

Capture The Cache
Engineer: Shawn Leedy
Project Sponsor: Dr. Deborah Hwang
Project Adviser: Dr. Deborah Hwang
Bachelor of Computer Science
University of Evansville
05/02/2019

ABSTRACT

The United States has a problem with people not getting enough physical activity. This project is an Android game based on Geocaching, where instead of searching for physical objects in the world, players will instead have to be physically active in order to find digital caches, encouraging physical activity to counteract this problem.

INTRODUCTION

The goal of this project was to create a phone game which will encourage the user to go outside and interact with the meta layer of the world, which is handled by the game itself.

Google Maps renders the local area to give players an idea of where they are and where the nearest caches are, and the device sends objects back and forth with the server to know when to create or delete caches.

PROBLEM STATEMENT AND BACKGROUND

There is an epidemic in the world today, where only 35~44% of adults are physically active, and there are a total of 8 million people aged six or older who are physically inactive[1]. A certain genre of mobile games, known as Augmented Reality(AR) games, have been working to help combat this lifestyle. The most notable example of this is the game “Pokemon Go” that took the world by storm, with a reported player count of 147 million people as of May 2018 [2], to a game based around encouraging running by building a “zombie apocalypse” around the player, a game which lays claim to 5 million downloads [3]. These games share one key mechanism, using the GPS tracking to make the game functionally impossible to play within the boundaries of one’s house, forcing them to go out and get active, even if it is mostly just walking or running around. The game being created for this project will function within the same broad genre of games, another attempt at making America more active. The largest method of exercise is walking/hiking, both things that people do on a daily basis, giving the game a solid root to ground itself in.

This project takes inspiration from Geocaching, which is a game where players put GPS co-ordinates into any GPS device in order to get a rough location of where the cache is and then

having to hunt for where the cache actually is [4], but improve on where some of Geocaching's faults lie. The problems with Geocaching come in a couple ways, which mostly boil down to safety and outside investment. On the topic of safety, GPS co-ordinates put you in a very specific location, so if the geocache is close to a road or another dangerous place it can put a potential player in danger, and since the co-ordinates of geocaches are stored on a website, a player can not just get going because they have to plan around where they will be going, instead of finding things on the way. Geocaches can also be in or around private property, which brings up issues Pokemon Go had in its infancy. Geocaching takes place across two distinct modes of play, which are rigidly split. Users are either trying to find discrete locations to hide away caches, then uploading the coordinates to the central Geocaching website, or they are looking at coordinates on the central website to try to find where the caches have been hidden.

REQUIREMENTS AND SPECIFICATIONS

The basic rules of the game is that there will be digital caches in the world created by users, with a circle drawn around it, which is the area the user has to be in in order to actually claim it for points. The older a cache is, the larger the circle is on the map, and the further out it can be claimed, but it is also worth less points to the person who found and captured it. The quicker a cache is captured after being created, more points would end up going to the person who captured it, as it is harder to find them.

Players of the game have two different modes of play, which players will frequently change between depending on their surroundings. The two modes of play are hunting down caches placed by other players, and the other is finding places to "hide" their own caches. While creating a cache, the player will want to find a spot as far out of the way as they can and then

press the button on their device labeled “Create Cache”, which will send the location of the cache to the central server, as well as put a visible ring underneath the cache. This ring represents how close someone trying to capture their cache would have to be to actually capture it. This ring grows over time, as the cache becomes less valuable to whoever comes along to catch it, as it becomes easier to catch. There are other components that ensures caches will always be caught eventually, as when nearby a cache, the circle will display on the phone screen, giving them a hint that it is nearby, as well as a menu in the profile screen that displays how far out the nearest caches are and a rough heading. Each cache has a point value of 50, but the longer it takes to be caught the more points the person who created it gets from the total point value.

All of this put together creates a game where caches should be created and captured rather frequently, and while someone could sit in one place and wait for a cache to grow to the point they can catch it from anywhere, they will get very few points out of it, ensuring that they do not get far, and must go out and hunt down out of the way locations to hide caches, as well as trying to catch them when they are still freshly created to get the most points.

In order to meet all of these specifications, the game has to be able to do several things. The game will be implemented in Java so that it will work on Android, with the knowledge of the device’s location and the ability to talk to an online database. Knowing the device’s location will let the game know when the user is inside of a cache’s area, and be able to place digital caches, as well as being able to list caches that are close by to them so they are not flying blind.

DESIGN AND IMPLEMENTATION

The front end of the device contains a few basic screens for the end user. The first screen is where the majority of play takes place. This screen displays the local area, any caches within

visible range with a colored overlay, as well as the buttons that capture and create caches. As users walk around, the map will follow them, letting them see the edges of caches and heading towards them for the capture, as long as the phone is on. All of the game logic takes place on this screen, including figuring out if a player is close enough to a cache to capture it.

The second screen is a profile screen, which displays the user's username (from Google Play Store, which is collected when they first boot the game up), their lifetime score, as well as a list of 5 caches that are nearby, but only those five until one is caught or they move sufficiently far away from those 5.

The server has two tables, one to list all of the caches, and one that lists users. The Cache table carries the latitude and longitude, as well as the owner of the cache to reward them points, and the time the cache was created to calculate how points are divided. The user table just carries the owner's ID (taken from Google Play), their display name, as well as the score the user currently has. All of these need to be able to be referenced and editable, so a program called W.A.M.P. (Windows, Apache, MySQL, PHPmyadmin) is used that sets up an Apache web server, which posts PHP web pages. When the app makes a call to one of these web pages, it calls a MySQL command to change the database, and returns a JSON string that tells of its success or failure, as well as any additional information needed by the specific call, such as the user's name or a list of caches.

The project uses a database infrastructure and a map API to display the local area to the game device. The map API used is Google Maps API, which provides free use of the API as long as all it does is display the area, and will work with the high status of the Google the servers are more likely to be reliable. There are other options, such as the open source TomTom, but for the scope of this project they are not needed. The major database structure option is MySQL, which

is a database software based on SQL. The circle in which the caches can be captured is just drawn using simple math with Java.

RESULTS

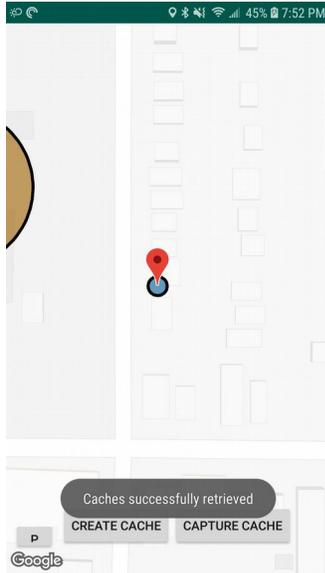


Figure 1: Main Play Screen

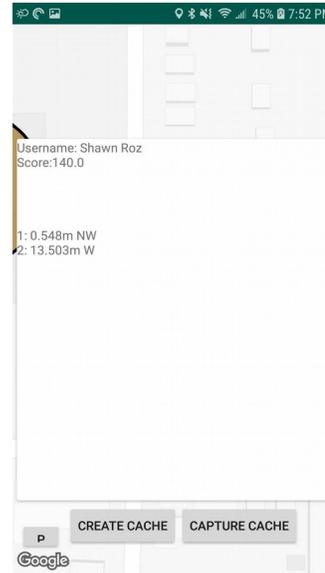


Figure 2: Profile Menu

The game is able to create, display, and score caches, as well as being able to get caches from a central server. The game would have some size problems when scaling up larger, as there are a few times that a sorting function has to be called, and as it is now every cache is downloaded to the user's device when it is being ran. This will be something that could easily be fixed, with a restructuring of the database calls and the use of a spatial index, the problem of speed and size would be alleviated, but not completely solved, which can be solved by using a sort and limit call in SQL instead of a pure call. The only thing that is missing in the game right now is a global leader board and this is something that should be added with relatively high priority. Showing how users compare to each other on a large scale solidifies the game as an

actual game, and there are also numerous ways that numbers could be tweaked in order to promote game balance.

CONCLUSION

The structure and base rules of Capture The Cache creates a game that improves upon its inspiration by solving the security and safety issues of Geocaching, and also avoids the potential issue of Geocaches going lost or disappearing, a monetary hit to whoever it was that made the physical cache in the first place. The app also gamifies common exercises of walking and hiking, encouraging players to go even further with what they would do on their own, filling the required daily amount of exercise, or at least getting closer to that number. The list of caches also helps with this, as if a cache is close by or just out of reach, they might go further just for the sake of catching it.

REFERENCES

[1] “Facts & Statistics”, *HHS.gov*, 09/04/2018.

<https://www.hhs.gov/fitness/resource-center/facts-and-statistics/index.html>

[2] Philips, Tom. “Pokémon Go active player count highest since 2016 summer launch” *Eurogamer*, 06/27/2018,

<https://www.eurogamer.net/articles/2018-06-27-pokemon-go-player-count-at-highest-since-2016-summer-launch>

[3] “Zombies, Run! (Free)”, *Google Play Store*, 09/04/2018 [https://play.google.com/store/apps/details?](https://play.google.com/store/apps/details?id=com.sixtostart.zombiesrunclient&hl=en_US)

[id=com.sixtostart.zombiesrunclient&hl=en_US](https://play.google.com/store/apps/details?id=com.sixtostart.zombiesrunclient&hl=en_US)

[4] “Geocaching 101”, *GeoCaching*, 09/04/2018, <https://www.geocaching.com/guide/>

BIOGRAPHY

Shawn Leedy is originally from West Palm Beach, Florida, but has lived in Indiana for almost all of their life. Since passing high school, they have been using their spare time to focus on running tabletop role playing games as well as enjoying videogames. After graduation, they are going to spend spare time to design their own videogames, a long time passion of theirs, as well as working on other work such as learning a musical instrument, and creating a variety of bots to make repetitive tasks easy.