RUN FOR FUN
USER STUDIES

An Interactive Qualifying Project submitted to the faculty of the
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the Degree of Bachelor of Science by

Christopher Bearce

Antony Qin

Brad Scuzzarella

Date: 23 March 2018

Report Submitted to**:**

Professor Emmanuel Agu
Worcester Polytechnic Institute

Abstract

This project examines exergame enjoyment in order to find clusters of exergames that produce similar enjoyment. First, the team developed a classification system to group exergames together. Classifications include Control-Sports, Control-Action, Control-Adventure, Directed-Static, and Directed-Mobile. Following that, data was gathered through experimentation, where participants played exergames and had values of their enjoyment recorded for each exergame played. The strongest relationship between classifications, according to a resulting relationship network, was between Control-Adventure and Control-Action, and so both classifications produced similar measured enjoyment. Finally, the gathered data was put to use in a functioning recommender system that recommended users exergames that they have not played. This developed recommender system returned accurate recommendations 83% of the time.

Table of Contents

CHAPTER 1: Introduction

1.1 - Introduction

Exergames are designed to make exercise more appealing and easier to enjoy. An exergame is defined as a “video game that requires gross motor activity” [25]. Therefore, for a game to be an exergame, it must meet two requirements. First, it must contain some form of exercise, such as walking, and second, it must have a gamified experience, such as a score system. A well-known game that drew widespread appeal is *Pokemon Go* [19]. The game exploded onto the scene in 2016, becoming the first mobile game to gross 100 million downloads within 20 days [28]. Although still popular, it experienced a sharp drop-off in user population [2]. These lost players are suddenly not playing *Pokemon Go* [19] anymore. This means that a percentage of these lost players are possibly not getting exercise anymore. It is speculated that players stop playing games when they become bored with their current game. No game can retain every player that plays, but perhaps a recommender system can be developed to recommend a new exergame to a bored player. A recommender system is a set of software tools and techniques providing suggestions for items to be of use to a user [21]. This way, the players that were motivated to exercise because of game elements, can continue to achieve fitness.

This project relies on the work and ideas of a previous Run for Fun IQP, and Cypress. The previous Run for Fun project was dedicated towards creating an Exergame Enjoyment Questionnaire, or EEQ, in order to quantify a player's enjoyment toward an exergame [7]. Enjoyment is defined as flow, “a subjective state that people experience when they are completely absorbed in something to the point of forgetting time, fatigue, and everything but the activity itself" [27]. At the time, the EEQ was the first questionnaire to measure enjoyment from exergames. The EEQ was used in this current project in order to record player enjoyment towards exergames during experimentation. Cypress was a proposed recommender system that does three things. First it uses smartphone sensing technology in order to measure user enjoyment of exergames, it learns the types of exergames the user enjoys, and finally it recommends new exergames to the user when the user becomes bored (enjoyment falls) with the current exergame [4]. Because user preference differs person to person, it is important to take it into account during recommendation. This project attempts to fulfill the vision of Cypress by creating a recommender system that takes user preference into account.

The goal of this IQP was to gain a better understanding of user preferences regarding smartphone exergames. This was accomplished with the following steps:

1. Create a classification system for smartphone exergames so that user preferences are directed toward groups of exergames, not to the individual games themselves.
2. Gather data from multiple experiments, where users play selected smartphone exergames and have their enjoyment measured with the EEQ.
3. Develop a recommendation system that can take input data of past user enjoyment, construct user preferences, and recommend different smartphone exergames tailored to each specific user.
4. Validate the recommendation system.

By completing the prior steps, it can be proven that user preferences regarding smartphone exergames can be extrapolated from prior enjoyment of particular exergames and then used to provide accurate recommendations.

Chapter 1, the Introduction, covers broad topics of the project and defines key terms such as the EEQ and recommender system. Chapter 2, Related Work / Background, delves deeper into the topics of Chapter 1, the Introduction. It gives some history on exergaming in general. It explains the EEQ, and how quantified enjoyment was derived from the questionnaire. It also walks through the process of exergame selection and how Hope Wallace's Evaluation of Exergames factored into the process. Chapter 3, Methodology, explains the thought process during the creation of exergame classifications. It walks through high-level experiment procedure and gets into the specific procedures by exergame. It goes over projected variables before experimentation occurred and ends with experiment experiences after experimentation concluded. Chapter 4, Results & Recommendation Analysis, displays the data gathered from experimentation in table form, as well as its resulting visualizations. It then analyzes both the raw results and the resulting figures. It also includes exact demographic data. Chapter 5, Recommender System, explores the recommender system created in response to the gathered data. Finally, Chapter 6, Conclusion, covers closing thoughts on the project and what the future entails.

CHAPTER 2: Related Work / Background

2.1 - Exergame History

Exergaming as a form of virtual entertainment began in the late 1980s when Nintendo and Atari premiered their Power Pad for the NES and Foot Craz for the Atari 2600, and had its first major success in the 1998 when Konami premiered their *Dance Dance Revolution* series [26]. The game started out as an arcade game with specialized hardware in Japan. In *Dance Dance Revolution*, players stand on a dance pad with four arrows while a song plays. As the song plays, the player is presented with patterns of arrows on screen, which the player would need to rhythmically stamp out on the pad, in order to achieve a higher score. This opened up the video game industry to the new concept of exergames as it spread across arcades worldwide and throughout homes on the Sony Playstation. Following *Dance Dance Revolution*, the doors opened up throughout the 2000s for exergames. Playstation tried their own hardware extension titled the EyeToy in 2004, which was a camera that allowed a player to interact with the system using body movements [12]. Nintendo also joined the scene with the revolutionary motion controlled console, the Nintendo Wii. The console featured the hugely successful *Wii Sports*, which sold over 21 million copies [5]. This led to 2 billion in sales of health games in 2009 [18]. Although, most of the sales could be attributed to console games such as *Wii Fit* and *EA Sports Active* [18]. But even that great sum pales in comparison to the mobile gaming market of today. In the past few years the amount of players of mobile games has skyrocketed, along with the profit they bring in. In the 2017 fiscal year, mobile games generated 46.1 billion dollars, or 42% of the income of all video games [15]. This was the largest share of profits from the main three groups of video games today; mobile, console, and pc gaming. This boom in the market caused developers to put out mobile exergames using the GPS capabilities of modern smartphones, which led to the production and rise of the mobile goliath that was *Pokemon Go* [19]. *Pokemon Go* [19] took the world by storm when it was initially released and continued to gain incredible popularity for the months that followed. Over time, however, something happened. The popularity finally hit a peak, and gamers began to grow tired of what the game had to offer [2]. This problem leads into the purpose of this study. The aim was to find different correlations between mobile exergames so that players can receive tailored recommendations of exergames before losing interest. Looking forward, phones and tablets are not going anywhere, and neither is the video game industry. Mobile game popularity is growing as fast as the mobile phone industry is. In the past 5 years, the percentage of adults in the United States that owned a smartphone spiked from 35% to 77%, and the percentage of tablet owners rose from 3% to 51% in the same timeframe [24]. A majority of these devices are equipped with GPS capabilities, which is perfect for exergaming. And now the video game industry has gained virtual reality technology with devices such as *Oculus Rift*. The future of exergaming could very well lie there.

2.2 - Exergame Enjoyment Questionnaire (EEQ)

In order to measure the user's enjoyment of a particular exergame, a questionnaire was distributed after play. The Exergame Enjoyment Questionnaire, also known as the EEQ, produces an e-score, or enjoyment score, that measures a player's particular enjoyment toward a game [7]. The EEQ was developed from prior questionnaires such as the GEQ, or Game Engagement Questionnaire [3]. It includes questions that test the categories of immersion, intrinsically rewarding activity, control, and exercise. Each of these categories "assess player enjoyment of game elements" [7].

2.2.1 - Example EEQ

This was the following questionnaire that was given to a participant of an experiment after playing an exergame:

Run for Fun: Exergame Enjoyment Questionnaire

For each of the statements below, please circle how much you agree or disagree with the statement.

1. I felt excited about the physical activities in the game.

   Strongly Disagree   Disagree   Neutral   Agree   Strongly Agree

2. The exercise in this game made me feel good.

   Strongly Disagree   Disagree   Neutral   Agree   Strongly Agree

3. I felt like I lost track of time while playing

Strongly Disagree Disagree Neutral Agree Strongly Agree

4. I felt that it was difficult to understand how the game works.

Strongly Disagree Disagree Neutral Agree Strongly Agree

5. I was focused on the game.

Strongly Disagree Disagree Neutral Agree Strongly Agree

6. I felt that the game would have been more enjoyable without physical activity.

Strongly Disagree Disagree Neutral Agree Strongly Agree

7. I felt that it was easy to familiarize myself with the game controls.

Strongly Disagree Disagree Neutral Agree Strongly Agree

8. I felt emotionally attached to the game.

Strongly Disagree Disagree Neutral Agree Strongly Agree

9. I consider playing the game “exercise”.

Strongly Disagree Disagree Neutral Agree Strongly Agree

10. I felt that the physical activity was too intense for me.

Strongly Disagree Disagree Neutral Agree Strongly Agree

11. I did not feel a desire to make progress in the game.

Strongly Disagree Disagree Neutral Agree Strongly Agree

12. I felt a strong sense of being in the world of the game to the point that I was unaware of my surroundings.

Strongly Disagree Disagree Neutral Agree Strongly Agree

13. I would rather not be exercising, even though the exercise was accompanied by game elements.

Strongly Disagree *Disagree* *Neutral* *Agree* *Strongly Agree*

14. I felt that playing the game was beneficial for my physical well-being.

Strongly Disagree *Disagree* *Neutral* *Agree* *Strongly Agree*

15. I felt that this game provided an enjoyable challenge.

Strongly Disagree *Disagree* *Neutral* *Agree* *Strongly Agree*

16. I felt a sense of accomplishment from playing the game.

Strongly Disagree *Disagree* *Neutral* *Agree* *Strongly Agree*

17. I felt that the game reacted quickly to my actions.

Strongly Disagree *Disagree* *Neutral* *Agree* *Strongly Agree*

18. I did not feel like I wanted to keep playing.

Strongly Disagree *Disagree* *Neutral* *Agree* *Strongly Agree*

19. I would prefer that this physical activity was not accompanied by game elements.

Strongly Disagree *Disagree* *Neutral* *Agree* *Strongly Agree*

20. I felt in control of the game.

Strongly Disagree *Disagree* *Neutral* *Agree* *Strongly Agree*

2.2.2 - Scoring the EEQ

After taking the EEQ, the questionnaire produced an e-score that quantified the player's enjoyment towards the game. If the question was phrased positively, the e-score increased by 1 to 5 corresponding to strongly disagree to strongly agree. If the question was phrased negatively, the e-score increased from 5 to 1 corresponding to strongly disagree to strongly agree. For example, the question numbered 20, "I felt in control of this game" was phrased positively, and an answer of strongly agree increased the e-score by 5. In contrast, the question numbered 18, "I did not feel like I wanted to keep playing" was phrased negatively, and an answer of strongly agree resulted in the e-score being increased by 1. Therefore, the theoretical minimum e-score was 20, and the theoretical maximum e-score was 100. Positively phrased questions are as follows: 1, 2, 3, 5, 7, 8, 9, 12, 14, 15, 16, 17, and 20. Negatively phrased questions are as follows: 4, 6, 10, 11, 13, 18, and 19. In the creation of a questionnaire such as the EEQ, it was essential that there are a range of questions varying in agreement difficulty. Because negatively phrased questions are deemed the most difficult to agree with, they are balanced out with positively phrased questions that are the easiest to agree with [3].

2.3 - Hope Wallace's Evaluation of Exergames

One resource useful to the study was a previous student's work searching for available exergames on Android and Apple devices. The student, Hope Wallace, put together a spreadsheet of roughly 80 exercise-based mobile games that she made available to the public. An example "snapshot" of her spreadsheet is shown in Table

2.3.1. The ellipses represent columns/rows that were in the original spreadsheet, but took up too much space or did not show pertinent information. Information included a brief description of the game, reviews, and category. Although she did not formally define the categories where she placed games, the idea itself helped the development of the study's classifications. A lot of the applications on the spreadsheet, such as *S Health*, *Nike+ Running*, and *Fitbit*, were just simple programs that took note of the user's progress while exercising. By definition, these are not considered exergames because there was no gamification to them. However, a few games from Hope's spreadsheet were useful for this study. Some of the games she listed that are being adopted for the experiment are *Ingress* [10] and *Just Dance Now* [11]. A few of her search protocols were followed to great success as well. For example, some of the keywords she noted in the spreadsheet were used when searching for newer exergames on the Google Play Store.

Table 2.3.1: Snapshot of Hope Wallace's Exergame Spreadsheet

Name	**...**	**Brief Description**	**Category**	**...**
Fitbit	...	Tracks runs, steps, miles, calories burned, etc.	Pedometer Gamification	...
Google Fit - Fitness Tracking	...	Tracks runs, steps, miles, calories burned, etc.	Pedometer Gamification	...
...	...	...	...	...
7 Minute Workout	...	Virtual fitness coach, workouts in 7 minute increments	Fitness	...
Strava Running and Cycling GPS	...	Tracks distance, speed, pace, elevation, etc. Social network	Pedometer Gamification	...
Under Armour Record	...	Tracks and analyzes fitness, sleep, steps, nutrition, weight	Pedometer Gamification	...
Ingress	...	Augmented reality, move in the real world to help your team win	Adventure	...
Turf Wars - GPS-Based Mafia	...	Augmented reality, move in the real world to claim territory	Adventure	...
...	...	...	...	...

2.4 - Exergame Selection

Exergames were chosen from the Google Play Store. Alongside searching within the Google Play Store, the internet was employed. The team took to various exergaming blogs with rankings of some of the best exergames on the market. Keywords used for searching included, “exergame”, “gps game”, “exercise game”, and “movement games”. The internet was home to many different forums ranking exergames for their playability and enjoyment, as well as lists of categorized mobile exergames. Initially, games were picked and tested without rhyme or reason. There was no criteria for narrowing down games prior to testing, so the process was inefficient at first. In order to rectify this, a set of requirements had to be met for each potential exergame. These requirements include a freemium or free business model, a minimum number of downloads, a minimum number of ratings, and a minimum overall rating. The requirements are noted in Table 2.4.1 below.

Table 2.4.1: Exergame Requirements

Criteria	Reasoning
Free to Play	● Easy to acquire and test
>2.5 star rating	● Less chance of a buggy game
Primarily be in English	● The game must be understandable
>1000 downloads and >20 reviews	● The more popular the game, the better tested and safe it is

By primarily picking free games, obstacles arose such as paywalls locking away a majority of the game, a constant slew of ads, and overall poor craftsmanship. After trawling through the available free games, it was determined that in order to acquire the desired amount of games, pay-to-play games needed consideration as well. This refined search, including paid games, returned additional, genuine exergames. It also allowed for further exploration into the full games of the free versions that were originally downloaded.

CHAPTER 3: Methodology

3.1 - Exergame Classifications

In order to compare and contrast exergame enjoyment between different games, categories of exergames need to be established. In doing so, clusters of categories may be possible, where a particular category of exergames lends itself to the same enjoyment as another. This study created exergames classifications with specific rules. Games are assigned a type and a genre under its type. This combination makes up a classification. Opposing categories are always mutually exclusive. This ensures that every exergame was properly categorized into one classification. A diagram that shows the following branches of classifications is shown in Figure 3.1.1.

Control - Exergames that have the player controlling an in game character with their body. The focus is not on the player, but in the game. In other words, the game follows the player.

- **Sports -** Exergames that simulate popular sports in a gamified manner.
- **Non-Sports -** Exergames that do not exhibit popular sports.
  - **Action -** Exergames that are focused on the micro. This means that capturing the player’s immediate bodily movements is the focus.
  - **Adventure -** Exergames that are focused on the macro. This means that tracking the player’s movement across the real world is the focus.

Directed - Exergames that have the game directing the player to move in the real world. The focus is not in the game, but on the player. In other words, the player follows the game.

- **Mobile -** Exergames that have the player performing actions across multiple locations.
- **Static -** Exergames that have the player performing actions in one set location.

Some exergames may exhibit features of multiple categories. For example, *Spectrek Light* [23] exhibits Control-Adventure as the game tracks the player moving about in the real world using GPS, but it also has moments of Control-Action when the player must stop and aim their phone about to capture a ghost with the camera. In events like this, the dominant category the game exhibits in play becomes the category the game was assigned. *Spectrek Light* [23] is clearly a Control-Adventure game, as it exhibits features of that category with greater frequency. Table 3.1.2 shows the classifications assigned to each game used in experimentation, along with reasoning as to why.

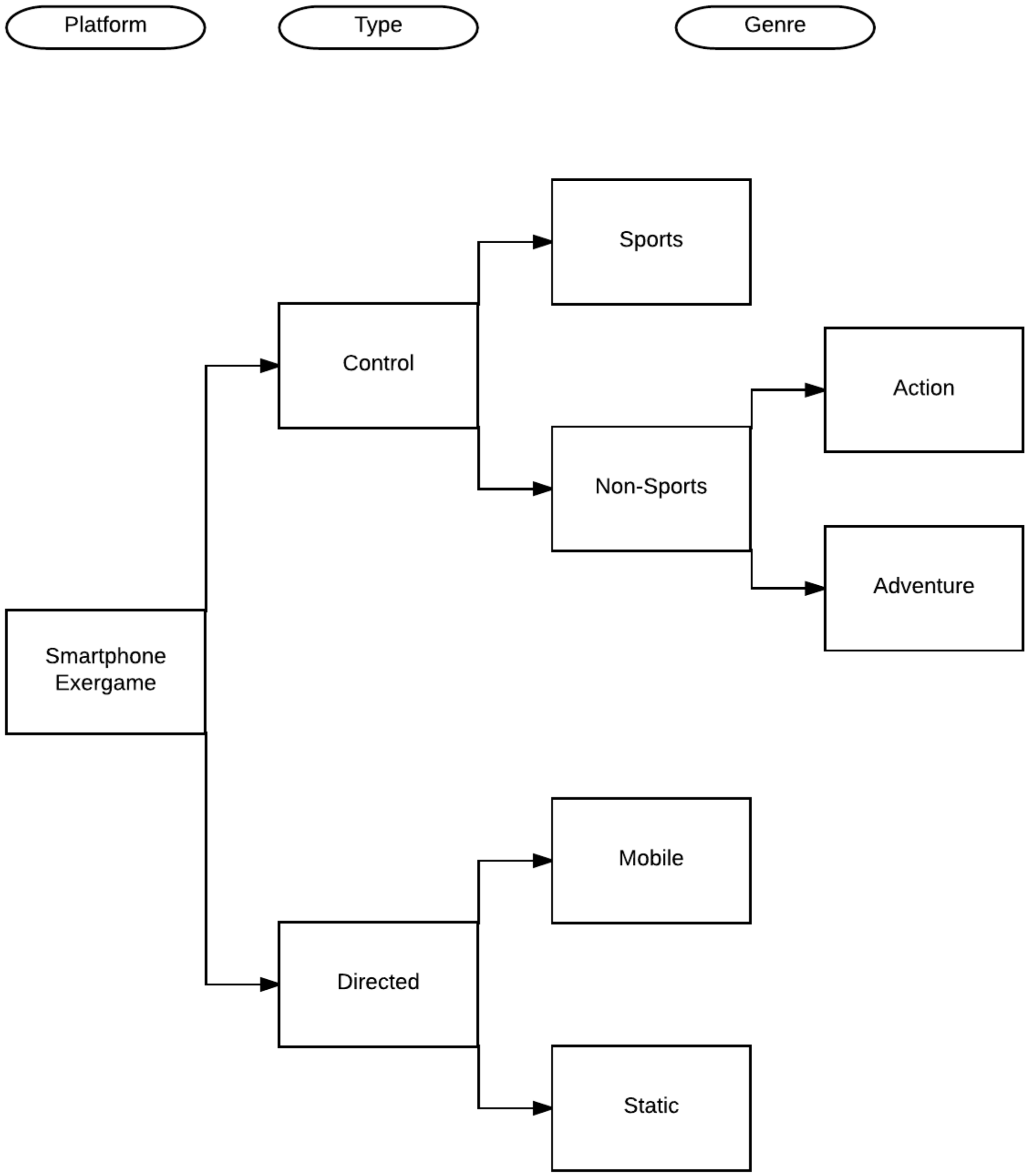


Figure 3.1.1: Exergame Classification Tree

Table 3.1.2: Table of Classified Exergames

Game	Price	Description	Classification	Reasoning
Motion Tennis [17]	Free	Player plays tennis using their phone as if it were a racket, against a CPU that is cast onto a screen.	Control-Sports	• The player is controlling an in game character with their phone. The phone tracks using accelerometer and gyroscope. • Exhibits a popular sport - tennis.
Motion Sports [16]	Free (limited version) $2.99 (full game)	The player may choose between swimming, running, skiing and soccer.	Control-Sports	• The player is controlling an in game character using their body, which the phone's camera tracks. • Exhibits multiple popular sports - swimming, running, skiing, and soccer.
FitFlap Motion [6]	Free	Player flaps arms to control an animal to move up or down, avoiding incoming obstacles.	Control-Action	• The player is controlling an in game character using their body, which the phone's camera tracks. • Does not exhibit a popular sport. • The control is centered around player's bodily movements.

Freeline [8]	Free (limited version) $2.99 (full version)	Player leans their body or flaps arms to control a spaceship in a race along a track.	Control-Action	- The player is controlling an in game character using their body, which the phone's camera tracks. - Does not exhibit a popular sport. - The control is centered around player's bodily movements.
Pokemon Go [19]	Free	Player walks about the real world, with options to capture pokemon or visit special locations.	Control-Adventure	- The player moves an in game character by walking about in the real world, which the phone's GPS tracks. - Does not exhibit a popular sport. - The control is centered around player's movement through the world.
Spectrek Light [11]	Free	Player walks about the real world in order to capture ghosts.	Control-Adventure	- The player moves an in game character by walking about in the real world, which the phone's GPS tracks. - Does not exhibit a popular sport. - The control is centered around player's movement through the world.

Ingress [10]	Free	Player walks about the real world with options to capture “portals” that represent real world landmarks.	Control-Adventure	● The player moves an in game character by walking about in the real world, which the phone’s GPS tracks. ● Does not exhibit a popular sport. ● The control is centered around player’s movement through the world.
Just Dance Now [11]	Free (limited version) $2.99/ month to unlock all songs	Player moves in conjunction with the model on the screen in dance motions to the music selected.	Directed-Static	● The game directs the player how to dance to a particular song. ● The game takes place in one location.
7 Minute Superhero Workout [1]	$4.99	The player is told to do various exercises to progress a superhero themed story	Directed-Static	● The game directs the player how to exercise. ● The game takes place in one location.
Shape Up Battle Run [22]	Free	Player is guided through a running workout for selected times and intensities.	Directed-Mobile	● The game directs the player how to run. ● The game takes place as the player moves about locations.
Zombies, Run! [29]	Free	The player’s run is accompanied by a story of a zombie apocalypse.	Directed-Mobile	● The game directs the player how to run. ● The game takes place as the player moves about locations.

3.2 - Experiment Procedure

The following three sections (Section 1.1-1.3) refer to proper protocols when conducting an experiment. Procedure starts from Section 1.1 and then may flow into Section 1.2 or Section 1.3 depending on location of play.

SECTION 1.1

1. Investigators explain the nature of the study and review the informed consent form with participants. (This step was only performed for the SONA studies)
2. Participant signs the informed consent form. (This step was only performed for the SONA studies)
3. Investigators administer the pre-participation questionnaire.
4. Investigators inform participants what games they play.
5. Participant then plays one game for 15 minutes
    - ➢ If outside refer to Section 1.2
    - ➢ If inside refer to Section 1.3
6. Participant fills out a post-game questionnaire
7. Participant then plays another game for 15 minutes
    - ➢ If outside refer to Section 1.2
    - ➢ If inside refer to Section 1.3
8. Participant fills out a final post-game questionnaire
9. Investigators debrief the participants about the study.

SECTION 1.2

1. Participant is given phone with games preloaded
2. Investigators assign the participant a game to play
3. Investigators define the boundaries of the playing field
4. Investigators give instructions depending on the game assigned
5. Participant plays the assigned game for 15 minutes while remaining inside the assigned boundary
6. Participant returns to the investigator

SECTION 1.3

1. Participant is given phone with games preloaded
2. Investigators assign the participant a game to play
3. Investigators demonstrate proper techniques to safely maneuver in the indoor space
4. Investigators give instructions depending on the game assigned
5. Participant plays the assigned game for 15 minutes
6. Participant returns to the investigator

3.2.1 - Game Specific Procedures

Here follows the procedure used to test specific games. The participants are free to download and play them on their own time. Every game was available from the Google Play Store. Participants may experience some exhaustion or dehydration due to exercising. At any time, the participants may have taken a break or asked for water if they felt as if they are over exerting themselves. Any game that was selected for the participants to play is explained as follows:

FitFlap Motion [6]:

For this game, the phone was placed on a desk and connected a laptop screen by USB. The participant stood in front of the laptop screen, which acts as a large phone screen through the program SideSync, and played the game. In any event of confusion, the proctor helped the participant by answering any questions or troubleshooting any technical difficulty. During play, the participant moved his/her body as the controller of the animal within the game. Movement involves participants flapping their arms to mimic flapping wings. This game is not physically demanding and was played indoors. For a screenshot of the game, refer to Figure 3.2.1.1.

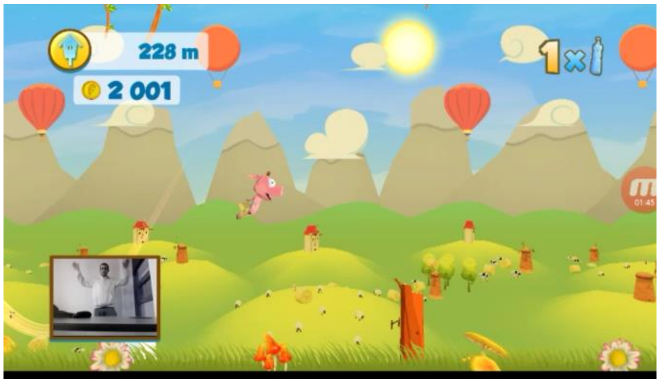


Figure 3.2.1.1: Screenshot of FitFlap Motion [6]

Freeline [8]:

For this game, the phone was placed on a desk and connected a laptop screen by USB. The participant stood in front of the laptop screen, which acts as a large phone screen through the program SideSync. The participant was asked to play the tutorial, and upon completion, had free rein in choosing a racetrack to play. In any event of confusion, the proctor helped the participant by answering any questions or troubleshooting any technical difficulty. During play, the participant moved his/her body as the controller of the spaceship within the game. Movement involves participants leaning their body side to side, or raising arms to fly. This game is not physically demanding and was played indoors. For screenshots of the game, refer to Figure 3.2.1.2.

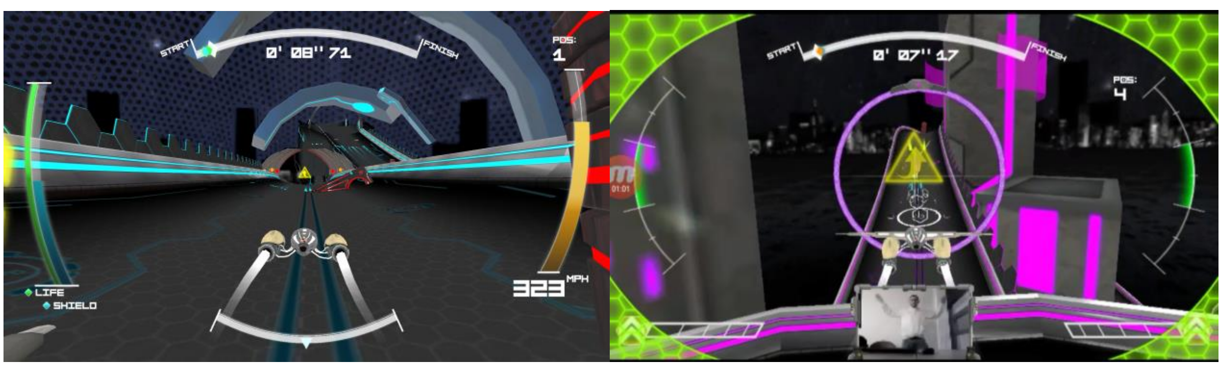


Figure 3.2.1.2: Screenshots of Freeline [8]

Ingress [10]:

For this game, the participant was given a mobile phone and instructed to stay within WPI campus bounds. After the game was explained, a timer was enabled on the phone, and the participant was instructed to return when it goes off. The participant chooses a faction, either resistance or the enlightened, to join. The participant was then released outside, and can walk from location to location on campus. The locations that the participant travels to are portals that the participant can either try to capture or perform various other actions if the participant's faction owns the portal. Exercise is non-strenuous, unless the participant wants to hurry from location to location. For screenshots of the game, refer to Figure 3.2.1.3.

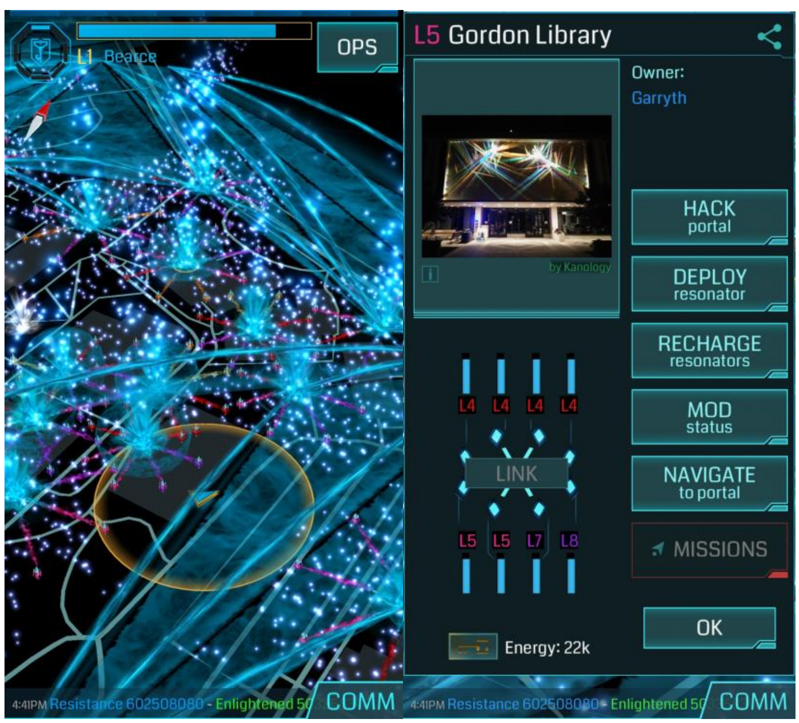


Figure 3.2.1.3: Screenshots of Ingress [10]

Pokemon Go [19]:

For this game, the participant was given a mobile phone and instructed to stay within campus bounds. After the game was explained, a timer was enabled on the phone, and the participant was instructed to return when it goes off. The participant was then released outside, and can walk from location to location on WPI campus. During play, these locations range from Pokestops (to gather items) to Pokemon (to catch). Exercise is non-strenuous, unless the participant wants to hurry from location to location. For screenshots of the game, refer to Figure 3.2.1.4.

Figure 3.2.1.4: Screenshots of Pokemon Go [19]

Spectrek Light [23]:

For this game, the participant was given a mobile phone and instructed to stay within campus bounds. After the game was explained, the participant was instructed to return when time runs out. The participant was then released outside. During play, the phone displays a map of the playing area with question marks noting areas of interest. The player can walk from area to area on WPI campus. These areas of interest are ghosts to catch. When a participant reaches a ghost on the map, s/he can raise the phone and capture the ghost with the phone's camera. Exercise is non-strenuous, unless the participant itself wants to hurry from location to location. For screenshots of the game, refer to Figure 3.2.1.5.

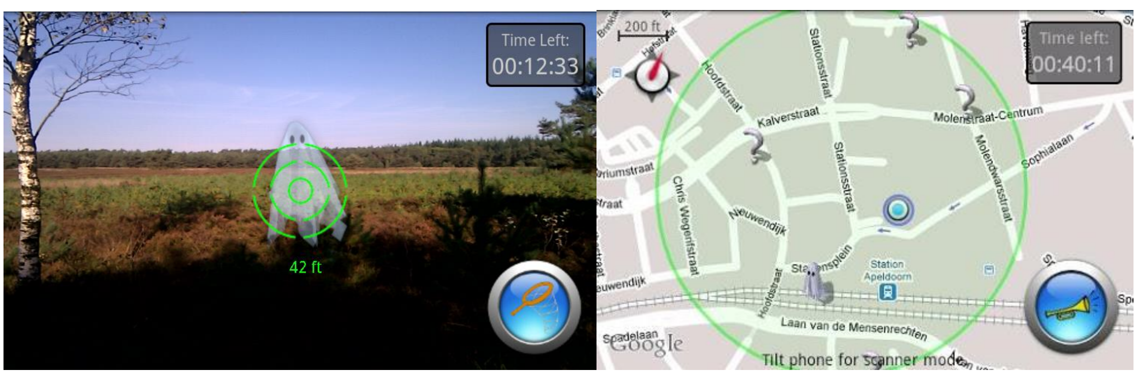


Figure 3.2.1.5: Screenshots of Spectrek Light [23]

Just Dance Now [11]:

For this game, the participant was given a phone. The phone was connected to a projector screen through the game's unique browser based Wi-Fi connection. In order to do this the game's site (justdancenow.com) was opened on the browser, which displays a QR code and number called the "dancing room". The game was then opened on the phone, and connected to the browser either by scanning the QR code, or inputting the "dancing room" number. The participant stood in front of the screen. After explaining the game, the participant played the game. During play, the participant danced to a chosen song, mimicking the actions of the onscreen actors, where a better mimic leads to a higher score. Exercise may be moderately strenuous, depending on the intensity of the song, but doable. For screenshots of the game, refer to Figure 3.2.1.6.

Figure 3.2.1.6: Screenshots of Just Dance Now [11]

Motion Tennis [17]:

For this game, a mobile phone was given to the player and connected to a projector screen using Chromecast. To do this, a separate device started a mobile hotspot, which the given mobile phone and Chromecast connected to. The Chromecast was connected to the projector by HDMI. The participant stood in front projector screen and could choose any difficulty to play. In any event of confusion, the proctor helped the participant by answering any questions or troubleshooting any technical difficulty. During play, the participant uses the phone as if it were a racket. The tennis game was shown on the projector screen, and the in game character's racket moves according to the participants movements. This game is not physically demanding and was played indoors. For screenshots of the game, refer to Figure 3.2.1.7.

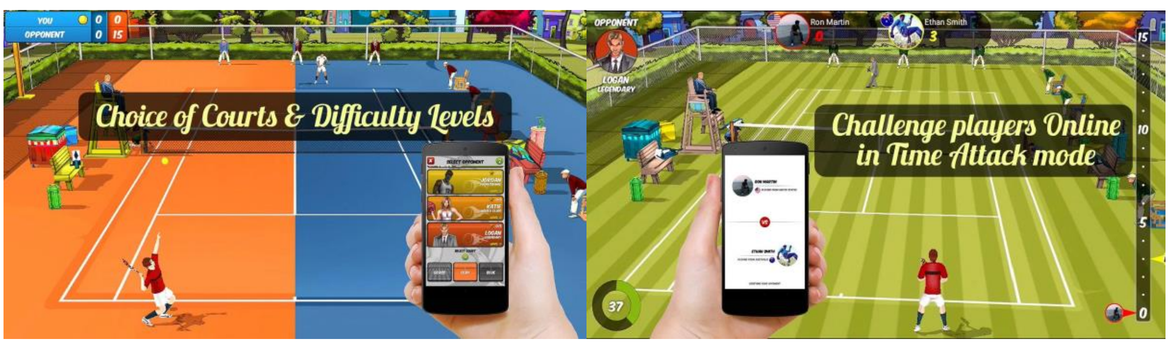


Figure 3.2.1.7: Screenshots of Motion Tennis [17]

Motion Sports [16]:

For this game, the phone was placed on a desk and connected to a laptop screen by USB. The participant stood in front of the laptop screen, which acts as a large phone screen through the program SideSync. In any event of confusion, the proctor helped the participant by answering any questions or troubleshooting any technical difficulty. During play, the participant moved his/her body as the controller of the athlete within the game depending on the sport selected. During swimming, movement involves the participant wind milling their arms as if they were swimming freestyle. During skiing, movement involves the participant moving side to side to dodge incoming obstacles. During running, movement involves the participant running in place. During soccer, the participant moves side to side to block the ball. This game is not physically demanding and was played indoors. For screenshots of the game, refer to Figure 3.2.1.8.

Figure 3.2.1.8: Screenshots of Motion Sports [16]

Shape Up Battle Run [22]:

For this game, the participant was given a mobile phone and headphones if they do not have a pair on hand. There were multiple workout options, but the participant was instructed to select the 15 minute workout. The 15 minute option is a “run” option in which the participant runs for 15 minutes at their own pace while periodically being challenged to speed up for a short amount of time. During these more intense sections, the player was encouraged to pick up the pace to obtain a higher score, however, the sped up portion is dependent on the initial running speed of the participant to avoid over exerting themselves. This game is moderately physically demanding and was played outdoors or indoors on a track. For screenshots of the game, refer to Figure 3.2.1.9.

Figure 3.2.1.9: Screenshots of Shape Up Battle Run [22]

7 Minute Superhero Workout [1]:

For this game, the mobile phone was placed on a desk, as the camera was used to track the participants exercise. Following the directions from the phone, the player engages in a workout, such as pushups, crunches, or punches. The workouts are wrapped together with a story, or a mission, to give the player a reason for their actions. This game is moderately physically demanding and was played indoors. For screenshots of the game, refer to Figure 3.2.1.10.

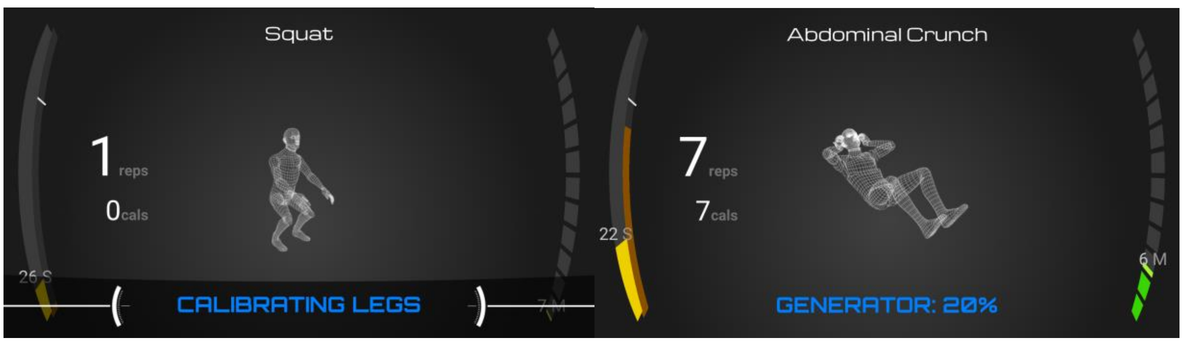


Figure 3.2.1.10: Screenshots of Seven Minute Superhero Workout [1]

Zombies, Run! [29]:

For this game, the participant was given a mobile phone and headphones if they do not have a pair on hand. After the game was explained, a timer was enabled on the phone, and the participant was instructed to return when it goes off. During play, while instructed to stay within campus bounds, the participant can choose to either jog or run. While in motion, a zombie apocalypse driven story was narrated for the player, encouraging them to move away from virtual dangers. This game is moderately physically demanding and was played outdoors or indoors on a track. For screenshots of the game, refer to Figure 3.2.1.11.

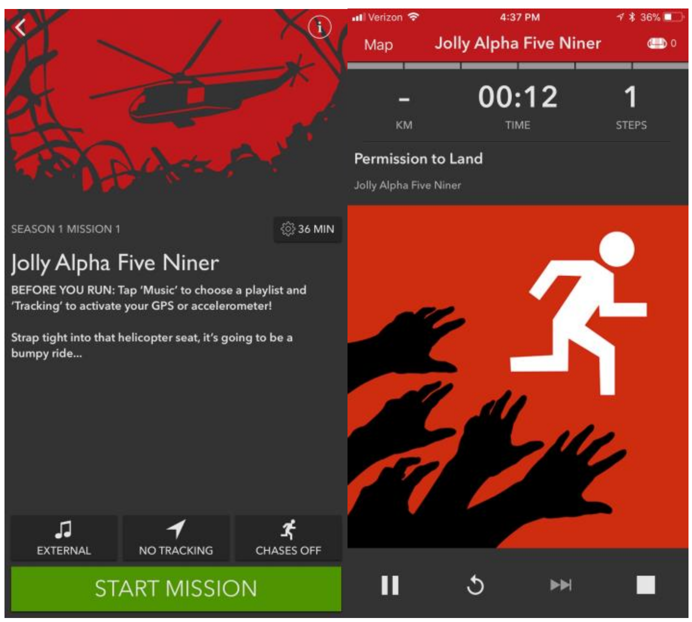


Figure 3.2.1.11: Screenshots of Zombies, Run! [29]

3.3 - Experimentation - Prior

3.3.1 - Participant Target

The main share of participants should come from SONA, also known as the WPI Social Science Research Participation System. The study was posted to the SONA system. WPI students could then sign up for the study. In return for participation, students receive credit towards their social science classes. In all psychology classes at WPI, a student's grade is dependent on the credits gained from SONA. Another plan to recruit students around WPI campus was to post flyers (shown as Figure 3.3.2.1) in various locations throughout campus. The final method for recruiting participants was gathering friends and peers from organizations around campus. The goal was to gather 125-150 people in order to supply enough data to find correlations and clusters between player enjoyment of various exergames. WPI's gender ratio was skewed to the male side, so the SONA participants for this study were selected to make the split as even as possible.

Figure 3.3.1.1: Example Flyer

3.3.2 - Data Collection

As stated previously, the goal was to recruit around 125-150 people to participate in the study. Each student plays at least two games during the study, resulting in about 250-300 data points. A balance of quantity and quality was needed in the selection of testable games. Quantity was mostly maintained, with 2 exergames for each classification except for Control-Adventure, which has 3. There were originally 4 games under Control-Adventure, until *Treazr Hunt* was cut in order to maintain quality. There are 11 exergames selected for this study. This should generate between 25 and 30 data points for each individual game and be enough information to identify different clusters of exergame enjoyment. Having a large amount of information for each game helped ensure the legitimacy of any conclusions regarding the data.

3.3.3 - Pilot Testing of Selected Exergames

As different exergames to be included in the study were found, each member of the team individually tested the game. After testing the game for playability and gamified exercising, it was then tested under the defined experimental conditions, and the post-game questionnaire was filled out. If it was believed that the game was worthy of inclusion in the study, the game was brought up to the other group members to also test out and take the questionnaire. If the other group members considered it a worthy game to include in the research, the game was then be added to the list of games. This prior testing also provided three points of preliminary data for each game.

3.3.4 - Testing Pattern

In order to assure that the data that was gathered was evenly spread out by game and category, the team came up with a formal testing pattern. The testing pattern was set up as a simple sequential cycle, testing each category against every other. Upon reaching the final step in the sequence, the next combination of categories to be tested against one another would once again start from the first step. In the event of a combination that had already been tested, then it was skipped until the next time it showed up. The sequence is shown below in Figure 3.3.4.1:

1. Sports ➞ Action
2. Sports ➞ Adventure
3. Sports ➞ Static
4. Sports ➞ Mobile
5. Action ➞ Adventure
6. Action ➞ Static
7. Action ➞ Mobile
8. Adventure ➞ Static
9. Adventure ➞ Mobile
10. Static ➞ Mobile

Figure 3.3.4.1: Testing Sequence

The specific games that were tested together were also noted. This was in the effort to limit the amount of times the same two games are tested with each other.

3.4 - Experimentation - After

3.4.1 - SONA Participant Pool

Throughout the length of the study, there were two main groups of participants. The first being participants through SONA through the WPI Social Science Research

Department. The SONA system was a helpful resource for acquiring data. Some of these participants, while they did generate data points, did not seem very interested in the study itself, because they were only there to get a grade in return. Despite this, most participants were attentive during the study and volunteered their time as per the researcher's requests.

3.4.2 - Peer Participant Pool

The other main group of participants came from the peers of the researchers. There were not as many obstacles when holding studies with peers. The researchers could have the participants download the games on their own phones and play them together with other peers to record data in bulk. Peers also seemed to be more interested in the study and making sure the study went well, as they did not have any other incentives to participate. The WPI Wireless networking issues were also reduced if the participants were using their own devices. The research team was also free to hold experiments with peers off the WPI campus during their own time.

3.4.3 - Experiment Difficulties

There were a few challenges in coordinating these studies. One being the fact that the team needed to reserve a room on campus to hold the study. Using 25live off a browser, rooms could be reserved while skirting the official reservations of class times, which showed up on the calendar. As students, unfortunately, even reserved rooms may be overturned by higher authorities. For example, after showing up to a reserved room for an experiment, it turned out that the room was suddenly being used to hold a statistics test. Reservations may be overturned as last minute as a day before the study in the reserved room. Due to this, the team was forced to make changes to the testing patterns on the fly with whatever equipment that was readily available.

Technology was also an issue. The nexus phone that was supplied to conduct the studies with refused to connect to the WPI Wireless Network for a few days. Any game that required the GPS or internet was not playable on this device. After troubleshooting this, however, the team was able to get it up and running. A Chromecast was used to cast the phone screen onto a larger screen. This allowed for the players to see the games better. This Chromecast had no way to connect to the WPI Wireless system and could therefore not be used. However, this problem was circumvented using a mobile hotspot. Both the phone and the Chromecast were connected to a researcher’s mobile hotspot.

Finally, poor weather conditions affected the study. During some studies, because of the extreme cold and icy conditions, Control-Adventure and Directed-Mobile games could not be tested in order to maintain safety.

CHAPTER 4: Results & Recommendation Analysis

4.1 - Raw Results

All participants were assigned a numerical id to remain anonymous. Results from each anonymized participant are tabulated in a master spreadsheet, shown in Table 4.1.1. The spreadsheet denotes an e-score for a particular game of a participant, underneath the game's given classification. The titles of the games have been concatenated to conserve space, but starting from the "Game ->" row, the full titles are *Motion Sports* [16], *Motion Tennis* [17], *FitFlap Motion* [6], *Freeline* [8], *Pokemon Go* [19], *Spectrek Light* [23], *Ingress* [10], *Just Dance Now* [11], *7 Minute Superhero Workout* [1], *Shape Up Battle Run* [22], and *Zombies Run* [29].

Table 4.1.1: Raw Results

	Smartphone Exergame											
	Control							Directed				
	Sports		Non Sports					Static		Mobile		
			Action		Adventure							
Game ->	Mot. Spts.	Mot. Tns.	Fit. Mot.	Frln.	Pkm. Go	Spk. Lt.	Ingr.	Just Dnce. Now	7 Min. Sup. Wrk.	Shp. Up Btl. Run	Zmb. Run
Partic. #											
1	71		78								
2		47						64			
3		75		74							
4	58					68					
5							63	77			
6					76					66	

7									57		78
8				58	76						
9			68						78		
10	64			55							
11			43					88			
12	72		72								
13				68					64		
14							62	69			
15		78									77
16	61				75						
17			53						62		
18				73						76	
19					66					57	
20								60			66
21		74					69				
22			66						66		
23	68			66							
24	78						53				
25								67		69	
26		63									58
27			68	58							
28					59				55		
29							66				63
30				60				64			
31			62							67	
32		78							77		
33	61									68	
34			60		62						
35	56	74									
36				65			69				
37					60			66			
38									65		78
39		72				63					

40						72			65		
41						65					76
42						69				74	
43						60	60				
44				78				96			
45	39		41								
46	69		53								
47				72						78	
48				72							79
49			57							81	
50				70		71					
51				74			73				
52			65			62					
53		74							59		
54		68								76	
55	62										74
56		75						69			
57		69							66		

4.2 - Demographics & Initial Analysis

4.2.1 - Participant Demographics

The participant pool was made up of a total of 57 participants throughout the course of the study. Of these 57 participants, 14 of them came through the SONA participant pool. This was much fewer than was anticipated, leading to fewer total data points. The remaining 43 participants came among the peers of the researchers. Of the 57 participants, 11 of them, or about 20% were female. The other 46 were male. Finally, the participants ranged in age from 18-22. During the pre-questionnaire, the participants were asked if they had any previous exergaming experience. From this, the researchers were able to see that 56% of participants had any previous experience with exergames.

4.2.2 - Initial EEQ Analysis

The mean score of the EEQ across all games was 67.1. The median e-score gathered from the questionnaire was 67.5. Due to the mean and median being close to one another, it can be seen that there should be a rather even distribution of participants who scored their games positively and negatively. It does not seem that all participants loved each game they had played and it does not seem the opposite extreme either.

Continuing off that, the maximum e-score gathered was 96, and the minimum was 39. This gives our dataset a rather wide range of 57. The maximum observed e-score, 96, was a bit of an outlier, with the next highest score being an 88, followed by an 81. The data does begin to cluster again following that, however. Most of the data points gathered are clustered within the range of 60-75. This may be visualized in the box and whisker plot shown in Figure 4.2.2.1.

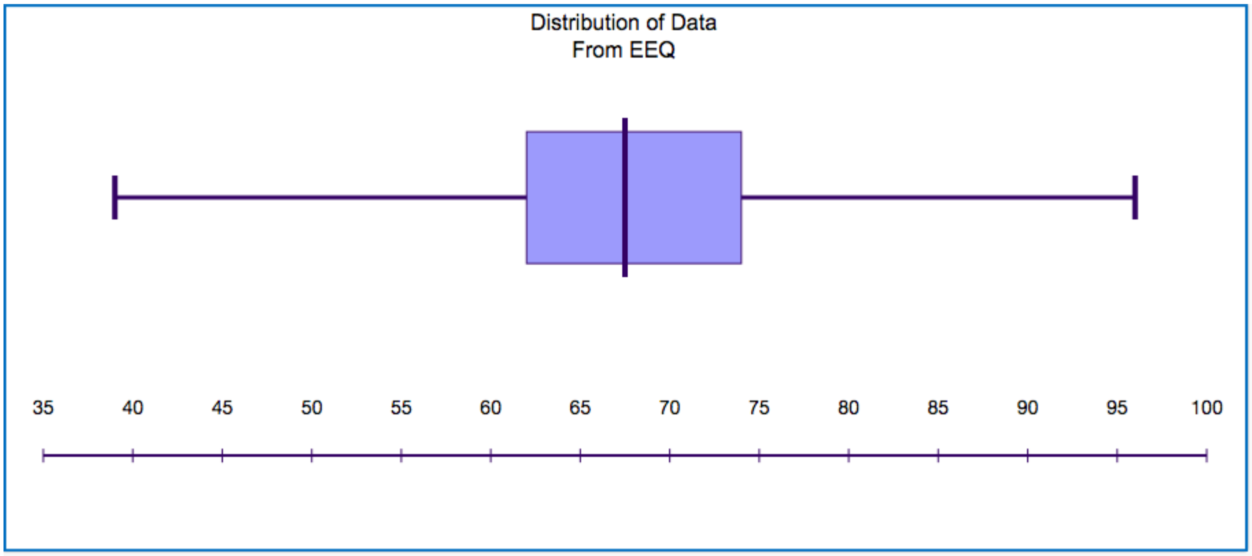


Figure 4.2.2.1: Box and Whisker Plot of Data

4.3 - Visualizations & Analysis

4.3.1 - Chord Diagrams

Chord diagrams were put together in order to show the relationships between the different genres of exergames. The data described in each diagram is found in Table 4.3.1.1 below. The table displays how many times each category was tested against each other, how many times the participant scored both games “positive” or “negative”, as well as how many times the participant scored one game “positive” and the other “negative”. The meaning of positive and negative in this instance is further explained below.

Table 4.3.1.1: Chord Diagram Data

	Sports + Action	Sports + Adventure	Sports + Static	Sports + Mobile	Action + Adventure	Action + Static	Action + Mobile	Adventure + Static	Adventure + Mobile	Static + Mobile
Times Tested	7	5	5	5	6	7	5	5	5	4
Positive-Positive	3	1	2	1	2	3	3	1	1	1
Negative-Negative	3	4	2	2	3	2	2	1	3	1
Positive-Negative	1	0	1	2	1	2	0	3	1	2
Similar	4	1	3	3	3	5	3	4	2	3

In each chord diagram, the chords represent a connection between the two categories. The larger the width of the chord diagram, the more of a relationship there was between the two categories. For these diagrams, each participant's e-score was compared to the mean for its respective game. If they rated it above the mean, they were given a positive, if it was below the mean, they were given a negative. Since each participant played two games, the relationship fell into the category of positive-positive, positive-negative, or negative-negative. Each chord is color coded and hatched with respect to a certain category. The colors do not signify anything in particular, they are merely a visual aide to help distinguish between each category. For each of the following graphs, Sports is represented by the green dash pattern, Action is represented by the gold diagonal pattern, Adventure is represented by the orange sand pattern, Static is represented by the purple brick pattern, and finally, Mobile is represented by the red cross pattern. It may also be noted that there are no red cross pattern chords. This is due to the fact that the Mobile category was the last input for the chord diagram, so the relationship with every other category was already portrayed in their respective sections.

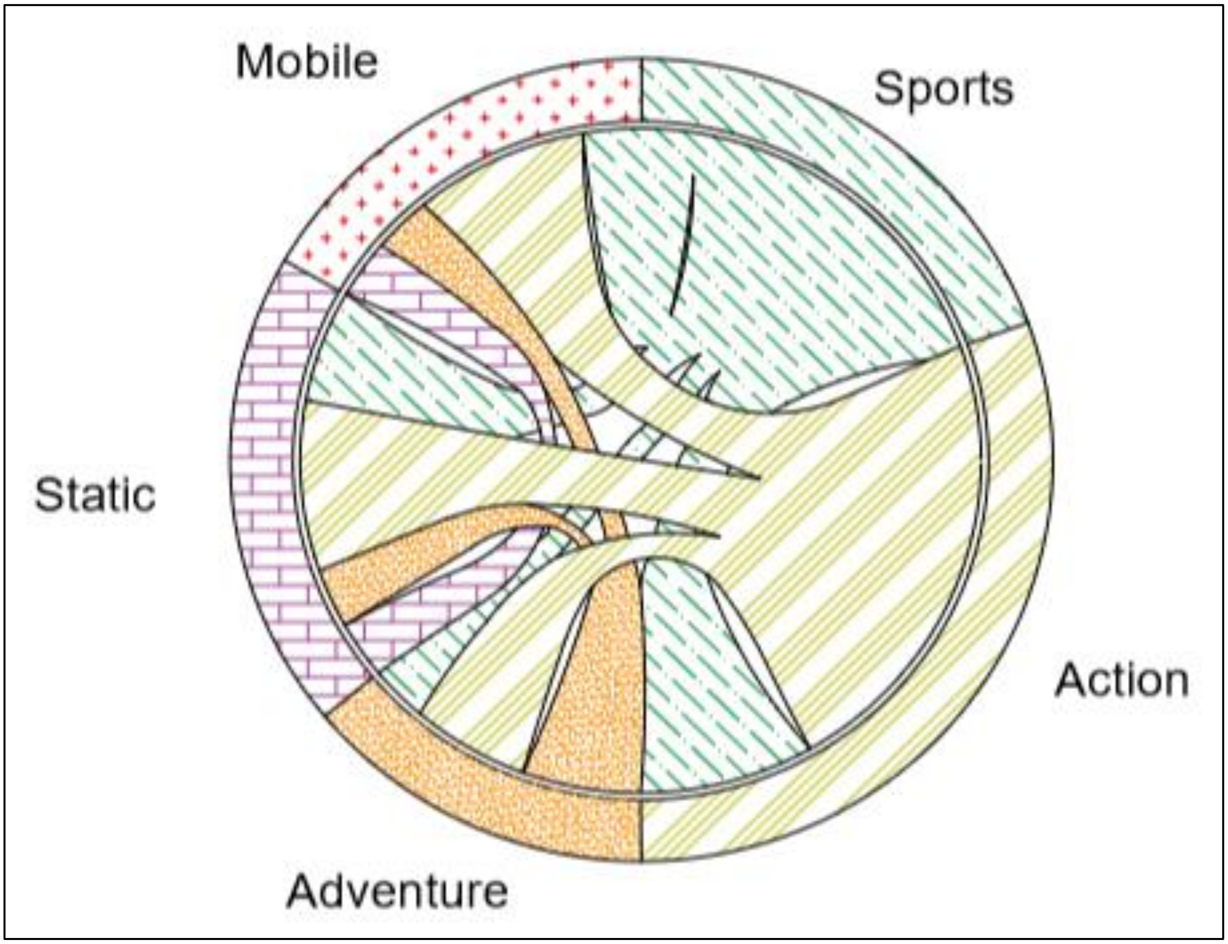


Figure 4.3.1.1: Positive-Positive Chord diagram

Figure 4.3.1.1 depicts the positive-positive chord diagram. Each participant played two exergames. If a participant played a sports and action exergame and scored them both above the respective games mean e-score, then a chord between those two categories was drawn. From this diagram, it can be concluded that sports and action games give similar enjoyment. Another few notable connections are between Static and Action as well as Action and Mobile. There is a lesser connection between Static and Sports as well. This shows that there is a correlation between these two types, but it is not as strong as the previous types mentioned. This diagram also shows that there is no positive connection between mobile and static games, as well as mobile and action games. The other genres each had limited connections between each other, which shows that the categories may produce similar enjoyment to some, but not to others. This diagram shows positive correlations.

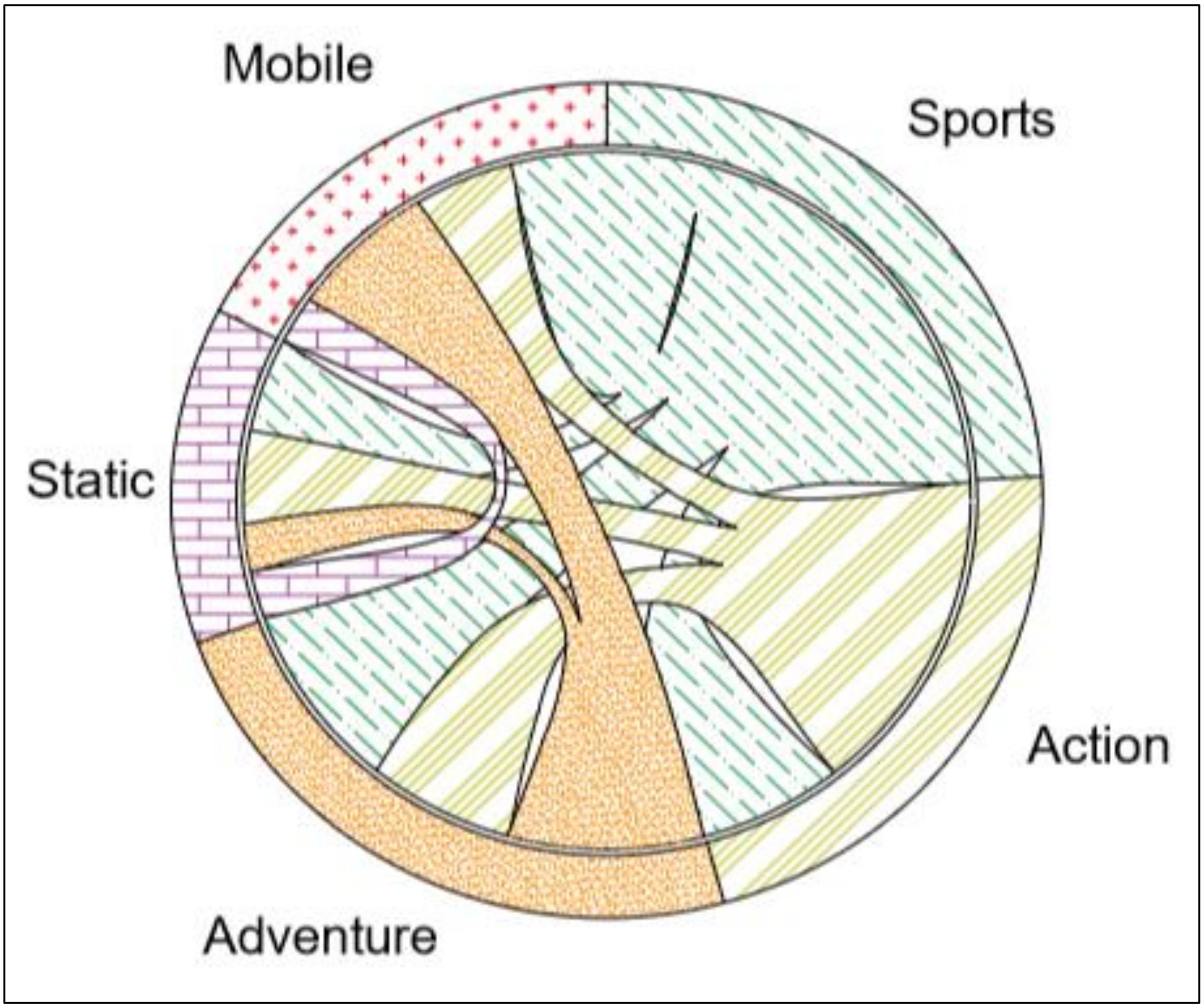


Figure 4.3.1.2: Negative-Negative Chord diagram

Figure 4.3.1.2 depicts the negative-negative chord diagram. Each participant played two exergames. From this diagram, it can be concluded that sports and mobile games give similar enjoyment. A few weaker combinations seem to be Static and Action, Mobile and Static, and Static and Adventure. This shows that there is a correlation between these two types, but it is not as strong as the previous combinations mentioned. This diagram also shows that there is no negative connection between sports and adventure games, as well as mobile and action games. The other genres each had limited connections between each other, which shows that the categories may produce similar enjoyment to some, but not to others. This diagram depicts negative correlations.

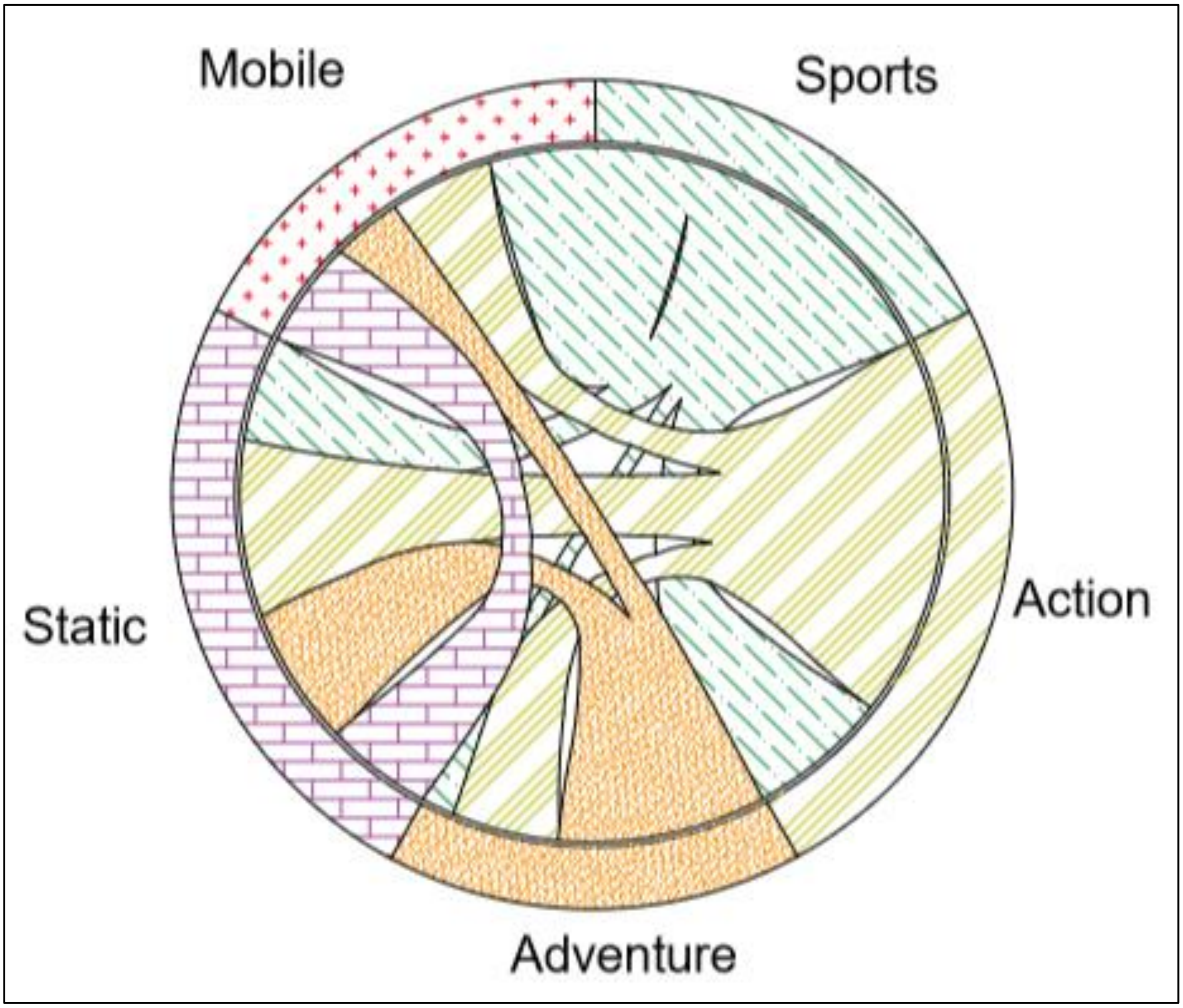


Figure 4.3.1.3: Combined PP-NN Chord diagram

Figure 4.3.1.3 depicts the combined data of the positive-positive and the negative-negative chord diagrams. Judging by this diagram, the research team can look at the bigger picture of genre correlations. Whether a participant liked both games in each genre or hated both games in each genre, this still shows a correlation. This depiction shows that Static and Action games show a strong correlation, as well as sports and mobile games. A multitude of other genre combinations show weaker correlations, such as the sports and action combination and the static and adventure combination. Again, this figure cannot help define whether the correlation between genres is positive or negative, but it does help show if there is a correlation at all.

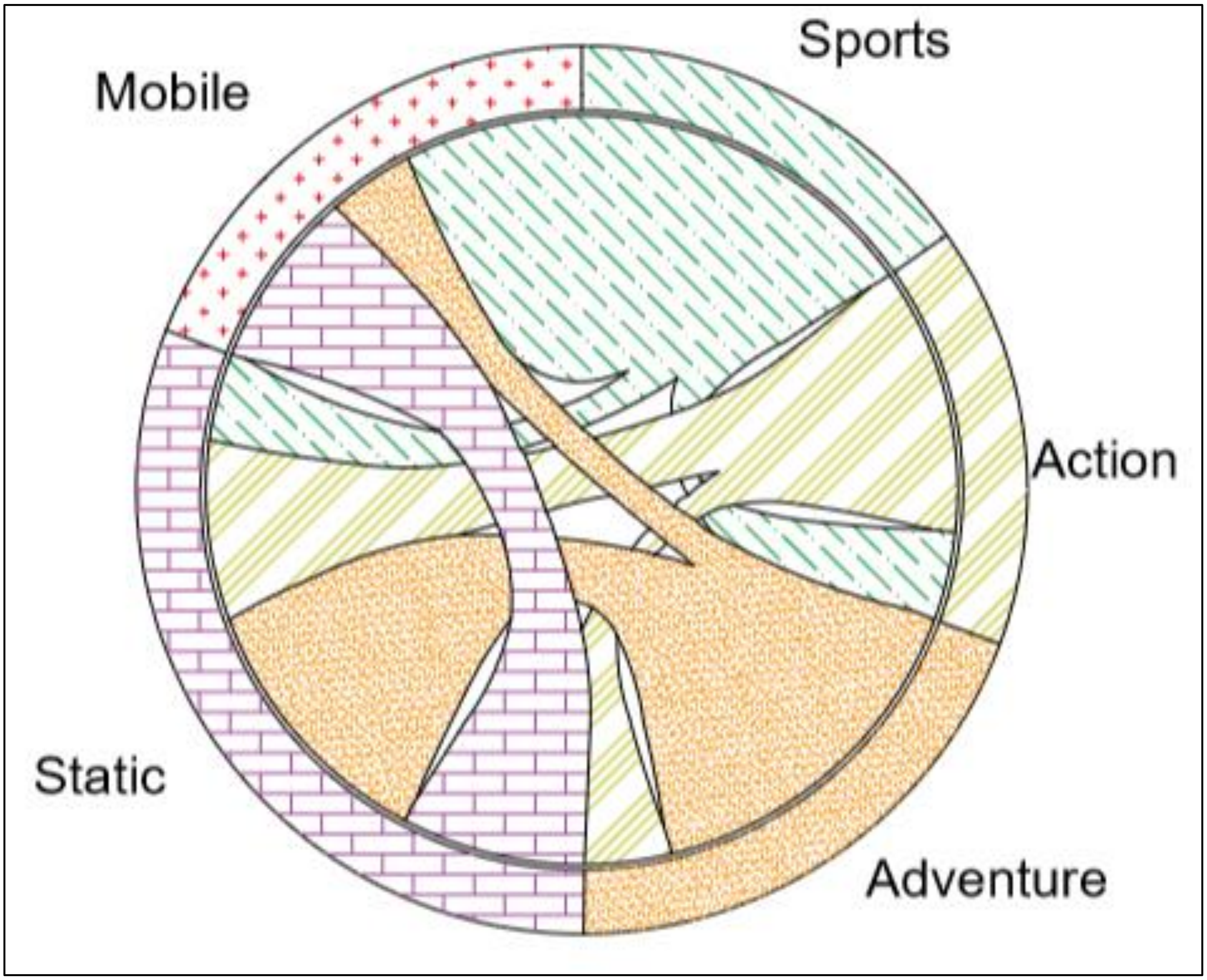


Figure 4.3.1.4: Positive-Negative Chord diagram

Figure 4.3.1.4 depicts the positive-negative chord diagram. This diagram depicts the users that scored a positive in one game and a negative in the other. This could mean one of two things. The first being that there is no correlation between the two categories. The second being that the data may have been skewed due to having such a small participant pool. Since this data is driven by the mean e-score for a particular game, more participation resulted in a much more accurate and precise mean.

This enabled the research team take into account other factors as to why these genres were in a grey area. One factor could be the EEQ. Not every question in the EEQ pertains to enjoyment. For example, question number 9 asks the user if they consider the game “exercise”. This does not necessarily pertain to how much enjoyment the user felt. For a game like *7 Minute Superhero Workout* [1], which is legitimately guiding you through a workout, this question may not help the research team determine whether the participant enjoyed the game or not. Questions such as these may push the e-score above or below the mean, resulting in a possible false positive or false negative. Another variable that could invalidate the data was whether a game was just not good in general. If a participant loves action games, and was given a poorly put together action game, they may not score it very high at all. Some of the lower end games used in this study may have skewed the results a bit.

4.3.2 - Relationship Network

Relationship networks are programmed using the d3 JavaScript library. Each node of the network represented a classification. The lines between each pair of nodes represents the strength of the relationship between them. A thick line represents a strong relationship, where two connected nodes produce similar e-scores. A thin line represents a weak relationship, where two connected nodes produce dissimilar e-scores. Lines are created by averaging the difference of e-score between two classifications during an experiment that tests those two classifications.

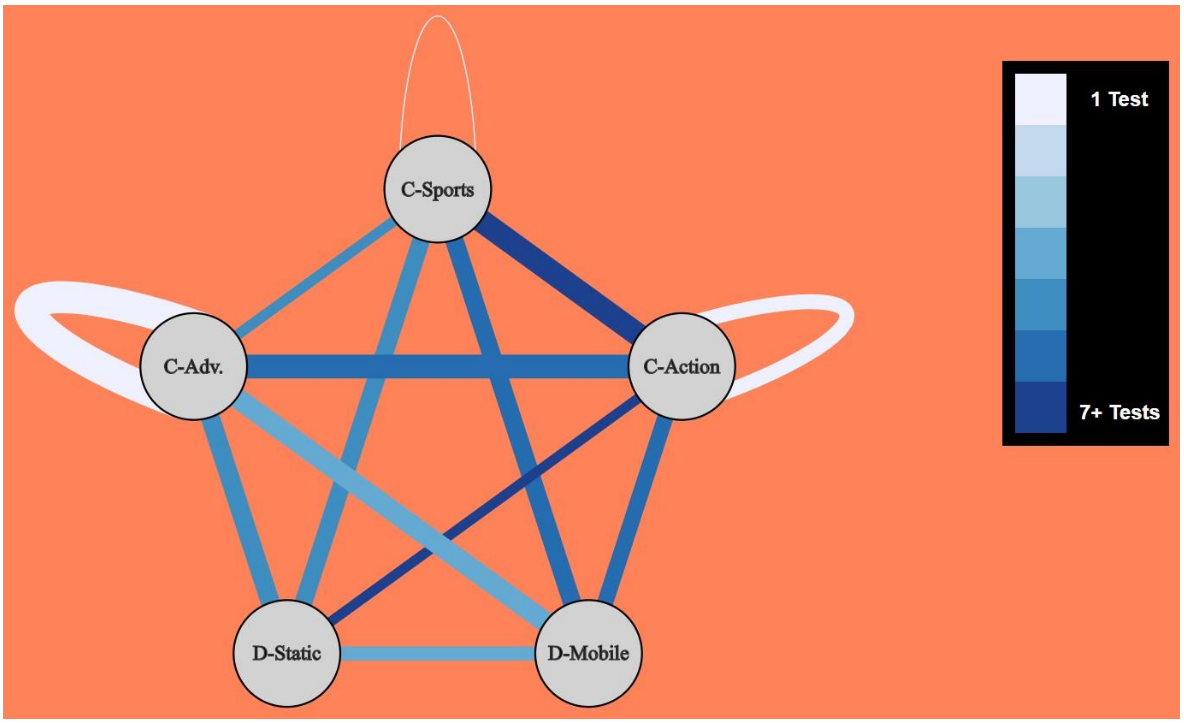


Figure 4.3.2.1: Relationship Network between Classifications

Figure 4.2.2.1 depicts a relationship network between the classifications of the study. The thickest line is actually between games of the same Control-Adventure classification. Although one single experiment ran with a combination of two different Control-Adventure games, the two games received the same e-score from the participant. Even though this suggests a very strong relationship, more data is needed to ascertain the relationship between games of the Control-Adventure classification. The thick line between Control-Sports and Control-Action suggests a strong relationship. Especially as it was a combination tested 7 times. Users that receive a certain enjoyment from Control-Sports are likely to receive similar enjoyment from Control-Action. Interestingly enough, even though Control-Sports is strongly linked to Control-Action, and Control-Action is strongly linked to Control-Adventure, Control-Adventure has a weak relationship with Control-Sports. This suggests that the Control type, the ability to control an in-game character, is not a good springboard for recommendations. Instead, a closer look into the gameplay is required. Perhaps Control-Adventure's playstyle is difficult to relate to (controlling a character walking about a fantasy world) compared to Control-Sport's playstyle, controlling a character in the thick of it (playing a well-known sport). Control-Sport's has the weakest relationship with itself, though with only one test, there is no significant evidence to support the theory. Perhaps Control-Sports as a classification is too weak. Thematically, *Motion Sports* [16] and *Motion Tennis* [17] match up, but gameplay-wise, *Motion Sports* has the player set the smart device down and use its camera to capture the player's bodily movements, while *Motion Tennis* has the player hold the smart device and use it like a racket. Unfortunately, most of the lines between different classifications are similar in thickness to each other. This suggests that e-scores for all classifications cluster across a range that is not very steep. Perhaps the experiment group is too narrow (undergraduates mostly), and always experience a similar baseline of enjoyment. Perhaps smartphone exergames in general are too similar in taste and prevent actual divides in enjoyment. Future analysis would be required.

Relationship Network Build:

First each e-score difference between classifications for each experiment was tallied and the average difference between each classification combination was calculated. This is shown in Table 4.3.2.2 and Table 4.3.2.3.

Table 4.3.2.2: Average Differences

AVG. DIFF.	Control - Sports	Control - Action	Control - Adventure	Directed - Static	Directed - Mobile
Control - Sports	18.00	5.29	12.60	8.40	8.00
Control - Action	5.29	10.00	4.83	12.86	9.33
Control - Adventure	12.60	4.83	0.00	7.60	6.75
Directed - Static	8.40	12.86	7.60	N/A	10.50
Directed - Mobile	8	9.33	6.75	10.50	N/A

Table 4.3.2.3: Count of each Combination

COUNT	Control - Sports	Control - Action	Control - Adventure	Directed - Static	Directed - Mobile
Control - Sports	1	7	5	5	6
Control - Action	7	1	6	7	6
Control - Adventure	5	6	1	5	4
Directed - Static	5	7	5	0	4
Directed - Mobile	6	6	4	4	0

Then the average differences were translated to a given scale. The new weighted differences (for line thickness) ensured that the resulting graph was readable. The formula to do so is as follows: (MAX - MIN) * (AVG_DIFF_MAX - AVG_DIFF_MIN) / (AVG_DIFF_MAX - MIN) + AVG_DIFF_MIN. For a readable graph, 25 was chosen for MAX and 1 was chosen for MIN. This is shown in Table 4.3.2.4.

Table 4.3.2.4: Average Differences Translated

AVG. TRANS. DIFF.	Control - Sports	Control - Action	Control - Adventure	Directed - Static	Directed - Mobile
Control - Sports	25.00	8.05	17.80	12.20	11.67
Control - Action	8.05	14.33	7.44	18.14	13.44
Control - Adventure	17.80	7.44	1.00	11.13	10.00
Directed - Static	12.20	18.14	11.13	N/A	15.00
Directed - Mobile	11.67	13.44	10.00	15.00	N/A

Finally, the translated average differences were flipped so that the greatest difference would result in the thinnest line (dissimilar scores indicates weak relationship). This is shown in Table 4.3.2.5.

Table 4.3.2.5: Average Difference Translated Flipped

AVG. FLIP TRANS.	Control - Sports	Control - Action	Control - Adventure	Directed - Static	Directed - Mobile
Control - Sports	1.00	17.95	8.20	13.80	14.33
Control - Action	17.95	11.67	18.56	7.86	12.56
Control - Adventure	8.20	18.56	25.00	14.87	16.00
Directed - Static	13.80	7.86	14.87	N/A	11.00
Directed - Mobile	14.33	12.56	16.00	11.00	N/A

These final values were inputted into an HTML file as line weights, as in Figure 4.3.2.6, in order to draw up the relationship network graph. For a deeper understanding on how the network was drawn, the full code is located in Appendix B.

```
    ],
    "links":
    [
        {"source":0, "target":0, "count":1, "weight":1},
        {"source":0, "target":1, "count":7, "weight":17.95},
        {"source":0, "target":2, "count":5, "weight":8.2},
        {"source":0, "target":3, "count":5, "weight":13.8},
        {"source":0, "target":4, "count":6, "weight":14.33},
        {"source":1, "target":1, "count":1, "weight":11.67},
        {"source":1, "target":2, "count":6, "weight":18.56},
        {"source":1, "target":3, "count":7, "weight":7.86},
        {"source":1, "target":4, "count":6, "weight":12.56},
        {"source":2, "target":2, "count":1, "weight":25},
        {"source":2, "target":3, "count":5, "weight":14.87},
        {"source":2, "target":4, "count":4, "weight":16},
        {"source":3, "target":4, "count":4, "weight":11}
    ]
};
```


Figure 4.3.2.6: Values as Line Weight

Table 4.3.2.7: Recommendation Quality Summarized

REC. QUALITY	Control - Sports	Control - Action	Control - Adventure	Directed - Static	Directed - Mobile
Control - Sports	**No**	Yes	**No**	Yes	Yes
Control - Action	Yes	Yes	Yes	**No**	Yes
Control - Adventure	**No**	Yes	Yes	Yes	Yes
Directed - Static	Yes	**No**	Yes	N/A	Yes
Directed - Mobile	Yes	Yes	Yes	Yes	N/A

Table 4.3.2.7 offers a quick take away from the relationship network in Figure 4.3.2.1. A “Yes” symbolizes that the two classifications produce similar e-scores, so a recommendation would produce similar enjoyment. A “No” symbolizes that the two classifications produce dissimilar e-scores, so a recommendation would produce

dissimilar enjoyment. Whether or not a combination received a “Yes” or “No” was due to the thickness of the lines in the relationship network as seen in Figure 4.3.2.6. Control-Sports and Control-Sports, Control-Sports and Control-Adventure, Control-Action and Directed-Static, were all combinations that with a line thickness under 8.98 and deemed “No” (Recommendations produce dissimilar enjoyment). All thicknesses greater than 8.98 were deemed “Yes” (Recommendations produce similar enjoyment). The reasoning for this cut-off is because half of the most tested greatest line thickness (17.95 “weight” as seen in Figure 4.3.2.6) is 8.98. The reasoning why 25 was not considered the greatest line thickness in this calculation was because that combination was only tested once.

CHAPTER 5: Recommender System

5.1 - System Build

The recommender system was programmed in Python, using Surprise. Surprise is an add-on to the SciPy library, and it focuses on the building and analyzing of recommender systems [9]. The recommender system was adapted from an article written by Maher Malaeb [14]. The system approaches recommendations using collaborative filtering. The collaborative filtering approach recommends items to a user based on the collective past ratings of other users [20]. To implement this approach, the system uses the Singular Value Decomposition (SVD) algorithm.

Such an algorithm attempts to model a complex matrix with a simpler one [13]. For example, Figure 5.1.1 shows a picture of the scientist Feynman, an image composed of 400 row matrices.

Figure 5.1.1: Image of 400 Row Matrices [13]

Figure 5.1.2: Image of 50 Row Matrices [13]

Figure 5.1.2 shows the same image of the scientist Feynman, but composed of only 50 row matrices.

The simple series of matrices (50 row matrices) is not a perfect copy of the more complex series of matrices (400 row matrices), but it is a good approximation. In the case of this recommender system, the complex matrix would be a number of row matrices, one for each user, filled with the e-scores for every single game. The simple matrix was a number of row matrices, one for each user, filled with the two e-scores gained from experimentation. With the SVD algorithm, the recommender system predicted the e-scores of games the user had not played by returning e-scores from an approximated complex matrix. An example of the system making a recommendation is shown in Figure 5.1.3.

```
A:\Python\python.exe "A:/Python IDE/Projects/simpleRecommender/recSystem.py"

Provide me a user (-1 will exit my program), and I will return up to three recommendations: a
Valid input please.

Provide me a user (-1 will exit my program), and I will return up to three recommendations: 0
Sorry, but that user is not within my knowledge.

Provide me a user (-1 will exit my program), and I will return up to three recommendations: 1

For user 1, I would recommend (with an error of give or take 7.22):
Zombies Run for an estimated 68.78 enjoyment score.
Motion Tennis for an estimated 68.50 enjoyment score.
Pokemon Go for an estimated 67.46 enjoyment score.

Provide me a user (-1 will exit my program), and I will return up to three recommendations: -1
Thanks for coming, bye now.

Process finished with exit code 0
```

Figure 5.1.3: Recommender System in Use

5.2 - System Flowchart

Figure 5.2.1 and Figure 5.2.2 goes over a high-level walkthrough of how the recommender system works. For a deeper understanding of how the system works, the full code of the system is found in Appendix A.

Figure 5.2.1: High Level Flow 1

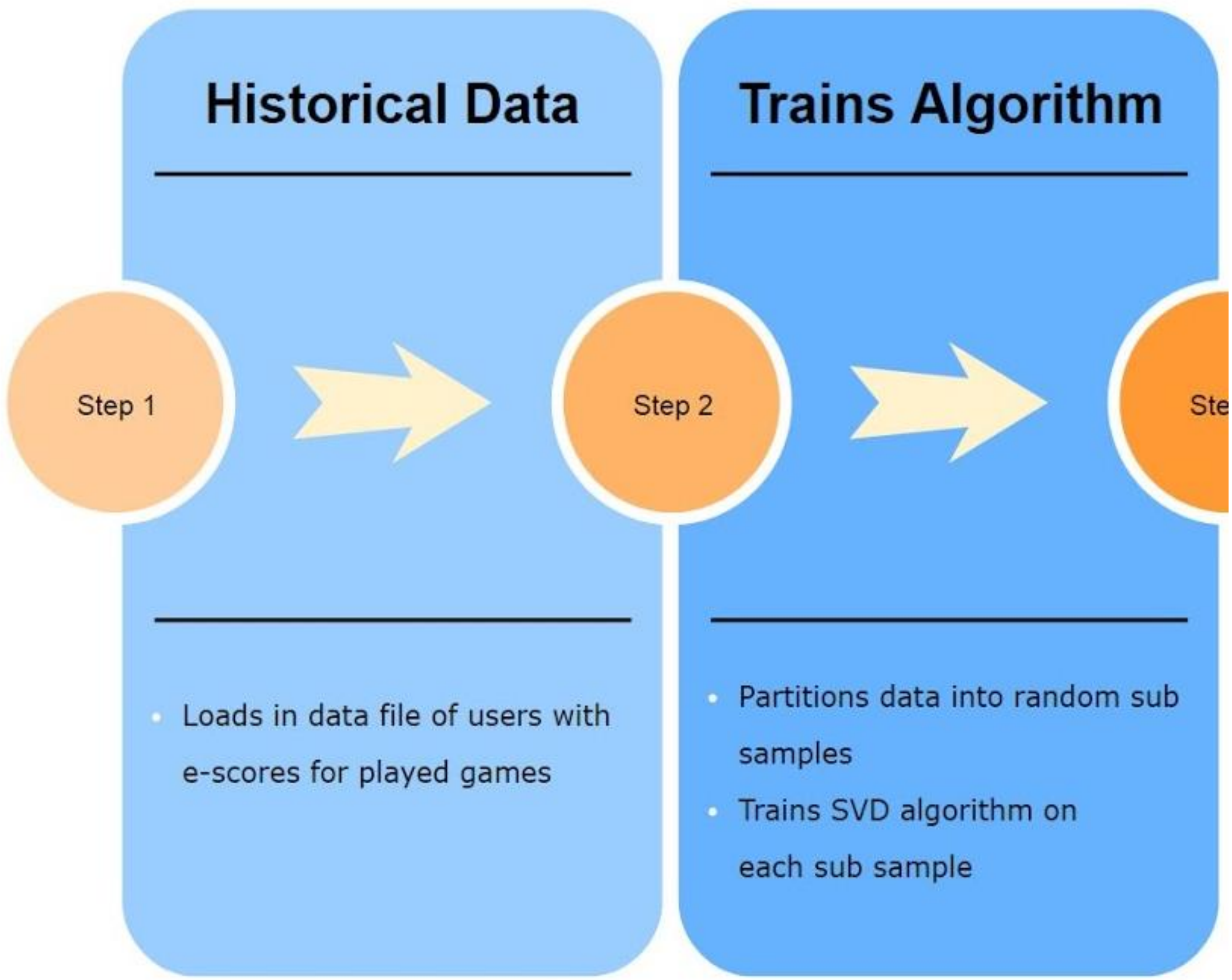
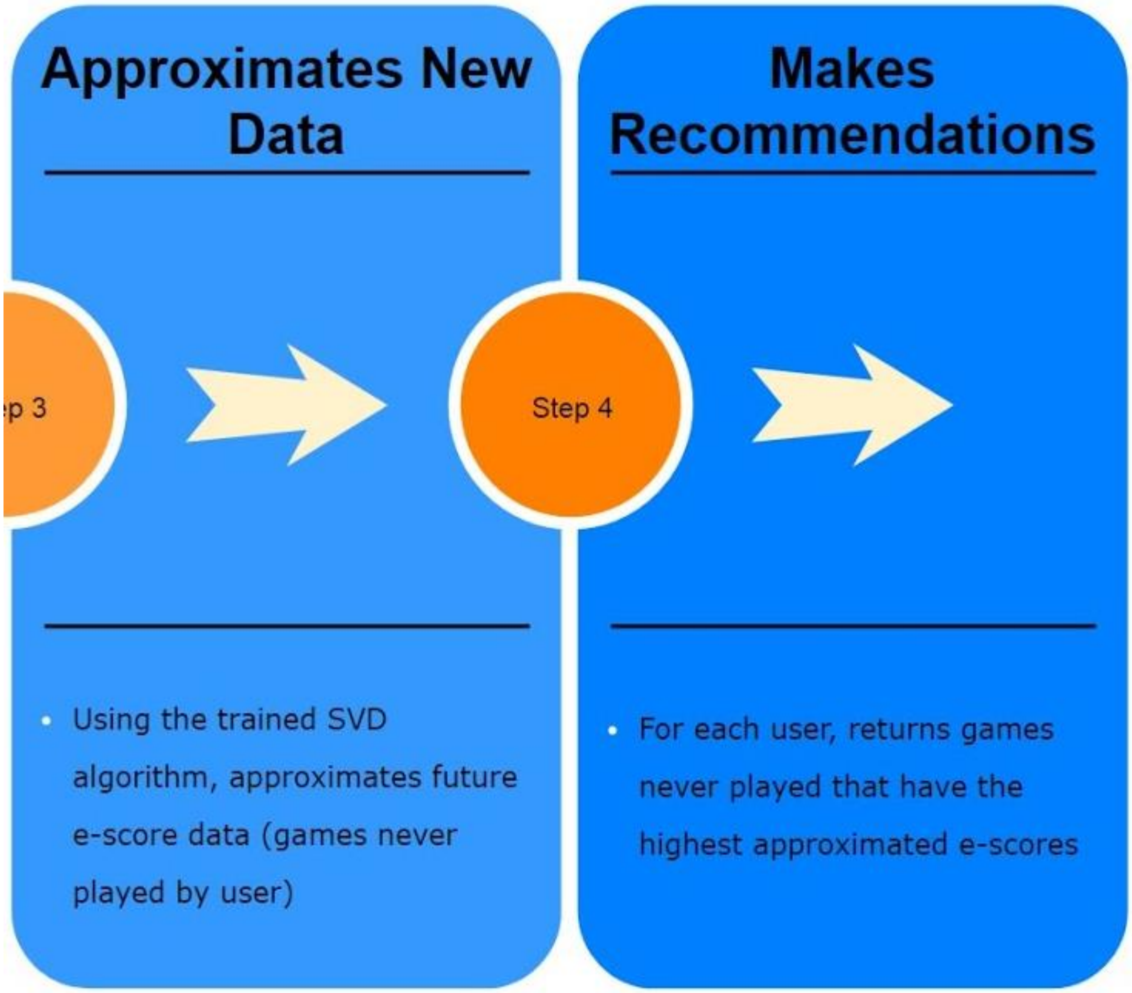


Figure 5.2.2: High Level Flow 2



5.3 - System Validity

In order to test the validity of the recommender system, the research team went back to some of the participants who had already played two exergames each. The recommender system was run with the participant's previous data. The system returns the highest projected e-scores (with a range of error) for up to 3 games a user had never played. So the user was asked to play the exergame with the highest projected e-score (what was basically the top recommendation). Following this, the participant played the game for the standard 15 minutes and took the EEQ afterwards. In order to remove bias, the participants were not told they were testing the validity of a recommender system, only that the researchers needed them to play one more game. Table 5.3.1 shows the data gathered from these tests. In order for the recommender system to pass a test, the actual e-score of a recommended game must fall within the range of the project e-score and error.

Table 5.3.1: Testing the System

Participant #	Top Game	Projected e-score	Actual e-score	Pass/fail
29	Just Dance	69.54 ± 7	74	P
35	Zombies Run	69.33 ± 7	66	P
47	Zombies Run	69.55 ± 7	73	P
48	Just Dance	71.31 ± 8	78	P
41	Just Dance	68.21 ± 8	61	P
20	Shape Up Battle Run	67.85 ± 7	78	**F**

As seen from Table 5.3.1, the recommender system passed 5 out of 6 tests. Therefore, it has a 5/6, or 83% success rate. However, because there was less than 30 tests, the success rate is to be taken with a grain of salt. Further testing is required in the future.

Chapter 6 – Conclusion

6.1 – Conclusion

By completing the 4 steps in the Introduction, the evidence seems to suggest that user preferences regarding smartphone exergames can be extrapolated from prior enjoyment of particular exergames and then used to provide accurate recommendations.

Throughout the study, the team noticed that the EEQ adopted could use some work to further help us find connections between exergames. Some of the questions asked were presented in an awkward tone, so that participants were not quite sure how to answer the question. Another thing that could use some work was the wording of the some of the questions asked. The wording of some negatively phrased questions was confusing, and thus could result in some false data.

Some improvements could be made to the current study. Although Control-Sports seemed like a thematically divisive classification (mutually exclusive between Sports and non-Sports) during classification development, perhaps it was not worth a classification. Themes aside, the gameplay of the two games under Control-Sports (*Motion Sports* [16] and *Motion Tennis* [17]) would fall under Control-Action. Most of the participants were undergraduates. A greater, more diverse, group of participants in the experiments would most likely result in a better range of e-scores. Participants could have also played more exergames per session (current study only had participants play 2 for the sake of time). More data in general would strengthen (or disprove) any drawn conclusions. Experimentation timing could have been held before winter began, in order to test the greatest number of exergames, such as those that required outside play, while remaining safe.

After verifying the validity of the recommender system, it should be possible to incorporate the system in future projects such as the proposed Cypress system [4]. Currently, this project's recommender system has a validated 83% success rate in giving recommendations based on past measured enjoyment. Further testing on the system would make its success rate more statistically relevant. But currently, as long as it receives proper data, it makes a successful recommendation more often than not. Unfortunately, the enjoyment data of this project was gathered through a manual questionnaire, and the proposed Cypress system requires enjoyment data collection to happen automatically, on the mobile phone itself [4]. Future work on the Cypress system would have to incorporate enjoyment measuring onto the mobile phone itself. Future designers would also have to make a distinction on which enjoyment values to base a recommendation on. Cypress recommends a new exergame after player enjoyment decreases [4], but this project only recorded player enjoyment for each game once and had its recommender system make a recommendation as if player enjoyment on past games was static and unchanging.

References

[1] 7 Minute Superhero Workout. Six to Start. 4 July, 2016, Accessed Feb. 28, 2018 from superheroworkoutgame.com/.

[2] Amplitude Analytics "How Pokemon GO Lost the Players, But Won the Game." 27 October 2016, Accessed Feb. 28, 2018 from amplitude.com/blog/2016/10/27/pokemon-go-lost-players-won-game/.

[3] Brockmyer, J. H., Christine, F. M., Curtiss, K. A., McBroom E., Burkhart, K. M., Pidruzny, J. N., The Development of the Game Engagement Questionnaire: A Measure of Engagement in Video Game-playing. The Development of the Game Engagement Questionnaire: A Measure of Engagement in Video Game Playing: Response to Reviews. *Journal of Experimental Social Psychology* 45 (2009) 624–634, 12 March 2009. Web. 10 September 2016.

[4] Emmanuel Agu and Mark Claypool. "Cypress: A Cyber-Physical Recommender System to Discover SmartPhone Exergame Enjoyment", *In Proceedings of the ACM Workshop on Engendering Health with Recommender Systems*, Boston, Massachusetts, USA, September 15, 2016, Accessed Feb. 28, 2018 from cs.wpi.edu/~claypool/papers/cypress-recsys/.

[5] "Financial Results Briefing for the Three-Month Period Ended June 2009" (PDF). Nintendo. 31 July 2009, Accessed Feb. 28, 2018 from nintendo.co.jp/ir/pdf/2009/090731e.pdf#page=7/.

[6] FitFlap Motion. Android. BreakFirst. 26 October, 2015, Accessed Feb. 28, 2018 from play.google.com/store/apps/details?id=com.breakfirst.fitflap&hl=en/.

[7] Fitzgerald A., Huang S., Sposato K., Wang D. "Run for Fun." *Worcester Polytechnic Institute*, Worcester, 2017. Print.

[8] Freeline. Android. BreakFirst. 8 October, 2015, Accessed Feb. 28, 2018 from play.google.com/store/apps/details?id=com.breakfirst.freeline&hl=en/.

[9] Hug, Nicolas. "Surprise Library." Surprise, 2018, Accessed Feb. 28, 2018 from surpriselib.com/.

[10] Ingress. Android. Niantic, Inc. 9 January, 2018, Accessed Feb. 28, 2018 from ingress.com/.

[11] Just Dance Now. Android. Ubisoft Entertainment. 13 December 2017, Accessed Feb. 28, 2018 from justdancenow.com/.

[12] Kim, Tom “In-Depth: Eye To Eye - The History Of EyeToy.” *Gamasutra*. 6 November 2008, Accessed Feb. 28, 2018 from gamasutra.com/php-bin/news_index.php?story=20975./

[13] Liu, Jason. “What Is an Intuitive Explanation of Singular Value Decomposition? (SVD)?” Quora, Sept. 2017, Accessed Feb. 28, 2018 quora.com/What-is-an-intuitive-explanation-of-singular-value-decomposition-SVD/.

[14] Malaeb, Maher. “The Easy Guide for Building Python Collaborative Filtering Recommendation System.” Medium, Medium, 18 Feb. 2017, Accessed Feb. 28, 2018 from medium.com/@m_n_malaeb/the-easy-guide-for-building-python-collaborative-filtering-recommendation-system-in-2017-d2736d2e92a8/.

[15] McDonald, E. “THE GLOBAL GAMES MARKET WILL REACH $108.9 BILLION IN 2017 WITH MOBILE TAKING 42%.” 20 April 2017, Accessed Feb. 28, 2018 from newzoo.com/insights/articles/the-global-games-market-will-reach-108-9-billion-in-2017-with-mobile-taking-42/.

[16] Motion Sports. Android. BreakFirst. 7 January, 2016, Accessed Feb. 28, 2018 from play.google.com/store/apps/details?id=com.breakfirst.kuriosports&hl=en/.

[17] Motion Tennis. Android. Rolocule Games. 27 May, 2015, Accessed Feb. 28, 2018 from play.google.com/store/apps/details?id=com.rolocule.motiontennis&hl=en/.

[18] Pigna, Kris. "Health Games Generate $2 Billion in Worldwide Sales". 1UP.com. 28 June 2009. Archived from the original on 28 July 2012, Accessed Feb. 28, 2018 from web.archive.org/web/20120728181756/http://www.1up.com/news/health-games-generate-2-billion/.

[19] Pokemon Go. Android. Niantic, Inc. 12 February, 2018, Accessed Feb. 28, 2018 from pokemongo.com/.

[20] Prem Melville & Vikas Sindhwani. “Recommender Systems.” *In Encyclopedia of Machine Learning*, Claude Sammut and Geoffrey Webb (Eds), Springer, 2010.

[21] Ricci F., Rokach L., Shapira B. (2011) Introduction to Recommender Systems Handbook. *In: Ricci F., Rokach L., Shapira B., Kantor P. (eds) Recommender Systems Handbook*. Springer, Boston, MA. 2011.

[22] Shape Up: Battle Run. Android. 2014, Accessed Feb. 28, 2018 from support.ubi.com/en-US/Faqs/000021579/Shape-Up-Battle-Run-Gameplay-and-Support-Devices/.

[23] Spectrek Light. Android. 25 July, 2016, Accessed Feb. 28, 2018 from play.google.com/store/apps/details?id=com.spectrekking.light&hl=en/.

[24] Smith, A. “Record Shares of Americans Now own Smartphones, Have Home Broadband” 12 January 2017, Accessed Feb. 28, 2018 from pewresearch.org/fact-tank/2017/01/12/evolution-of-technology/.

[25] Staiano, Amanda E., et al. “Adolescent Exergame Play for Weight Loss and Psychosocial Improvement: A Controlled Physical Activity Intervention.” Obesity, vol. 21, no. 3, 2013, pp. 598–601. *Wiley Online Library*, Accessed Feb. 28, 2018 from onlinelibrary.wiley.com/doi/10.1002/oby.20282/pdf/.

[26] Star, Lawrence. “Exercise, Lose Weight With 'Exergaming'.” *Fox News*. 15 January 2005, Accessed Feb. 28, 2018 from foxnews.com/story/2005/01/18/exercise-lose-weight-with-exergaming.html/.

[27] Stevens M., Moget P., Greef M. H. G., Lemmink K. A. P. M., Rispens P., “The Groningen Enjoyment Questionnaire: A Measure of Enjoyment in Leisure-Time Physical Activity.” *Department of Human Movement Sciences. University of Groningen, Netherlands*. 2000.

[28] Swatman, R “Pokemon GO catches five new world records” 10 August 2017, Accessed Feb. 28, 2018 from guinnessworldrecords.com/news/2016/8/pokemon-go-catches-five-world-records-439327/.

[29] Zombies, Run!. Android. Six to Start. 16 January, 2018, Accessed Feb. 28, 2018 from zombiesrungame.com/.

APPENDIX A

```
# Recommender System
# Adapted from https://medium.com/@m_n_malaeb/the-easy-guide-for-building-
python-collaborative-filtering-recommendation-system-in-2017-d2736d2e92a8
# As well as python api/documentation
# And lots of stack overflow

from surprise import Reader, Dataset, accuracy, SVD
from surprise.model_selection import KFold
import csv

# START: Building underlying recommendation system part

# Define format & rating scale (of e-score)
reader = Reader(line_format='user item rating', sep=',', rating_scale=(0,
100))

# Load data from .csv file
data = Dataset.load_from_file('data.csv', reader=reader)

# Cross-validation (cv) partitions a sample into a training set to train the
model,
# and a test set for evaluation. With k-fold, the original sample is randomly
partitioned into k equal size sub-samples
folds = 5
cv = KFold(n_splits=folds)

# Using the SVD (singular value decomposition) algorithm
# Basically tries to reduce a complex matrix into a simpler one
algo = SVD()

errorAvg = 0  # Average error
for trainset, testset in cv.split(data):

    algo.fit(trainset)  # Trains algorithm
    predictions = algo.test(testset)  # Tests trained algorithm

    errorAvg += accuracy.rmse(predictions, verbose=False)  # Compute Root
Mean Squared Error for this particular set
    errorAvg += accuracy.mae(predictions, verbose=False)  # Compute Mean
Absolute Error for this particular set

errorAvg /= folds * 2  # Average out the error values

# END: Building underlying recommendation system part

# START: Returning recommendations part
```

```
# Dictionary of exergames and their respective score
gamesScores = {'Motion Sports': None, 'Motion Tennis': None, 'FitFlap
Motion': None, 'Freeline': None, 'Pokemon Go': None, 'Spectrek Light': None,
'Ingress': None, 'Just Dance Now': None, '7 Minute Superhero Workout': None,
'Shape Up Battle Run': None, 'Zombies Run': None}
users = {}  # Dictionary of users (each user includes a dictionary of
gamesScores with actual e-scores)
ests = {}  # Dictionary of user estimates (each user includes a dictionary of
gamesScores with estimated e-scores)
recs = {}  # Dict of highest user recommendations (3 arrays that contain an
array of a game and its estimated e-score)

with open('data.csv') as data_csv:  # Opens data in .csv file
    csvReader = csv.reader(data_csv)
    rows = list(csvReader)
    for row in rows:
        users[int(row[0])] = gamesScores.copy()  # Initializes users
        ests[int(row[0])] = gamesScores.copy()  # Initializes user estimates
        recs[int(row[0])] = []  # Initializes user recommendations
    for row in rows:
        users[int(row[0])][row[1]] = row[2]  # Puts actual e-scores into each
user

for user in users.items():  # For each user
    for combo in gamesScores.items():
        if user[1][combo[0]] is not None:  # User has an e-score for this
game, don't put prediction into est dict
            p = algo.predict(str(user[0]), combo[0],
float(user[1][combo[0]]))
            ests[int(p.uid)][p.iid] = -1
        else:  # Put a predicted e-score into ests dict if a particular game
has not been played by user
            p = algo.predict(str(user[0]), combo[0])
            ests[int(p.uid)][p.iid] = p.est

for userId, userItems in ests.items():
    topGame = [None, -1]  # Game with highest predicted e-score
    midGame = [None, -1]  # Game with second highest predicted e-score
    botGame = [None, -1]  # Game with third highest predicted e-score

    for combo in userItems.items():  # Assigns games to top/mid/bot depending
on the estimated e-score
        if combo[1] > topGame[1]:
            botGame = midGame
            midGame = topGame
            topGame = [combo[0], combo[1]]
        elif combo[1] > midGame[1]:
            botGame = midGame
```

```
            midGame = [combo[0], combo[1]]
        elif combo[1] > botGame[1]:
            botGame = [combo[0], combo[1]]
    recs[userId] = [topGame, midGame, botGame]  # Assigns to that user, 3
recommendations (some could be none)

# Handles user input and output
rec_id = 0
while rec_id != -1:
    rec_id = input("\nProvide me a user (-1 will exit my program), and I will
return up to three recommendations: ")

    try:
        rec_id = int(rec_id)
    except ValueError:
        print("Valid input please.")
        continue

    if int(rec_id) not in recs:
        rec_id = int(rec_id)
        if rec_id != -1:
            print("Sorry, but that user is not within my knowledge.")
        else:
            print("Thanks for coming, bye now.")
    else:
        rec_id = int(rec_id)

        if recs[rec_id][0][0] is None:
            print("I have no games to recommend user " + str(rec_id))

        print("\nFor user " + str(rec_id) + ", I would recommend (with an
error of give or take " + "{0:.2f})".format(errorAvg) + ":")
        print(str(recs[rec_id][0][0]) + " for an estimated " +
"{0:.2f}".format(recs[rec_id][0][1]) + " enjoyment score.")
        if recs[rec_id][1][0] is not None:
            print(str(recs[rec_id][1][0]) + " for an estimated " +
"{0:.2f}".format(recs[rec_id][1][1]) + " enjoyment score.")
        if recs[rec_id][2][0] is not None:
            print(str(recs[rec_id][2][0]) + " for an estimated " +
"{0:.2f}".format(recs[rec_id][2][1]) + " enjoyment score.")

# START: Returning recommendations part
```

APPENDIX B

```
<!-- Relationship Network Code (Modified some of the tabbing from the
original code so that lines of code would fit better)-->

<!DOCTYPE html>
<meta charset="utf-8">

<script src="http://d3js.org/d3.v2.min.js?2.9.3"></script>

<style>

.link
{
    stroke: #000;
    fill: none;
    stroke-width: 2px;
}

.node circle
{
    stroke: #000;
    stroke-width: 1.5px;
}

.node text
{
    stroke: #333;
    cursos: pointer;
}

body
{
    background-color: coral;
}

</style>

<body>
<script>

var width = 1280,
    height = 720;

var color = d3.scale.category20();

var radius = d3.scale.sqrt()
```

```
  .range([0, 6]);

var svg = d3.select("body").append("svg")
  .attr("style", "outline: thick solid black;") // Used to see the canvas
bounds
  .attr("width", width)
  .attr("height", height);

// START: Legend construction
var legendX = width/2 + 440,
    legendY = height/2 - 300;
var box = svg.append('rect')
  .attr("x", legendX)
  .attr("y", legendY)
  .attr("width", 130)
  .attr("height", 300)
  .attr("fill", "#000");
svg.append('rect')
  .attr("x", legendX + 10)
  .attr("y", legendY + 10)
  .attr("width", 40)
  .attr("height", 40)
  .attr("fill", "#eff3ff");
svg.append("text")
  .attr("x", legendX + 91)
  .attr("y", legendY + 35)
  .attr('fill', 'white')
  .attr("text-anchor", "middle")
  .attr("font-weight", "bold")
  .attr("font-family", "sans-serif")
  .text("1 Test");
svg.append('rect')
  .attr("x", legendX + 10)
  .attr("y", legendY + 50)
  .attr("width", 40)
  .attr("height", 40)
  .attr("fill", "#c6dbef");
svg.append('rect')
  .attr("x", legendX + 10)
  .attr("y", legendY + 90)
  .attr("width", 40)
  .attr("height", 40)
  .attr("fill", "#9ecae1");
svg.append('rect')
  .attr("x", legendX + 10)
  .attr("y", legendY + 130)
  .attr("width", 40)
  .attr("height", 40)
  .attr("fill", "#6baed6");
```

```
svg.append('rect')
  .attr("x", legendX + 10)
  .attr("y", legendY + 170)
  .attr("width", 40)
  .attr("height", 40)
  .attr("fill", "#4292c6");
svg.append('rect')
  .attr("x", legendX + 10)
  .attr("y", legendY + 210)
  .attr("width", 40)
  .attr("height", 40)
  .attr("fill", "#2171b5");
svg.append('rect')
  .attr("x", legendX + 10)
  .attr("y", legendY + 250)
  .attr("width", 40)
  .attr("height", 40)
  .attr("fill", "#084594");
svg.append("text")
  .attr("x", legendX + 60)
  .attr("y", legendY + 279)
  .attr('fill', 'white')
  .attr("font-weight", "bold")
  .attr("font-family", "sans-serif")
  .text("7+ Tests");
// END: Legend construction

var force = d3.layout.force()
  .size([width, height]);

var cW = width/2, // Center Width
    cH = height/2; // Center Height

var graph =
{
  "nodes": // Pentagram formation
  [
    {"x": cW, "y": cH - 200, "fixed": true, "name":"C-Sports","size":48},
   {"x": cW + 200*Math.cos(18 * (Math.PI/180)), "y": cH - 200*Math.sin(18 *
(Math.PI/180)), "fixed": true, "name":"C-Action","size":48},
    {"x": cW + 200*Math.cos(-198 * (Math.PI/180)), "y": cH - 200*Math.sin(-
198 * (Math.PI/180)), "fixed": true, "name":"C-Adv.","size":48},
    {"x": cW + 200*Math.cos(-126 * (Math.PI/180)), "y": cH - 200*Math.sin(-
126 * (Math.PI/180)), "fixed": true, "name":"D-Static","size":48},
    {"x": cW + 200*Math.cos(-54 * (Math.PI/180)), "y": cH - 200*Math.sin(-54
* (Math.PI/180)), "fixed": true, "name":"D-Mobile","size":48}
    ],
```

```
  "links":
  [
    {"source":0, "target":0, "count":1, "weight":1},
    {"source":0, "target":1, "count":7, "weight":17.95},
    {"source":0, "target":2, "count":5, "weight":8.2},
    {"source":0, "target":3, "count":5, "weight":13.8},
    {"source":0, "target":4, "count":6, "weight":14.33},
    {"source":1, "target":1, "count":1, "weight":11.67},
    {"source":1, "target":2, "count":6, "weight":18.56},
    {"source":1, "target":3, "count":7, "weight":7.86},
    {"source":1, "target":4, "count":6, "weight":12.56},
    {"source":2, "target":2, "count":1, "weight":25},
    {"source":2, "target":3, "count":5, "weight":14.87},
    {"source":2, "target":4, "count":4, "weight":16},
    {"source":3, "target":4, "count":4, "weight":11}
  ]
};

function numTests(d) // Color gets darker (Almost White Blue to Really Dark
Navy) the more tests are performed
{
  if(d.count == 1)
  {
      return "#eff3ff";
  }
  else if(d.count == 2) // Doesn't occur in our data
  {
      return "#c6dbef";
  }
  else if(d.count == 3) // Doesn't occur in our data
  {
      return "#9ecae1";
  }
  else if(d.count == 4)
  {
      return "#6baed6";
  }
  else if(d.count == 5)
  {
      return "#4292c6";
  }
  else if(d.count == 6)
  {
      return "#2171b5";
  }
  else if(d.count >= 7) // Greater than or equal to 7 (Our data only has 7
tests max)
  {
```

```
      return "#084594";
  }
}

var drawGraph = function(graph) // Displays graph
{
  force // Force thingy, function that steps each tick
      .nodes(graph.nodes)
      .links(graph.links)
      .on("tick", tick)
      .start();

  var link = svg.selectAll(".link") // Append links to canvas
      .data(graph.links)
      .enter().append("path")
      .attr("class","link")
      .style("stroke-width", function(d) { return d.weight })
      .style("stroke", numTests);

  var node = svg.selectAll(".node") // Append nodes to canvas
      .data(graph.nodes)
      .enter().append("g")
      .attr("class", "node")
      .call(force.drag);

  node.append("circle") // Node is gray circle
      .attr("r", function(d) { return radius(d.size); })
      .style("fill", "#D3D3D3");

  node.append("text") // Text in nodes
      .attr("text-anchor", "middle")
      .attr("y", 5)
      .text(function(d) { return d.name });

  function tick()
  {
      link.attr("d", function(d)
      {
          var x1 = d.source.x,
          y1 = d.source.y,
          x2 = d.target.x,
          y2 = d.target.y,

          // Defaults
          drx = 0,
          dry = 0,
          xRotation = 0, // degrees
          largeArc = 0, // 1 or 0
          sweep = 0, // 1 or 0
```

```
          width = 1280,
          height = 720;

          // Self referencing arc
          if ( x1 == x2 && y1 == y2 )
          {
          // Needs to be 1.
          largeArc = 1;

          // Angle of node from center of the circle
          var angle = Math.abs((Math.atan2(y1 - height/2, x1 -
width/2))*(180/Math.PI));

          // Line to node from center
          var m = (y1 - height/2) / (x1 - width/2); // m = (y - y2) / (x -
x2)
          var b = (height/2) - m*(width/2); // b = y - m(x)

          // Perpendicular to above line
          var m_p = -1 / (m); // Perpendicular lines have negative reciprocal
slopes
          var b_p = (y1) - m_p*(x1); // b = y - m(x)

          var anchor = 30;

          // New X on perpendicular line
          x2 = x1 + anchor/Math.sqrt(1 + m_p*m_p);
          y2 = m_p*x2 + b_p;

          x1 = x1 - anchor/Math.sqrt(1 + m_p*m_p);
          y1 = m_p*x1 + b_p;

          // Used to see the perpendicular line
          /*
          svg.append("line")          // attach a line
              .style("stroke", "black")  // colour the line
              .attr("x1", x1)     // x position of the first end of the line
              .attr("y1", y1)      // y position of the first end of the line
              .attr("x2", x2)     // x position of the second end of the line
              .attr("y2", y2);    // y position of the second end of the line
              */

          if(d.source.name == "C-Sports")
          {
              xRotation = 90;
              sweep = 1;
              drx = 45;
              dry = 10;
          }
```

```
            else if(d.source.name == "C-Action")
            {
                xRotation = 72;
                sweep = 1;
                drx = 10;
                dry = 45;
            }
            else if(d.source.name == "C-Adv.")
            {
                xRotation = 200;
                sweep = 1;
                drx = 45;
                dry = 10;
            }
        }
        return "M" + x1 + "," + y1 + "A" + drx + "," + dry + " " + xRotation +
"," + largeArc + "," + sweep + " " + x2 + "," + y2;
  });
      node.attr("transform", function(d) { return "translate(" + d.x + "," +
d.y + ")"; });
}
};

drawGraph(graph);

</script>
```