of the participants or more had a positive result, the document was deemed to be efficient in informing the given population sample about that particular concept. It is noteworthy to mention that all results are representative of the given population and experiment parameters.

5. DEMOGRAPHIC OF PARTICIPANTS

The population sample consisted of a largely non-technical audience of college students. The students were selected from classes that were either non-CS related or lower-level CS courses [*refer to Figure 1*]. The inclusion of CS students with some prior security knowledge ensured the population had a mixture of individuals with varying levels of information [*refer to Figure 1 & 2]*. Hence, the pre-survey was able to capture this information, which would provide a more accurate depiction of the overall performance. Due to a small population size, the information was organized using first and last names, however, all information was anonymized in the aggregate result.

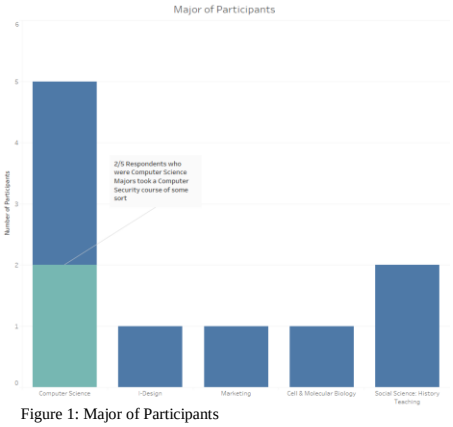


Figure 1: Major of Participants

6. RESULTS

The results were largely indicative that the developed document was indeed effective for the given population and circumstances. However, to guarantee the universal efficacy of the developed document, additional refining of the document, surveys, and a larger population sample would be required, which will be considered in future work.

The first indication that the document was effective for the given population, is the participants giving more definitive answers. One such example is participants providing a definition of what they think is a strong password.

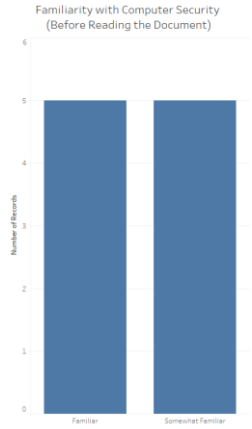


Figure 2: Familiarity with Computer Security (Pre-Survey)

Prior to reading the document, of the 10 participants, 5 participants responded they were "familiar" and 5 responded that they were "somewhat familiar" with computer security in general [*refer to Figure 2]*. After reading the document, 3 individuals felt they were "very familiar" whereas they had responded that they felt only "familiar" prior to reading the document. Additionally, 4 of the individuals who had responded that they felt "somewhat familiar" in the pre-survey, said that after reading the document they now felt "familiar" with computer security [*refer to Figure 3]*.

Hence, the document assisted the given population in becoming familiar with computer security and its basic concepts.

Prior to reading the document, in the pre-survey, 2 participants said they were "barely familiar" with phishing, 2 responded they were "not familiar", 1 responded they were "somewhat familiar", and 5 said they were "very familiar" [*refer to Figure 4]*.

After reading the document, 8/10 participants responded they felt "very familiar" and 2 said they felt "somewhat familiar" [*refer to Figure 5]*. Hence, the document assisted the given population in becoming familiar with what phishing is.

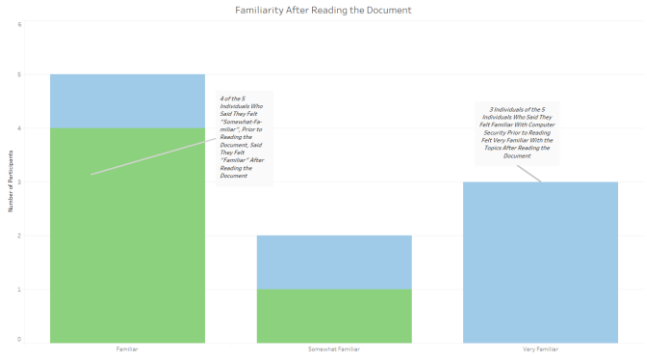


Figure 3: Familiarity with Computer Security (Post-Survey)

The participants' opinion of the document, albeit relative, is also indicative of the effectiveness of the document considering 7 out of 10 participants responded that the document covered the topics "very well", and the remaining 3 participants responded it covered the topics "well" [*refer to Figure 6]*. Although not quantitative, the opinion of the participants is also indicative that the document was indeed successful for the given population.

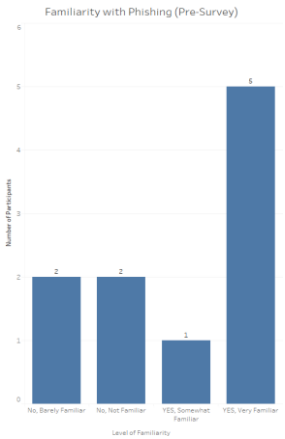


Figure 4: Familiarity with Phishing (Pre-Survey)

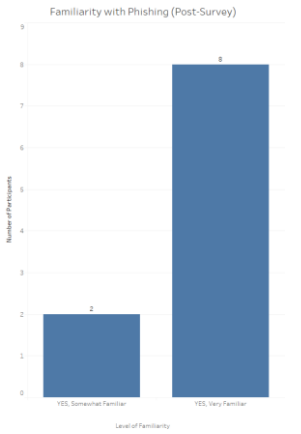


Figure 5: Familiarity with Phishing (Post-Survey)

Given multiple options, participants were asked to select the options that were an example of a phishing attempt; only one participant was able to correctly answer the question in the pre-survey. All other participants were only able to accurately identify some of the attempts.

After reading the document, 5 individuals were able to identify all the phishing attempts, 1 individual was still unable to identify any of the attempts, and the rest were able to identify most of the attempts.

Hence, considering that 5 of the participants were able to correctly answer the question after reading the document, whereas only 1 participant was able to do so prior, the document was effective in informing the given population about identifying potential phishing attempts.

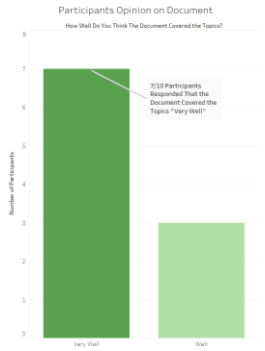


Figure 6: Participant Opinion on Document

The results are indicative that the document was not only successful in assisting the given population in identifying potential phishing attempts, but also indicative that reading the document increased how confident participants were in their ability to differentiate between a phishing attempt or not [*refer to Figure 7]*. Therefore, the document increased how confident the given population was in recognizing a phishing attempt, and increased the accuracy of identifying the phishing attempt(s).

Figure 7: Participant Confidence Regarding Recognizing Phishing Attempt

The participants, after reading the document, were able to provide correct answers to questions they were not able to answer correctly in the pre-survey.

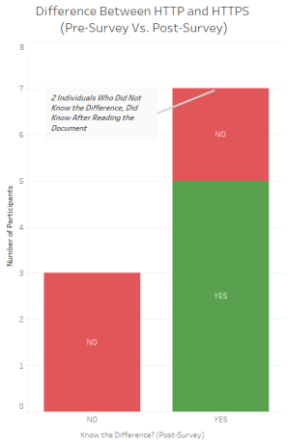


Figure 8: HTTP vs. HTTPS

Although the document could have been more effective in explaining the difference between HTTP and HTTPS, 2 individuals who did not know the difference, between the HTTP and HTTPS, did know after reading the document. However, unfortunately 3 individuals still had difficulty in explaining the difference between HTTP and HTTPS [*refer to Figure 8]*.

The aforementioned ideas were the critical areas of assessment, and in all of which, reading the document increased the accuracy of answers, familiarity with the concepts, and a better understanding of topics the given population was unfamiliar with prior.

7. CONCLUSION

For the given population and parameters, the document was effective in informing the participants and familiarizing them with basic concepts of computer security. The results of the pre-survey in comparison to the post-survey enabled the assessment of the documents efficacy. The document was developed with the intention of increasing information regarding computer security. Individuals of all backgrounds should be familiar with computer security. In a growing digital world and the age of the internet, technology is part and parcel of everything we do, and we must be secure in the process. There is hope that ventures such as this research open a more technical dialogue in all facets of society, in business, in politics, in science, and all other industries.

8. FUTURE WORK

Future work would entail refining, editing, and expanding the developed document. Future work would also involve re-engineering the surveys based on the effectiveness of the questions and user feedback. Any future work will ideally incorporate a larger population sample, meaning more participants. More participants from a variety of backgrounds would provide a better assessment of the document. Future work would also concentrate on a population that is non-technical but also in the professional workplace not just college students. Future revisions of the document would be more applicable to technical individuals as well as non-technical.

9. ACKNOWLEDGMENTS:

Special Thanks to Professor Dr. Gerald Cichanowski and Advisors Dr. Sudharsan Iyengar and Dr. Mingrui Zhang.

10. REFERENCES

[1] FAU. “FAU Researchers Investigate User Behaviour When Unknown Messages Are Received Online.” *Friedrich-Alexander-University Erlangen-Nürnberg*, FAU, Friedrich-Alexander-University , 25 Aug. 2016,www.fau.eu/2016/08/25/news/research/one-in-two-users-click-on-links-from-unknown-senders/.

[2] United States, Congress, “How to Protect Your Networks from Ransomware: Technical Guidance Document.” *How to Protect Your Networks from Ransomware: Technical Guidance Document*, Computer Crime and Intellectual Property Section (CCIPS), United States Department of Justice. www.justice.gov/criminal-ccips/file/872771/download.

[3] “2017 DBIR: Understand Your Cybersecurity Threats.” *Verizon’s 2017 Data Breach Investigations*

Report, Verizon Enterprise Solutions, 2 May 2017, www.verizonenterprise.com/verizon-insights-lab/dbir/2017/.

[4] "US-CERT Tips." US-CERT: Tips, United States Computer Emergency Readiness Team, Department of Homeland Security. www.us-cert.gov/ncas/tips/.

[5] Gasser, Morrie. *Building a Secure Computer System*. Reinhold, 1988. University of Nebraska at Omaha. cs2.ist.unomaha.edu/~stanw/gasserbook.pdf.

[6] Schatz, Daniel; Bashroush, Rabih; and Wall, Julie (2017) "Towards a More Representative Definition of Cyber Security," Journal of Digital Forensics, Security and Law: Vol. 12: No. 2 , Article 8. http://commons.erau.edu/jdfsl/vol12/iss2/8

[7] Zlanatov, Nikola. "Tech Security Conference." Computer Security and Mobile Security Challenges, 2015, www.researchgate.net/publication/298807979_Computer_Security_and_Mobile_Security_Challenges.

[8] "Whole Product Dynamic "Real-World" Protection Test: July-November 2017." *AV-Comparatives: Independent Tests of Antivirus Software*, AV-Comparatives, 12 Dec. 2017, www.av-comparatives.org/wpcontent/uploads/2017/12/avc_prot_2017b_en.pdf.

[9] Morgan, Steve. "Top 5 Cybersecurity Facts, Figures and Statistics for 2018." *CSO Online*, CSO, 23 Jan. 2018, https://www.csoonline.com/article/3153707/security/top-5-cybersecurity-facts-figures-and-statistics.html.

[10] Crowe, Jonathan. "Must-Know Phishing Statistics 2017." *Barkly Endpoint Security Blog*, Barkly, July 2017, https://blog.barkly.com/phishing-statistics-2017.

[11] "Enterprise Phishing Resiliency and Defense Report 2017: Analysis of Susceptibility, Resiliency And Defense Against Simulated And Real Phishing Attacks." *Cofense*, PhishMe Inc., 2017,https://cofense.com/wp-content/uploads/2017/11/Enterprise-Phishing-Resiliency-and-Defense-Report-2017.pdf.

[12] MIT. Top Ten Safe Computing Tips. Retrieved April 19, 2018 from https://ist.mit.edu/security/tips

[13] "What Is the Difference: Viruses, Worms, Trojans, and Bots?" *Cisco*, Cisco Systems Inc., 20 Nov. 2017, www.cisco.com/c/en/us/about/security-center/virus-differences.html.