MASTER

Behavioral change visualization for GameBus

Wijnhoven, J.F.W.

Award date:
2020

EINDHOVEN UNIVERSITY OF TECHNOLOGY

MASTER THESIS

Behavioral change visualization for GameBus

Author:
J.F.W. Wijnhoven

Supervisor:
dr.ir. H.M.M. van de Wetering

A thesis submitted in fulfillment of the requirements for the degree of Master of science

in the

Visualization group
Department of Mathematics and Computer science

January 20, 2020

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Abstract

Mathematics and Computer science
Department of Mathematics and Computer science

Master of science

Behavioral change visualization for GameBus

by J.F.W. Wijnhoven

This graduation project aims to visualize behavioral change of people who use the mobile health application GameBus. GameBus is an application which promotes healthy behaviors (e.g. physical, social and cognitive activities) according to the age and interests of the user. Currently, analyzing the data of GameBus is time-consuming since there is no tool available to investigate and explore the behavior of GameBus users. Hence, a tool was created combining various visualization techniques to capture multiple types of behavior.

The implemented visualization tool, called GameBus Insight, allows the data analyst to explore the data of GameBus and see if a group of people have changed their behavior and why they have changed their behavior. This is done by comparing the behavior of multiple groups of people over multiple time periods.

Behavior can be measured in many ways. GameBus Insight uses adaptable measurements of behavior which make the tool flexible and future proof. Furthermore, this allows applications with a similar data structure as GameBus to also be investigated with GameBus Insight.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Health is an important element in today's modern lifestyle. According to "The Epidemiology of Obesity" [1], obesity, along with overweight, is affecting over a third of the world's population. The authors predict that if this trend continues that in 2030 38% of the world's adult population will be overweight and another 20% will be obese. The increase in obesity and other chronic health conditions cause some of the heaviest use of medical services. Chronic diseases are one of the most expensive health diseases, but also the most preventable [2].

With health closely linked to behavior, the interest in behavioral change is increasing rapidly. With smartphones and tablets becoming an essential part of people's life, a lot of "e-Health" applications are built for showing health behavior and to motivate users into changing their behavior. FitBit [3] is one of the most popular "e-Health" applications. FitBit can track users all-day activity, workouts, sleep, diets, etcetera, and show the progress in different kinds of visualizations, see Figure 1.1.

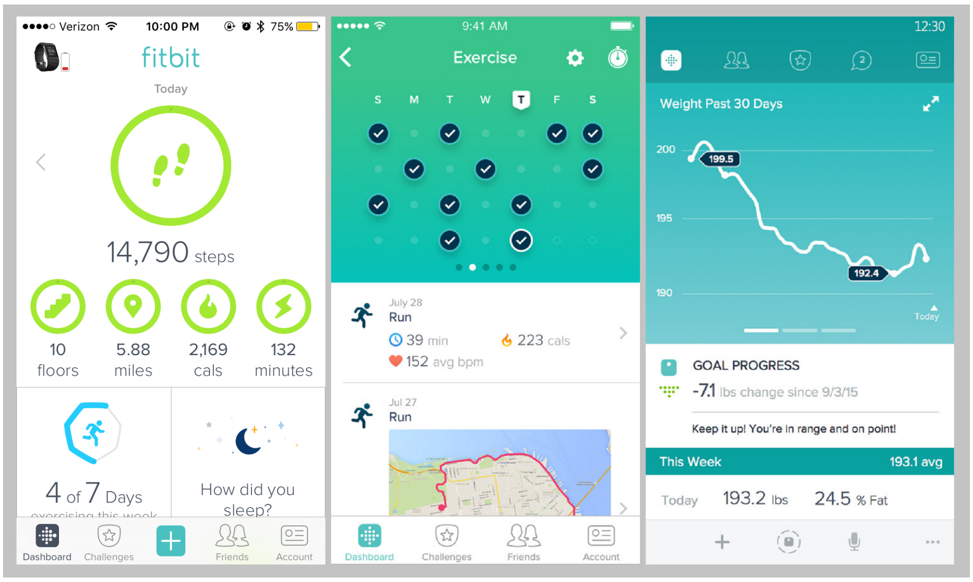


FIGURE 1.1: Three screenshots of Fitbit application.

Most "e-Health" applications, like FitBit, have some limitations. They focus primarily on physical activities and are therefore mostly used by adolescent and adults, ignoring the elderly and children. According to the World Health Organization (WHO) [4], "Health is a state of complete physical, mental and social well-being". This implies that applications like FitBit lack the social and mental components for showing overall well-being. To fill this hole, the Industrial Engineering & Innovation Sciences department of the University of Technology in Eindhoven constructed an app called GameBus [5].

1.1 GameBus

GameBus exists both as a mobile and a desktop application and is meant for creating interactions between friends and family. The mission statement of GameBus is formulated as follows:

"GameBus is a platform that aims at empowering families and friends to be social with one another whilst fostering their health and wellbeing." [5]

FIGURE 1.2: GameBus tracks your progress with the help of external applications or devices [6].

GameBus tracks your progress with the help of external applications or devices. There are three types of activities that are tracked for your progress which are cognitive, physical and social activities. For the cognitive data there are puzzle applications like Griddlers [7] that register the type, the duration and the difficulty level of the puzzle that you did. Social data was in the earlier releases of GameBus mostly gathered with apps like Facebook and Twitter. Nowadays, almost all the social data is gathered by registering when you meet other members or by making a picture of yourself and/or friends. Physical health is tracked by applications like FitBit and Google Fit [8], and can therefore also be tracked with a smart watch. GameBus stores all the activities done by each user. This makes GameBus the knowledge base for this project.

GameBus provides challenges to encourage people to take part in physical, cognitive and social activities. Within a challenge, people can earn points for doing certain activities. What makes it even more interesting, is that you can do these challenges in a group (e.g. of family members, or friends) and compete against other groups. Within GameBus, groups are called circles and every person is at least in one circle which is his own. Data of external applications, like FitBit and GoogleFit, are mapped to GameBus challenges so that people can earn points by using these external applications, see Figure 1.2. At the end of a challenge the group with the highest score earns a virtual badge and optionally also a tangible reward. These prices could come from GameBus, but could also be given from a sponsor that has made their own challenge to gain more reputation.

As an example, in 2019 there was a challenge created by GameBus for the whole month of January called "Healthy January challenge". This challenge has multiple types of activities assigned which can be seen in Figure 1.3. An activity represents what needs to be done by the user to achieve points in that particular challenge. The number of points can differ per activity which can be seen on the right of Figure 1.3. Next to the assigned points you sometimes see a camera icon. These activity types need a picture as prove that the activity has been done.

FIGURE 1.3: Activity types of the "Healthy January challenge".

You can earn points alone or you can earn points with the help of your circle. Playing together is encouraged because you can get more points faster. In this particular challenge, every circle that had above 50 points at the end of January won a price.

GameBus considers users from all ages and motivates the users to do activities according to their age and interests. For example, seniors may be less physical active compared to younger users or in some cases not able to participate in the physical activities, while they can still join in by doing cognitive activities (like memory or crossword puzzles) or social activities (like meeting other members). As seen in the example above, challenges exist of activities. Designing a challenge to provide a simple, healthy and fun gameplay remains a complex task. The creator of a challenge can make up which activities are in the challenge and how many points you can earn from a certain activity. To keep the app fun for the whole family, the points that can be earned for an activity can even differ per age group. By doing this, you encourage a specific age group for doing a type of activity that will get the whole circle more points.

1.2 Research questions

The main objective of this thesis is to visualize behavioral change. We will do this with the help of the event data of GameBus that was generated by users doing activities. Visualizations will be made for a data analyst of GameBus and should give more insight in what way people have changed their behavior and why. Currently, detecting behavioral change patterns for the GameBus data is time-consuming. The main reason for that is because behavior cannot be seen from a single visualization. To examine the change in behavior we need multiple visualizations that are linked to each other to give more insight. Therefore, we propose a visualization tool called "GameBus Insight" as a solution where the data analyst can compare behavior of GameBus users over time. A simple diagram in figure 1.4 shows the solution in the current context. GameBus users log their social, physical and cognitive activities with the help of external applications into the GameBus database. The data analyst can then use our visualization tool, which is connected to the database, in order to achieve new insights about the behavior of the GameBus users. These new insights can help the data analyst to create new challenges.

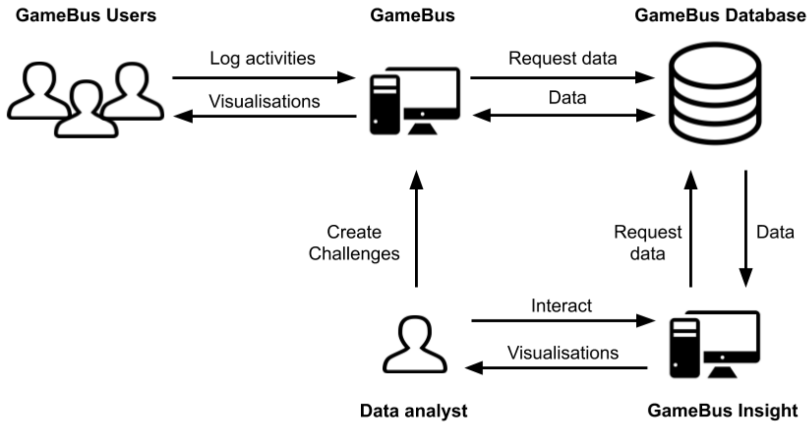


FIGURE 1.4: The visualization tool "GameBus Insight" in the current context of GameBus.

To detect behavioral change patterns, we must first know how we can compare users of GameBus and their behavior over time. This problem can be divided into three subproblems:

1. **What is behavior?**
2. **How can we measure behavior?**
3. **How can we measure behavioral change?**

When we have answers for these questions we can answer the following questions for designing the visualization tool:

1. **How can we visualize behavior?**
2. **How can we visualize behavioral change?**

These problems will be handled in Chapter 2 "Related work" and Chapter 3 "Problem Analysis". Chapter 2 describes how other papers measure behavior and Chapter 3 answers these questions in the context of GameBus.

1.3 Methods

In this project we will make use of the method rapid prototyping [9]. Rapid prototyping is an iterative software development process which exists of 3 steps: prototype, review and refine.

1. **Prototype**: In this step we want to design visualizations according to the requirements and implement the created designs.

2. **Review**: In the review step we gather feedback on the implemented visualizations to see if all requirements are fulfilled.

3. **Refine**: Based on the feedback from the previous step we need to identify the requirements that need to be clarified or redefined. With the redefined requirements we can go back to the prototype step.

There will be multiple review sessions during this project in which we will get feedback on the created visualizations. To verify that our tool works, there will also be an user evaluation study where the data analyst will have to perform different tasks with the visualization tool.

1.4 Scope

During this project, we will create a visualization tool to help with detecting behavioral change patterns in the GameBus event data. The tool will only focus on the physical and social activities of GameBus because there is not enough data gathered for the cognitive activities. With all the information that can be obtained from the GameBus event data, we focus on showing the change in behavior for groups of people. We define the scope further in chapter 3 "Problem analysis" after we gain a better understanding of what behavior is in the GameBus context.

Chapter 2

Related work

In this chapter, we present related work to behavior, behavioral change and visualizations.

2.1 Behavior

What is behavior? Behavior is in very broad terms the range of actions made by individuals, organisms, systems, or artificial entities in conjunction with themselves or their environment [10]. In this thesis we are only interested in the behavior of people. To explain behavior, we use the COM-B model [11]. The COM-B model exists of three elements that influence behavior: Capability, Opportunity and Motivation (see Figure 2.1). In this model, Capability, Opportunity and Motivation interact to generate a certain behavior. The paper "The Behaviour Change Wheel: a new method for characteris-

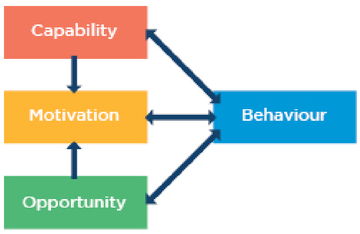


FIGURE 2.1: The COM-B model states that behavior is influenced by 3 elements: Capability, Opportunity and Motivation.

ing and designing behaviour change interventions" [11] describes the three elements in the following way: "Capability is defined as the individual's psychological and physical capacity to engage in the activity concerned. It includes having the necessary knowledge and skills. Motivation is defined as all those brain processes that energize and direct behaviour, not just goals and conscious decision-making. It includes habitual processes, emotional responding, as well as analytical decision-making. Opportunity is defined as all the factors that lie outside the individual that make the behaviour possible or prompt it.". These elements all have influence on behavior but behavior has also influence on these elements. Furthermore, capability and opportunity have impact on the motivation element. To get a better view on what motivates a person we use the self-determination theory [12]. The self-determination theory explains that there are 3 basic psychological needs for humans: Autonomy, Competence and Relatedness.

The paper "Self-Determination Theory" [12] describes the three needs in the following way: "Autonomy: the need to feel free and self-directed, Competence: the need to feel effective, and Relatedness: the need to connect closely with others.".

2.2 Measurements of behavior

To get a better insight in behavior and the measuring of behavior, other papers on behavioral change were examined. A literature review with multiple students was done to fulfill this task. With the help of the literature review we could find metrics that are most useful for measuring behavior.

2.2.1 Approach

With a group of 5 students we formed a group to review behavioral change literature. A total of 339 unique publications were found that could be interesting for review. Of these papers we looked which ones would be included with the help of the following criteria:

- The publication is written in English.
- The publication presents one or multiple empirical studies (i.e. experiments with Humans) where physical activity or healthy eating is promoted.
- Within an experiment, at least one IT-based support mechanism is adopted.
- Participants of an experiment should be based on general population.

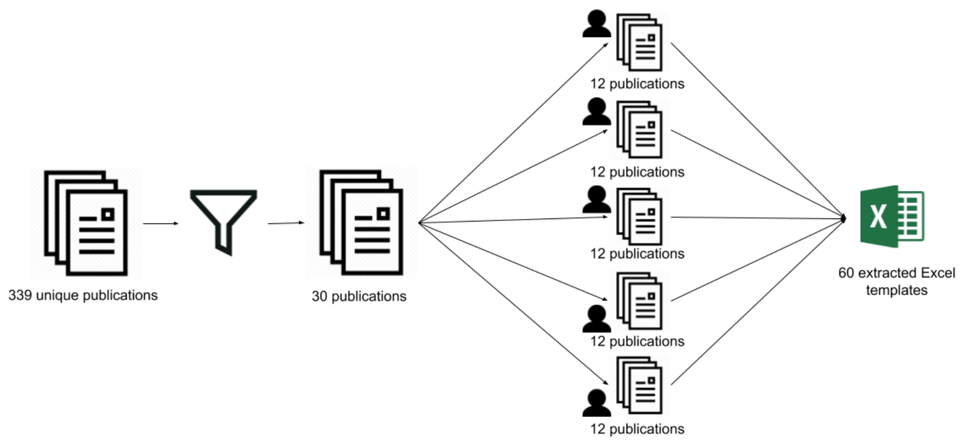


FIGURE 2.2: Process of the literature review. 30 papers that met the requirements were examined by the 5 students to extract the important information.

Figure 2.2 shows the process of the literature review. First, all 339 papers were filtered on the above criteria. This left us with 30 papers that met all the requirements. To decrease the amount of work, each member of the group examined 12 papers such that every paper was examined twice. For ease of comparison, the features from every paper were extracted into an excel template. Every student of the group could use those 60 (every paper was extracted twice) filled in excel templates for their own purpose. In this project, we are most interested in how other papers measure behavior.

2.2.2 Results

All the filled in excel templates were examined on the used measurements of behavior. For this thesis, we would like to get measurements for social and physical activities because these measurements could match with our data. Almost every paper mentioned a measurement for physical activities with most of them using a type of frequency. Some other interesting measurements for physical activities that were found are:

- Metabolic value per hour [20].
- Days per week exercise [21].
- Average hours per day exercise [22].
- Total duration of physical activities [23].
- Measuring physical activity with extra devices (accellerometer [19], bicycle ergometer [24]).

All measurements of behavior for the 30 papers can be found in Appendix A. These measurements are already reduced to a smaller subset by filtering out all the measurements about food. Next, the most used measurements were examined. In Table 2.1 you see the top 5 of the most used measurements.

	Measurement	**Unit**	**Times used**
1	Weight	Kilograms	19
2	BMI	KG(m^2)	13
3	Frequency physical activity	Count (integer)	10
4	Steps per day	Steps	4
5	Website logins	Count (integer)	4

TABLE 2.1: Most used measurements by the 30 papers that were reviewed.

2.2.3 Usage

Of all the measurements that are found, we are most interested in the ones that are feasible for this project. Measurements like weight and BMI are not included in our dataset and can therefore not be used. Chapter 3 discusses what measurements will be used for this project to measure behavior.

There were also techniques found in the papers that were used to give more insight into a measurement of behavior. One interesting technique that was found is called Labeling.

Labeling

Labeling is a pre-processing technique. Values are separated with the help of labels. By giving a measurement multiple labels you make it more specific and retrieve more insight. An example for labeling was seen in the paper of Campbell et al [20]. This paper examined the physical activities with two labels: Aerobic activities and strengthening/flexibility activities. Instead of showing the overall frequency in physical activity, it showed the frequency per label. The paper labeled "aerobic activities" and "strengthening/flexibility activities" even further. Examples of labels for Aerobic activities are: walk, swim and jog. Examples of strengthening/flexibility activities are: lift weights and stretching. Physical activity is now divided into multiple labels to give more insight into the change in physical behavior. For a given measurement, all labels need to be measured with the same unit to make labeling possible. In the paper of Campbell et al [20], the unit used for all labels is frequency.

2.3 Behavioral change

Initiating new behaviors and sustaining them can be challenging for individuals. This raises the questions: When is something called behavioral change? And how long does it take to change? According to the paper "Facilitating successful behavior change: beyond goal setting to goal flourishing" from Nowack, K [13], a behavior can become automatic on average between 18 to 254 days. The paper states that this is dependent on the complexity of the behavior and the personality of the individual. The experiment was done for 3 different kinds of behavior: eat behavior, drink behavior (e.g. drinking water) and exercise behavior. Every kind of behavior had around 30 participants who were asked to try a new behavior each day for 84 days. Analysis of all of these behaviors indicated that it took 66 days, on average, for this new behavior to become automatic and natural.

According to BJ Fogg [14], there are 15 ways behavior can change. In Figure 2.3, you see Fogg's behavior grid with the 15 ways behavior can change. The horizontal axis is called "flavor", which indicates the different kinds of behavior. The vertical axis defines the duration of the behavior. There are three types of durations of be-

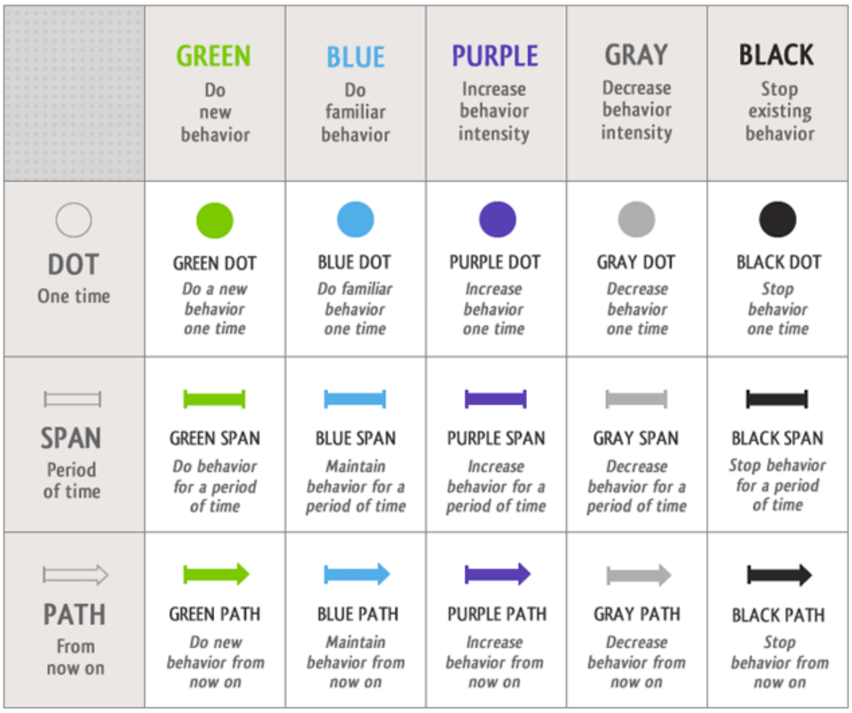


FIGURE 2.3: Fogg's behavior grid that specify 15 ways of how behavior can change [14].

havior: dot (one time), span (period of time), and path (from now on). Dot describes behavior that is done once, span describes behavior that is done for a period of time and path describes a permanent change in behavior. The goal of this model is to give a better definition of behavioral change. It gives a systematic way of thinking about behavioral change. Fogg's behavior grid is a conceptual framework which calls for support by a tool which can meaningfully render the various cases of the grid based on real behavior data. The visualization tool that we will make during this project should be able to show the different kinds of behavioral change that are given by Fogg's behavior grid.

2.4 Existing behavior visualizations

This section describes existing applications that show behavioral statistics for their users. Examples of these applications are FitBit [3] and Google Fit [8] which also contribute to our GameBus dataset. Fitbit and Google Fit show the user visualizations of their activities over a (fixed) time period. Over this time period they show information on duration, steps, frequency, etcetera. Using this information they try to provide the user with behavioral insight by means of comparing their statistics on the current week in comparison to previous weeks or other users. With these statistics, they try to motivate the users to do at least as many activities as those (past data) activities. For instance, a line graph showing the activities of last week in comparison to those of the current week.

This is not desired for our visualization tool because we want to give more insight into why behavior has changed. In this project we will make a visualization tool that is meant for the data analyst of GameBus. Our visualizations should not motivate the user into doing more activities but should give more insight into the behavior of the end-users. Because our target audience differs, our tool can have a steeper learning curve.

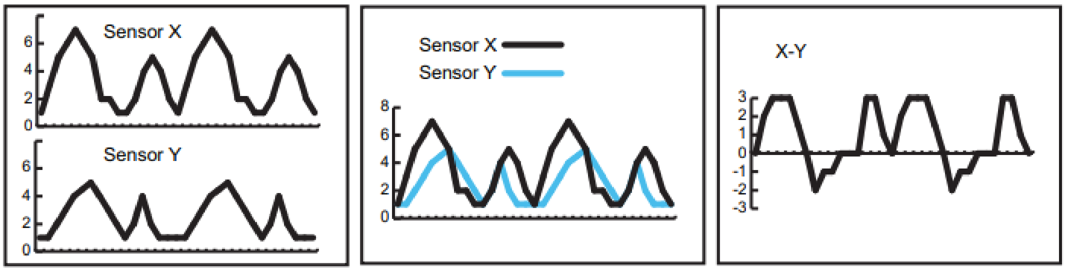
There are also generic tools that can easily visualize your data by connecting your database. Examples of these applications are Tableau [17] and Power BI [18]. The strength of these applications come from being very generic. Everything that is in data can be visualized as the user commands. Visualizations can easily be made with the help of drag-and-drop using the column names of the data. Types of visualization that can be made are: Barchart, Linechart, Pie chart, Bubble chart, Treemaps etcetera.

Power BI and Tableau are great tools to gain insight in your data. However, they are not suited to visualize behavioral change of the GameBus data because they are too generic. To show behavioral change of GameBus users we need a tool that consist of linking visualizations to explore and give a quick insight into the behavior of GameBus users. Furthermore, when using Power BI and Tableau we are limited to the visualizations that they provide. These visualizations are easy to understand but are maybe not the best fit to explain change in behavior. Our tool will be custom made and can therefore be tailored to the exact business needs. The tool will be specialized for the GameBus dataset.

2.5 Techniques for visualizing comparison

The most important part about our tool is the comparison of behavior. We need to find the appropriate techniques to visualize comparison in behavior. In this section we describe the different considerations for visualizing comparison. According to

"Visual Comparison for Information Visualization" [16], the visual designs for explicitly assisting with comparison fall into three categories: Juxtaposition, Superposition and Explicit encoding.



(a) Juxtaposition (b) Superposition (c) Explicit Encoding: Difference

FIGURE 2.4: Different types of visualizing comparison between datasets of sensor X and Y [15]

2.5.1 Juxtaposition

Juxtaposition (also called separation) designs place objects next to each other. Juxtaposition can help the user to see patterns between elements [16]. Figure 2.4 (a) shows an example of Juxtaposition. In this figure you see the values measured by sensor X and Y separately for the user to compare.

2.5.2 Superposition

A Superposition design overlays multiple objects which are presented at the same place and time [16]. Figure 2.4 (b) shows two lines on top of each other that are visually distinguished by color. This makes the difference between sensor X and sensor Y more clearly visible. Where the Juxtaposition design shows two visualizations for sensor X and Y, the Superposition design shows one visualization were both sensor X and sensor Y are represented.

2.5.3 Explicit Encoding

Explicit encoding designs compute a relationship (e.g. difference) between objects [16]. Figure 2.4 (c) shows an example of Explicit Encoding. The visualization shows the arithmetical difference between the values measured by sensor X and those of sensor Y.

Choosing the right visualization for comparison is dependent on how we describe behavior and how we will compare users and their behavior. We will answer these questions is the next chapter.

Chapter 3

Problem analysis

In this project, we aim to gain more knowledge of the behavior of GameBus users. By analyzing the event data of GameBus, patterns and anomalies in behavior can be found. It can be tricky to visualize behavioral change patterns because the term behavior can be described in many ways. In the previous chapter we explained what behavior is, how behavior is defined in scientific literature and how these papers measure behavior and behavioral change. In this chapter we describe what behavior is in our context and how we can measure behavior and behavioral change. The general problems we want to solve in this chapter are: *What is behavior in the context of GameBus? How can this behavior be measured? How can we measure behavioral change in the context of GameBus?*
The answers of these questions are necessary to come up with a solution to visualize behavioral change in the context of GameBus.

Behavior in the context of GameBus is linked to the GameBus dataset. We will therefore first explain the dataset structure of GameBus and how data is distributed.

3.1 GameBus dataset

The dataset for this project consists of real data of the GameBus application between the years 2013 and 2018 saved in a MySQL database. This database has around 50 tables. To illustrate how behavior appears in the context of GameBus we created a simple model, see Figure 3.1. How this model translates to the database can been seen Appendix B. At the top of the model is the Session entity. The Session entity is one of the largest tables in the current dataset with around two million records.

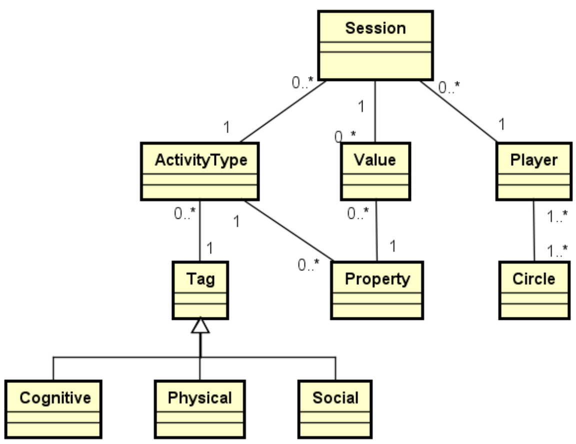


FIGURE 3.1: Simplified model of GameBus context.

Every activity that the player does via GameBus or an external application or device (e.g. FitBit) is stored as a session. Some examples of sessions can be seen in Table 3.1.

id	**created_at**	**dataProvider**	**activityType**	**player**	**creator**
2242016	2017-03-03 14:30:35	Google Fit	360448	100008	100019
2242017	2017-03-03 14:30:35	Google Fit	360448	100019	100019

TABLE 3.1: Example sessions

In this table you see two sessions of players that took place on 2017-03-03. The *player* attribute in a session is linked to the player that did the activity. A session can be registered by a different player which can be seen by the *creator* attribute. In Table 3.1 you see two sessions for the same activity. Both sessions are registered by player 100019, but one is done by user 100008 and the other by 100019. The dataProvider "Google Fit" tells us from which application the data is originated. In this example, the activity has been registered with the help of Google Fit.

A session is also linked to a type of an activity. This is done by the *activityType* attribute. In this case, the id "360448" is linked to "Walking". There are a total of 45 different types of activities in the current dataset with every app like Google Fit having its own types of activities. Every activity type is linked to one or more tags which declares if the activity is "social", "cognitive" or "physical".

As shown in Figure 3.1, every type of activity exists of properties to provide more information to a session. Examples of these properties are the number of steps, number of puzzles solved and GPS location. One session can have multiple values for different properties. In Table 3.2 you see the values provided for the two sessions that were given in Table 3.1. For session "2242016" there is a value "5000" registered for property "218". In Table 3.3 we see that property "218" is linked to the number of steps. This shows us that player "100008" and player "100019" have walked 5,000 steps. Also the property "220" is logged with a value of "3600". This tells us that the activity took one hour (3600 seconds).

id	**value**	**session**	**property**
24511057	5000	2242016	218
24511058	3600	2242016	220
24511059	5000	2242017	218
24511060	3600	2242017	220

TABLE 3.2: Example values for sessions

id	**property_name**
217	location
218	steps
219	when
220	duration (seconds)

TABLE 3.3: Example properties

3.2 Data analysis

With the tool Tableau [17] you can get fast insight in your data by creating visualizations. Tableau was used to see how the data is distributed per player, per year and per type of activity. In Figure 3.2 and Figure 3.3 you see the number of sessions for the first 30 players ordered by the total number of sessions. It can be seen that there are two players who have done much more sessions then the rest of the GameBus users. When excluding these players we get Figure 3.3, where data is more evenly distributed. The two extremely active players have around 2 million sessions and this leaves around 150,000 records for the rest of the players.

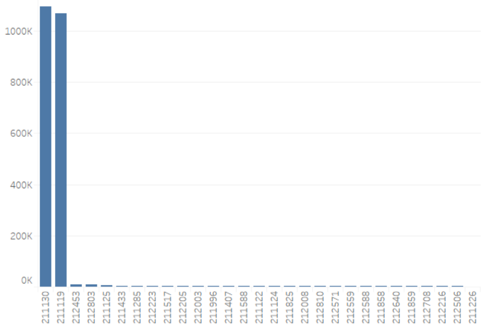


FIGURE 3.2: Number of sessions per player ordered in ascending order - all players included

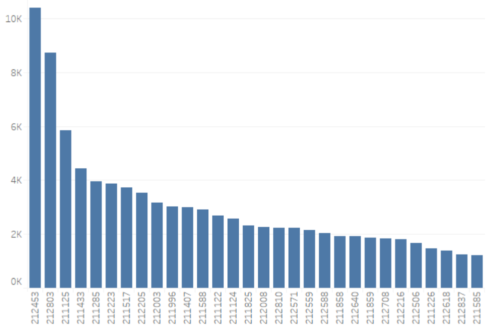


FIGURE 3.3: Number of sessions per players ordered in ascending order - outliers excluded

Most data of the dataset is gathered in 2016 because of the two extremely active players. When these players are excluded, you can see a growing number of activities over the years with the most data that is gathered in 2017.

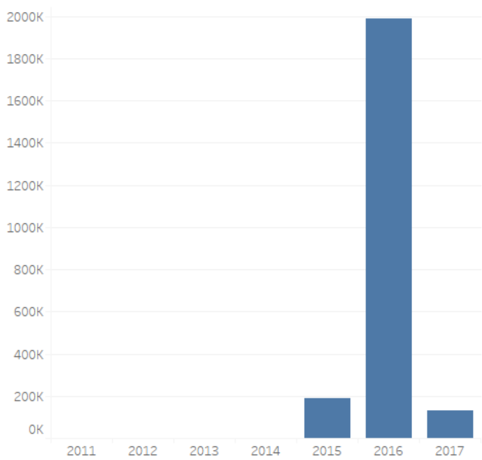


FIGURE 3.4: Number of sessions per year - all players included

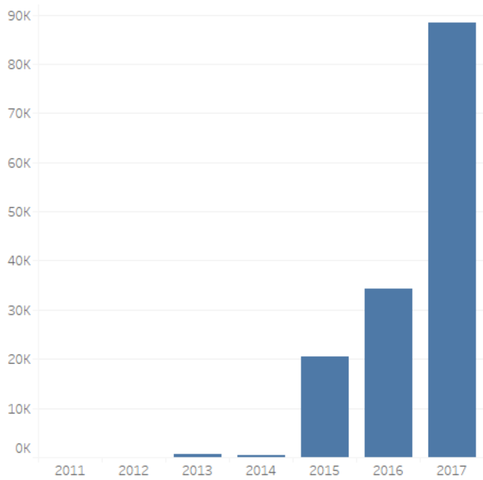


FIGURE 3.5: Number of sessions per year - outliers excluded

There are a total of 646 challenges within the current dataset. Not every session is linked to a challenge. Of the 150,000 sessions (with the two active players excluded), there are only 40,000 sessions linked to challenges. To give an example of how sessions are distributed within a challenge we look at challenge 730 from 2017.

This challenge took place from 2017-09-30 till 2017-11-01 and has a total of 1,359 sessions linked to it. Examples of activities that were awarded with points in this challenge are:
- Picture of doing sports with colleagues
- Picture of healthy meal
- Picture of exercise session (minimal 30 minutes)
- Healthy activity without picture
- 4000 steps

These activities belong to different types of activities. Figure 3.6 (a) shows the different types of activities with the registered number of sessions in this challenge.

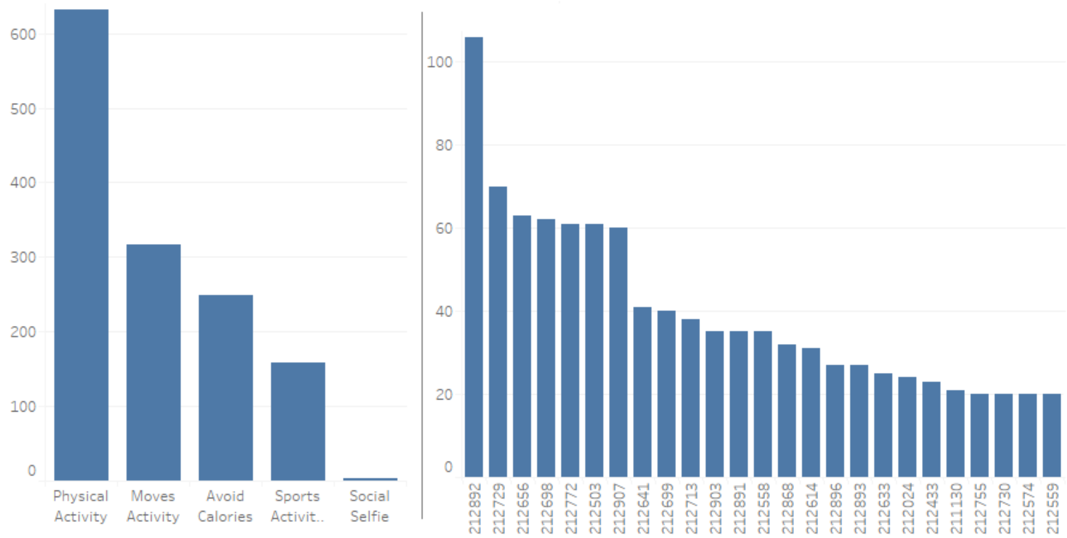


FIGURE 3.6: From left to right: (a) Types of activities in challenge 730 ordered by the number of sessions linked to that activity type. (b) Number of sessions for first 25 players in challenge 730 ordered by the number of sessions in ascending order.

In this challenge, 106 players participated. In Figure 3.6 (b) you see the first 25 players ordered by the number of registered sessions. A total of 151 circles participated in this challenge. You may notice that the number of circles is larger than the number of players. That is because a player can be in multiple circles. Of these 151 circles, 79 circles have 10 or less sessions. In Figure 3.7 you see the first 25 circles that participated in the challenge ordered by the number of sessions. Next to that, in Figure 3.8, you see the same 25 circles with their number of players. It can clearly be seen that the circle with the most sessions also has the most players.

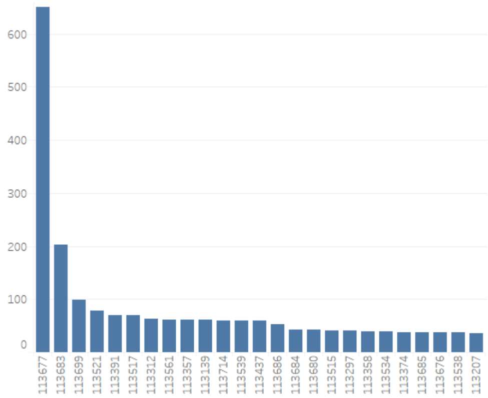


FIGURE 3.7: Number of sessions for first 25 circles in challenge 730 ordered in ascending order.

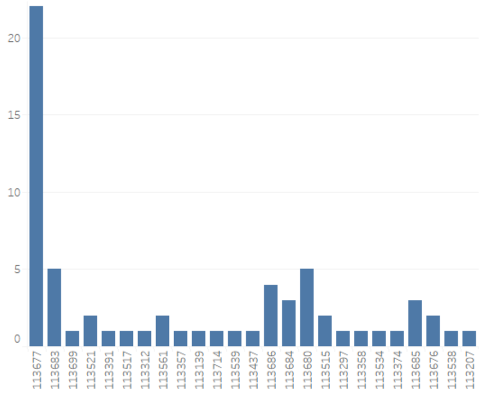


FIGURE 3.8: Number of players per circle in challenge 730. The order of the circles is the same as in Figure 3.7.

In conclusion, it can already be seen that there are some anomalies in the data. Two players have an extreme number of sessions which we could filter out because they dominate the results of circles. However, because they have so much data they could be great participants for behavior analysis. The amount of usage of the GameBus app differs a lot for every player and circle. To counter the dominating results of some players, our tool must be able to exclude players if necessary. The last conclusion that we can make from this analysis is that not every session is linked to a challenge. This means that not every activity is worth points. Challenges have influence on the behavior of players by motivating them into doing more activities. "GameBus Insight" should be able to show if this is really the case.

3.3 Behavior and measurements

Because behavior is a broad concept, there are many ways to measure behavior. We use the elements of behavior (capability, opportunity and motivation) as discussed in the chapter Related work as a starting point for measuring behavior. Measuring behavior is dependent on what can be done with the data. Looking at the GameBus application and the according data schema, it can be seen that the behavior of a user is reflected by his/her activities; in particular, the physical and/or social activities.

To say more about the behavior, we need measurements. The data for these measurements can be gathered from the property and value entities. Every type of activity has its own types of values (e.g. steps, duration, number of puzzles solved) and have therefore all their own kind of behavior. In this dataset, there are a total of 128 unique properties. The ten properties that occur the most can be seen in Table 3.4. For all the properties and their rate of occurrence in the dataset, we refer to Appendix C. The two extremely active users from the data analysis would dominate the result and are therefore excluded from the sessions to see what property is used most by the average user. The frequency of a property is influenced by the number of activity types that can use that property. The column "Activity types" states how many types of activities can use this property name and the column "Influencing activity types" states how many types of activities actually have data with this property in the database.

#	Property name	Frequency	Activity types	Influencing activity types
1	total time (seconds)	94544	11	6
2	when	72456	7	3
3	location	50873	6	5
4	distance (meters)	38462	4	3
5	calories burned	28421	5	4
6	activity type	25285	6	2
7	score	19732	7	7
8	start time	17800	2	1
9	steps	17720	6	2
10	tracking source	17193	1	1

TABLE 3.4: Property distribution of activities ordered by the frequency a property is used

For GameBus, we define behavior with the help of these properties. Looking at the measurements of behavior that other papers use in Chapter 2 Related work, we can give a definition for behavioral change. We do this by dividing change into two groups:
- Change in count of a measurement of behavior over time.
- Change in value of a measurement of behavior over time.
With the help of the different properties of GameBus and this definition, we create two functions to measure behavior for a user u.

First, we give some definitions. S is the set of all sessions within GameBus. For the sake of simplicity, a sessions $s \in S$ consists of a user u and a list of measurements M that are linked to that session. A measurement $m \in M$ exist of a key-value pair. The key is a string that exists of a GameBus property (e.g. "steps") and the value is a number (e.g. "1000"). P is the set of all possible properties (e.g. "steps") within GameBus.

The first function for measuring behavior is called b_{count} (behavior count). This function returns the total occurrence of a property for a person by giving a user u and a property p as parameters. In equation 3.1 you see the formula. It uses the *properties* function to return all the unique properties of the key-value pairs.

$$\begin{aligned} b_{count}(u,p) &= \#\{s \in S | s.u = u \wedge p \in properties(s.M)\} \\ properties(M) &= \{p | \exists_{v \in \mathbb{N}} : (p,v) \in M\} \end{aligned} \tag{3.1}$$

The second function is called b_{value} (behavior value). This function shows the total value of a property p for a user u. The *total* function returns the total value of all measurements in session s with property p.

$$\begin{aligned} b_{value}(u,p) &= \sum_{s \in S \wedge s.u = u} total(s,p) \\ total(s,p) &= \sum_{m \in s.M \wedge m.p = p} m.value \end{aligned} \tag{3.2}$$

As said above, there are a lot of different properties that can measure behavior. This amount keeps growing because the GameBus application is still in development. Therefore, we chose to make the number of measurements for behavior in our visualization tool limited but configurable. The user of the application can choose the properties that he would like to use to measure behavior.

The function b_{value} can only handle numerical measurements. This means that not all properties from above can be expressed with the help of the formula. An example of a property that cannot be expressed with the help of the formula is the type of the activity. You can still use the function b_{count} to look at the occurrence of a property but b_{value} cannot be used. To make sure that we cover all the elements of behavior (capability, opportunity and motivation), we have to make specific visualization to get more insight in the behavior. The types of these specific visualizations are again dependent on the data but should also be easy to expand. Examples for these visualizations are social interaction within a group of users and a map with all the locations of activities.

3.4 Behavioral change

To see change in behavior, the measured behavior has to be combined with time. We follow Fogg's behavior grid [14] that states that there are 15 ways of how behavior can change over time. The person that will be working with the visualization tool should be able to see these kinds of change in behavior with the help of "GameBus Insight". To compare behavior over time, measurements of behavior will be shown over different periods of time. As mentioned in the related work, changing your behavior can take between 18 to 254 days [13]. The time periods will therefore be constructed by weeks or months. Comparing behavior of days is not interesting because there is not enough data. Although years could also be an option, GameBus challenges typically take only a couple of months.

To conclude behavioral change, looking at different time periods is not enough. We should also compare different groups within the same time period. The tool "GameBus Insight" is then able to compare a control group with a treatment group and compare their difference in behavior within a challenge.

We extended the previous functions from equations 3.1 and 3.2 by adding a group G of users and a time period T. A time period T consists of a start and end date. Furthermore, we add a date $createdAt$ to all the sessions. A session $s \in S$ now consists of a user u, a list of measurements M and a date when the session has happened $createdAt$.

The formula bg_{count} (behavior group count) in equation 3.3 returns the total occurrence of a property p for a group of users within a certain time period.

$$\begin{aligned} bg_{count}(G,T,p) &= \sum_{u \in G} b_{count}(u,p,T) \\ b_{count}(u,p,T) &= \#\{s \in S | s.u = u \wedge p \in properties(s.M) \wedge time(s,T)\} \\ time(s,T) &= s.createdAt \in T \end{aligned} \tag{3.3}$$

The formula for the change in value was also extended (see equation 3.4). The formula bg_{value} returns the total value of a property p for a group of users within a certain time period.

$$\begin{aligned} bg_{value}(G,T,p) &= \sum_{u \in G} b_{value}(u,p,T) \\ b_{value}(u,p,T) &= \sum_{s \in S \wedge s.u = u \wedge time(s,T)} total(s,p) \\ total(s,p) &= \sum_{m \in s.M \wedge m.p = p} m.value \end{aligned} \tag{3.4}$$

The combination of these formulas shows the behavior for a group of users in a certain time period. To make the change in behavior more visible, we allow properties to be labelled (see Chapter 2). This was also done by other papers as mentioned in the literature review. By applying labeling, a property (e.g. "steps") can be divided into multiple labels (e.g. "0-100", "100-500", "500 or more"). This will not only show the frequency of how many times someone has walked but also tells you something about the length of the walk. This can make change in behavior more specific but also more easily visible.

3.5 Extra data

To give more insight into the behavior of the users of GameBus we are going to add the metabolic value and social interactions to the database.

3.5.1 Metabolic value

In Chapter 2 we mentioned that the MET-value (Metabolic Equivalent of Task) was used by a lot of papers for measuring physical activity. The MET-value is an objective measure for recording the intensity of a physical activity. One MET is the standard for the amount of energy expended at rest for one hour. Every type of activity has its own MET-value. These values are approximations because they can differ per individual. MET-values are an extension of the duration and frequency of physical activities. If a person is doing less physical activities for even less duration it does not mean that the person is less physically active. The person could have expended more energy because of doing physical activities that have a higher MET-value. The metabolic value is not yet in the database, but we would like to add it to gain a different perspective in the visualization tool. The implementation of how we add this data can be found in Chapter 5 Implementation.

3.5.2 Social interactions

Within GameBus users can do activities together. This is even encouraged. Some type of activities allow to give up a second user to indicate with who you did the activity with. Unfortunately, this property is not used by a lot of people. Therefore, we will use a different technique for creating this data. The implementation of this technique and the creation of this data can be found in Chapter 5 Implementation.

3.6 Requirements & Tasks

Now that there is a better understanding of the problem we can define the scope further with the help of requirements and tasks. The requirements will be defined first with some explanation and after that the tasks will be created with the requirements linked accordingly.

R1: GameBus Insight should work in a browser.

The visualization tool has to work in a browser and should therefore use JavaScript. By doing this, we give the opportunity for GameBus developers to add GameBus Insight to their already existing applications.

R2: GameBus Insight can create groups of users based on circles or challenges.

The tool will mainly be used to compare circles in different challenges. Therefore, the easiest way to create groups is based on circles or challenges. These groups can be named by the user of the tool.

R3: GameBus Insight can show an overview of the behavior for multiple groups of users over time.

As discussed above, there are a lot of different properties that can measure behavior. Therefore, we chose to make the number of measurements for behavior in our visualization tool limited but configurable. The user of the application can choose the properties that he would like to use to measure behavior. This will keep the application flexible for when the GameBus application and its data has updated. Measurements of behavior can also be labelled into multiple categories by the user to make change in behavior more detailed and also easily visible.

R4: GameBus Insight can show behavior of a group in a certain time period.

To show the difference in behavior between time periods we want to show the behavior of a single time period. The time periods will be constructed by weeks or months.

R5: GameBus Insight can be used to compare the behavior of one group in two or more time periods.

To detect change in behavior, we will look at the behavior of two time periods and compare them to see the difference.

R6: GameBus Insight can be used to compare the difference in behavior between two or more groups in one time period.

As mentioned above, only looking at the difference in behavior between two time periods is not enough because the GameBus application has a learning curve. Therefore, detecting change in behavior will be done by comparing groups and time periods.

R7: GameBus Insight can show the contribution of a user with respect to his/her group.

As seen in the Figure 3.2, a circle of GameBus users differ a lot in the number of activities that they do. To make a conclusion about a group and their behavior, we must also look at the contribution of the users individually.

R8: GameBus Insight can exclude selected users.

Because not all users do the same number of activities it can be necessary to exclude users to get a better insight in the behavior of the rest of the group.

R9: GameBus Insight should be able to filter data.

Being able to filter the data will help to get more insight in the behavior. In the data analysis we have already seen some anomalies in the data. It can be helpful to remove these anomalies which can be easily spotted with the visualization tool. GameBus Insight should have three types of filters to get more insight into the behavior:

1. **Sessions**. Filtering on sessions makes it possible to remove anomalies from the data.
2. **Weekdays**. Filtering on weekdays will make it possible to examine the data from certain days. For example, to examine the data that is gathered during the weekends.
3. **Hours**. Filtering on hours will make it possible to examine the data from certain hours. For example, to examine the data that is gathered after 17:00.

R10: GameBus Insight can import and export created groups.

Because the tool works in a browser, it would be helpful to save the groups that you have created when you quit and import the groups to go further where you left off.

R11: GameBus Insight can import and export a created view

With the applied filters and the chosen time periods you get a certain view. To again make the work of the data analyst easier we propose a function for importing and exporting a view (filters and time periods).

R12: GameBus Insight can show weekly or daily patterns for a measurement of behavior.

To spot behavior of a group of people, it can be interesting to look at the selected time period and see if there is a weekly or daily behavior. An example for this could be that a group of people always goes running on Wednesday at 18:00.

R13: GameBus Insight can show specific visualizations of behavior and these can be extended.

As mentioned before, not all data from the property and value entities can be used in the constructed formula for behavior. These visualizations will mostly be used for why behavior has changed. Specific visualization that we will make during this project are a map with the activities of the users per group and a graph that represents social activities between users in a group. These specific visualizations should be easily expandable for future use.

When working with "GameBus Insight" you come across different tasks to be able to visualize the behavior of GameBus users. We linked the requirements to these different tasks accordingly.

Task	**Task name**	**Requirements**
T1	Select users to form a group.	R2, R10
T2	Create a time period within the data of the group.	R3, R4
T3	Compare behavior with another group between time periods.	R5, R6, R7
T4	Remove outlier from a group.	R8
T5	Compare behavior of weekly (or daily) patterns between groups.	R12
T6	Exclude activities (e.g. because you suspect cheating).	R9
T7	Compare behavior with specialized visualizations to investigate the change in behavior.	R13
T8	Export view and groups.	R10, R11

Chapter 4

Design

This chapter describes the design of the visualization tool "GameBus Insight". In this chapter we will describe which design choices were made and what visualizations were chosen to compare behavior of users within GameBus. The information seeking mantra of Shneiderman [26] was used for designing the application:

> *"Overview first, zoom and filter then details on demand."*

4.1 Dashboard

To give insight into the behavior of GameBus users we have designed a dashboard. Looking at the tasks and requirements we divided the application into 5 components (see Figure 4.1):

1. **Users**: Select existing users to create a group.
2. **Group creator**: Show the created groups.
3. **Timeline**: Show the behavior of the created groups to select interesting time periods.
4. **Focuser**: Zoom in on the selected time periods to show the change in behavior per group.
5. **View**: Store settings to reproduce visualizations.

These components are closely linked together. To immediately see results when creating a group or selecting a time period, we want to show all these components on one page.

FIGURE 4.1: Overview of the application GameBus Insight existing of 5 components: (1) Users, (2) Group creator, (3) Timeline, (4) Focuser and (5) View.

4.2 Users

To compare groups of users and their behavior, we first need to create groups. The goal of the user component is to find the right users to form a group. The following requirements were taken into account for designing the user component:

1. Create groups based on circles or challenges (**R2**).

2. Show how many users are in a circle or challenge (additional requirement)

3. Show which users are in a circle or challenge (Additional requirement).

4. Create groups based on individual users (Additional requirement).

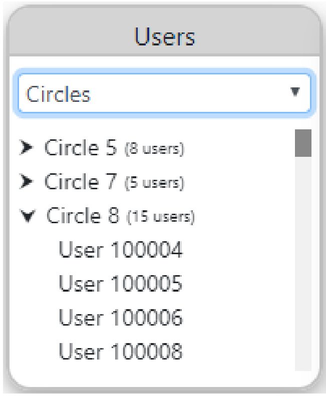


FIGURE 4.2: User component that shows a list of circles which can be used to create groups. Circle 8 is expanded to show which users are in that circle.

Figure 4.2 shows the created user component. We show a list for circles and for challenges. These are shown separately to reduce the size of the component. You can swap between circles and challenges with the help of a drop-down menu. Circles and challenges are sorted on their id to make them easier to find. After every circle and challenge you can read, in a smaller font, the size of the circle or challenge. Individual users can be found by expanding a circle or challenge. Circles, challenges and individual users can be used in the group creator component to create a group.

4.3 Group creator

The goal of the Group creator component is to create groups. Groups can be created by selecting users, circles and challenges from the user component. The output of the group creator component is a list of named groups, where a group exists of a list of users. These groups are used in the next visualizations. The requirements that were taken into account for designing the group creator component are:

1. Create groups (**R2**).

2. Delete groups (Additional requirement).

3. Assign name of group (Additional requirement).

4. Show which users are in a group (Additional requirement).

5. Enable and disable groups (Additional requirement).

6. Enable/disable individual users from group (Additional requirement).

7. Assign color to group (Additional requirement).

8. Import/export groups (**R10**).

There are a lot of additional requirements for this component. To remember which users are put together into a group, the name of the group should be assignable and it should also be possible to see which users are put together. To distinguish groups after creating them, every group should have an assigned color which can be used throughout the application.

The produced Group creator component can be seen in Figure 4.3. We chose to create new groups by dragging users, circles and challenges from the user component to the "Add" field. This is an intuitive way of creating groups. Instead of creating a new group, you can also drop the users, circles and challenges on an already existing group. This makes it possible to add multiple circles or challenges in one group. Groups automatically get a name assigned when created. The name is "group" plus the number of groups that already exist (see Figure 4.3). This is not necessarily unique but it works as a placeholder. Names can be edited by pressing the pencil next to the name. The users can be seen by expanding a group. Every user can be set on inactive individually by checking or unchecking a checkbox (Requirement **R8**). The whole group can be set inactive with the checkbox next to it. This will remove the group from the visualizations without deleting the group from the group creator.

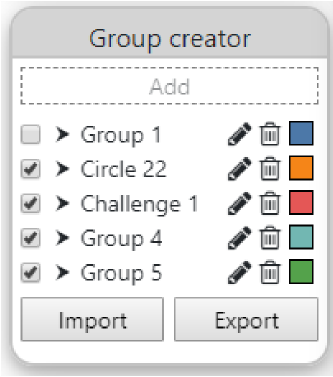


FIGURE 4.3: Group creator component with 5 groups created and group 1 disabled.

The number of groups that the Group creator component can show at once is dependent on the resolution of your screen. During development, we could see 5 groups at once if the groups were not expanded. Therefore, we limited the maximum number of groups that could be created to 5. If a group is expanded you need to scroll within the component to see all the groups. This choice was made to keep the size of the component at a minimum. Groups are distinguished by color throughout the whole application. The color of a group is automatically assigned. The color palette has been derived from the tool Tableau [17]. This color palette exists of 10 colors (see Figure 4.4) of which we only use the first 5.

FIGURE 4.4: Color palette of Tableau.

4.4 Timeline

The purpose of the timeline component is to show the user an overview of the behavior of the created groups. The user can then select interesting time periods to compare within the Focuser component (Task **T2**). As described in Chapter 2 and Chapter 3, behavior of groups is measured by comparing multiple measurements of behavior. The timeline component has the following requirements:

1. Show multiple measurements of behavior for a group of users over time (**R3**).
2. Create time periods by weeks or months (**R4**).
3. Compare measurements of behavior with each other (Additional requirement).

Figure 4.5 shows the created Timeline component. We have chosen Linegraphs as visualization for showing an overview of the behavior. A Linegraph is created for every measurement of behavior. By doing this we can compare the created groups per measurement (Superposition) within a large timeframe. This makes it easy for the user to spot interesting time periods. The different measurements of behavior are aligned vertically (Juxtaposition) for better comparison. Comparison with Juxtaposition can be hard because the user has to shift their attention between measurements to see patterns. To help the user with this comparison we added a vertical line in all the Linegraphs. This vertical line can be moved with the mouse and is on the same x-position on every graph.

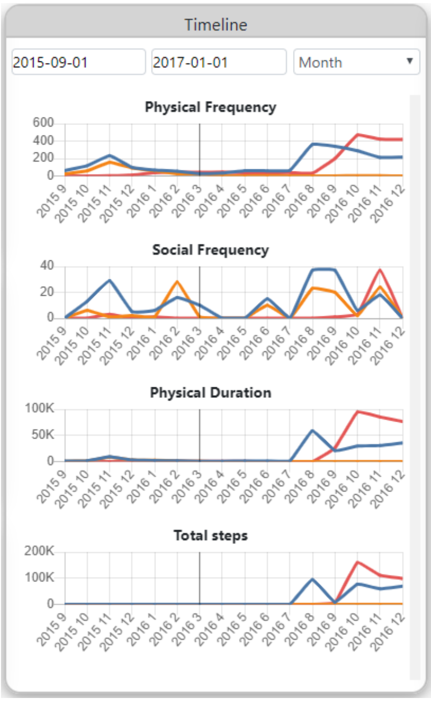


FIGURE 4.5: Timeline component that shows results for 3 groups (red, blue, orange) with the help of 4 measurements of behavior. These measurements are: "Physical frequency", "Social Frequency", "Physical duration" and "Total steps".

We have chosen to visualize 4 measurements of behavior which can be seen in Figure 4.5. In Chapter 2 and Chapter 3 we discovered that behavior can be measured in many ways. The measurements of this visualization tool are configurable to keep the tool flexible and future proof. The data analyst can choose the measurements of behavior for comparison. How these measurements can be configured is discussed in Chapter 5 Implementation.

Change in the measurements of behavior can be shown over months or weeks (see Figure 4.6). This can be changed with the help of the drop-down menu at the top of the component. The reason why measurements are shown over months and weeks is because time periods can be selected by weeks or months (Requirement **R4**). Time periods can be created by click-and-drag within a graph. If data is shown per month then time periods are also selected by months. If data is shown per week then time periods are selected by weeks. The maximum number of time periods that can be selected is three. It is not possible to select one time period by month and another by week. With three time periods, you can compare 5 groups over 3

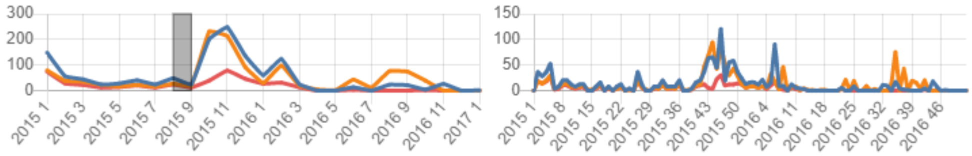


FIGURE 4.6: Linegraphs of a measurement of behavior for 3 groups (red, blue and orange) between the years 2015 and 2017. The left figure shows the data aggregated by month and the right figure shows the same data aggregated by week. The left figure also has a selected time period for the month September in 2015. This is indicated with the grey square.

different time periods. Selected time periods are indicated with a grey square within the Linegraphs, see Figure 4.6).

The Timeline component shows data for 1 year by default. 1 year was picked as a compromise between performance and giving a good overview. This date-range can be edited at the top of the Timeline component. When looking at the data per week for two years, see Figure 4.6, you see that the Linegraphs of some groups almost become unreadable and selecting one week as time period is almost impossible because it is so small. A zoom function was made to tackle this problem. When zooming in on a Linegraph, the other Linegraphs will also zoom in.

4.5 Focuser

Following the information seeking mantra of Shneiderman [26], we want to zoom in on the selected time periods. The Focuser is created with the goal of giving more insight and trying to understand the behavior of the created groups within the selected time periods. Understanding behavior can be done with respect to three key elements: capability, opportunity and motivation. As described in Chapter 2 Related work. We will give insight into the capability element with the help of our four measurements of behavior. The opportunity and motivation element will be shown with specialized visualizations (Task **T7**). Hence, we divide the Focuser into two parts:

1. Give more insight into measurements of behavior for the chosen time periods and groups.
2. Give insight into why behavior may have changed for the chosen time periods and groups with specialized visualizations (Task **T7**).

Figure 4.7 shows an overview of the Focuser component. Each part has its own page. We chose to visualize them both on different pages to make sure that you do not have to scroll while examining the visualizations. Furthermore, with pagination, we allow the tool to be extended further.

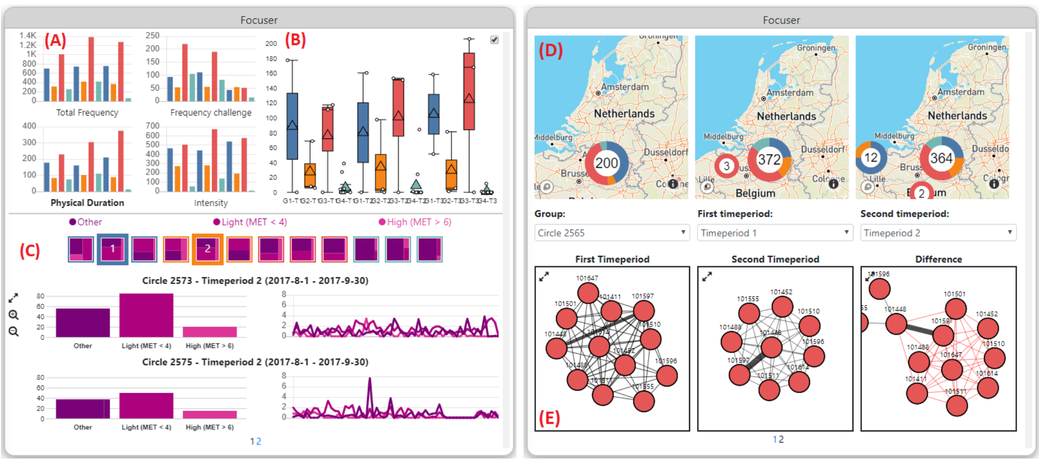


FIGURE 4.7: The two pages of the Focuser component. The colors are consistent with the Timeline component to indicate groups. The left figure shows the first page of the Focuser and it exists of: (A) Measurement Barcharts, (B) Boxplot, and (C) Labels of a measurement. The right figure shows the second page of the Focuser and it exists of: (D) Geographic location, and (E) Social circle.

4.5.1 Measurements of behavior

In this section we take a better look at the first part of the Focuser. The goal of this part is to show general statistics of the measurements of behavior to conclude change in behavior. This part was designed based on the following requirements:

1. Show behavior of a group in a certain time period (**R4**).
2. Compare the behavior of a group in two or more time periods (**R5**).
3. Compare the difference in behavior between two or more groups (**R6**).
4. Show the contribution of a user with respect to his/her group (**R7**).
5. Exclude selected users (**R8**).
6. Filter data (**R9**).
7. Show weekly or daily patterns for a measurement of behavior (**R12**).
8. Visualizations should be able to scale with number of groups and time periods (Additional requirement).

To meet all the above requirements, we need to make more than one visualization. These visualizations need to be able to scale with the number of time periods and groups. We have chosen to give insight into the measurements of behavior with three different parts, see Figure 4.7 (A), (B) and (C).

In the first part we want a simple visualization to show the results of the chosen groups and time periods. See Figure 4.7 (A). We will discuss this part in the section **Measurement Barcharts**. Second, we would like to see how individual users performed per measurement of behavior. See Figure 4.7 (B). We will discuss this part in the section **Boxplot**. Lastly, we want to give more insight per measurement of behavior. See Figure 4.7 (C). To give more insight into a measurement of behavior we have the requirement for a measurement to be labeled (Requirement **R3**). This part will be discussed in the section **Labels of a measurement**.

Measurement Barcharts

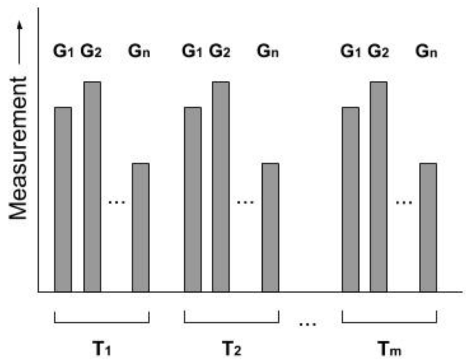


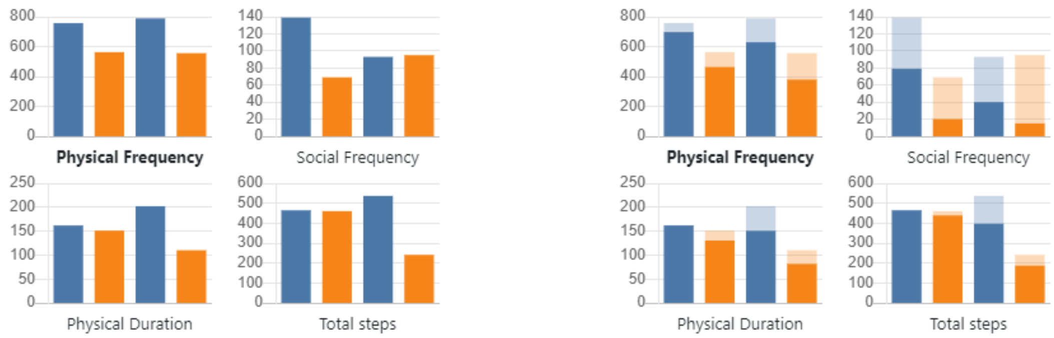
FIGURE 4.8: Conceptual design of a Barchart for a certain measurement. The Barchart can scale with the n number of groups and m number of time periods.

When the data analyst has selected multiple time periods in the Timeline component, you want to see the difference for a measurement of behavior per time period. At the same time you want to compare the difference with the different groups that are selected (Task **T3**). This will result in a value per group per time period for the 4 measurements of behavior. We have chosen a collection of Barcharts (Juxtaposition) to show the results per measurement. With a collection of Barcharts we can visualize the change per group and per time period for every measurement (see Figure 4.7 (A)).

We show a Barchart for every measurement. A Barchart can scale with the number of groups and time periods, see Figure 4.8. For a certain measurement, we show the results per group for every time period. This makes it possible to immediately compare the behavior between groups but also to examine the increase and decrease between groups.

Figure 4.9 shows how the Barcharts are visualized within "GameBus Insight". We use the same colors throughout the whole application to indicate different groups. In this example you see two groups for two different time periods. The four Barcharts (one Barchart for every measurement) are shown in a square to make the most out of the available space. The rest of the visualizations within this part of the Focuser component, the Boxplot (see Figure 4.7 (B)) and "Labels of a measurement" (see Figure 4.7 (C)), are linked to the Barchart. The Boxplot and "Labels of a measurement" show more insight in a selected measurement. A measurement can be selected by clicking on the title below the Barchart. In Figure 4.9 you see that the measurement "Physical Frequency" is selected which makes it bold.

In the Boxplot (see Figure4.7 (B)) and "Labels of a measurement" (see Figure 4.7 (C)), users can be selected. The contribution of the selected users with respect to the total value is also visualized in the Barcharts (see Figure 4.9 (B)). The color of the total value gets a lower opacity which makes it possible to spot the contribution of the selected users.



(A) (B)

FIGURE 4.9: Barcharts of all 4 measurements of 2 groups and two time periods. Groups are indicated by their color. The measurement "Physical frequency" is selected (bold). This selected measurement will be used in the other visualizations. Figure (B) shows how the Barcharts are visualized when users are selected. The color of the total value gets a lower opacity such that you can see the contribution of the selected users. Users can be selected in the Boxplot (see Figure 4.7 (B)) and "Labels of a measurement" (see Figure 4.7 (C)).

Boxplot

The goal of this project is to compare the behavior between groups of people. It often occurs that some people from the group are more active than other users. It is therefore necessary to show the contribution of a user with respect to his/her group (requirement **R7**). For example, you would like to get insight in how the average user performed, if there are users that did nothing at all or if all users are growing in a certain measurement of behavior. A visualization technique that can scale with the number of groups and time periods, and can answer all these questions is the Boxplot. It can be used to spot outliers for a group of users.

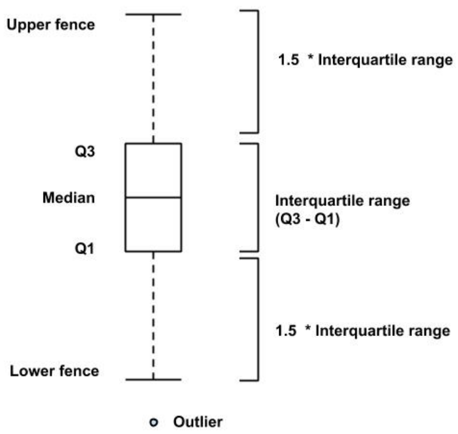


FIGURE 4.10: Standard configuration of Tukey Boxplot

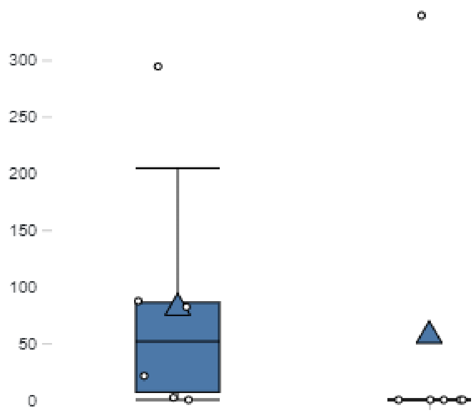


FIGURE 4.11: Example of two Boxplots in GameBus Insight. Each Boxplot corresponds with a time period and contains data of the blue group. The group exists of six users who are shown as small circles.

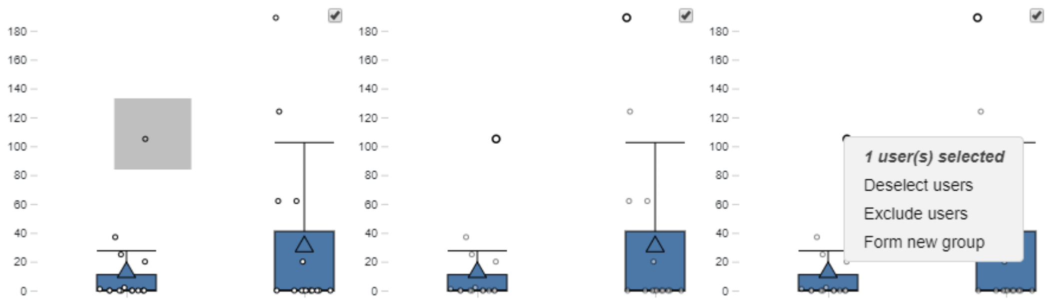
A standard Boxplot displays a batch of data by visualizing five values from the dataset: the extremes, the median, first quartile (the median of the data points to the left of the median) and third quartile (the median of the data points to the right of the median). There exist multiple variations of the Boxplot. These alternative implementations arise from different choices in computing quartiles and the fences that determine whether an outlier should be plotted individually or within the Boxplot. We choose a type of Boxplot which is often called the Tukey Boxplot [27]. This type makes it possible to spot outliers from a group.

Figure 4.10 shows the basic configuration of a Tukey Boxplot. The most important element about this type of Boxplot are the upper and lower fence. In other versions of Boxplots they often use these as the minimum and maximum of the datapoints. This is undesired if you want to spot outliers and that is why we choose this type of Boxplot. The upper fence is defined as:

$$min(1.5 * Interquartile\ range, max(datapoints))$$

The lower fence is defined in the same way with the min and max operators swapped. Datapoints that fall outside the boundaries of the upper and lower fence are called outliers. Outliers are usually drawn as individual points.

An example of a Boxplot within "GameBus Insight" can be seen in Figure 4.11. A Boxplot shows information of the selected measurement from the measurement Barcharts (see Figure 4.9). We have added three extra features to the Tukey Boxplot from Figure 4.10. First, we show all the users as individual points instead of only showing the outliers because we want to be able to select them. The location of these individual points is random for the x-axis, but it is in between the total width of the Boxplot (see Figure 4.11). By doing this, we scatter the users such that they can be selected easier. If a group exists of a lot of people it could become hard to read the Boxplot. The users can be hidden/shown with the help of a checkbox (see Figure 4.12 in the top right corner). The second feature that we added is a triangle. This triangle represents the average value of all the users, which can be interesting if you want to know how the average user performed. The third feature that we added was the color. It represents the group as mentioned in section 4.3. The Boxplots of Figure 4.11 show the frequency of physical activities of users from the blue group within two chosen time periods. In the first time period you have one outlier and the rest of the users are almost all within Q1 and Q3. In the second time period only one user has done physical activities. When hovering above the individual points the id of the user appears and you can conclude that the two outliers in the Boxplot are actually the same user.



(A) (B) (C)

FIGURE 4.12: Three figures that show the selection of users within a Boxplot. First you select the users by click-and-drag. The selected user(s) will be highlighted. When right-clicking on a user a menu will appear.

By showing the users individually, we can also show the change in behavior of an individual user. This can be done by selecting users. Users can be selected by clicking the individual points or by click-and-drag with the mouse (see Figure 4.12 (A)). Figure 4.12 (B) shows how the selected user(s) will be highlighted. It makes the selected users bigger and reduces the opacity of the not selected users. Selecting users makes it easy to show the change of a subset of the users over the different time periods. It also shows if a user is present in a different group. Selecting users also updates the other visualizations within the Focuser component as you will see in the other sections. Selecting users can also be used to exclude users from a group or to create a new group (Task **T4**). This can be done by right-clicking a selected user (see Figure 4.12 (C)).

Labels of a measurement

To give more insight into a measurement of behavior we have the requirement for a measurement to be labeled (Requirement **R3**). Measurements will be labeled into multiple categories by the data analyst to make change in behavior more specific and more easily visible. This labeling can be shown with different types of visualizations. A measurement could be shown as a Piechart for every time period with every label as a slice. Another way would be to show it as a stacked Barchart or as a normal Barchart with every label as a different color. However, this is not enough. To show change for a group we must show again the behavior of the individual users to be able to conclude something. This will result in hierarchical data. An example of such data can be seen in Listing 4.1. This example describes a measurement called "Total Frequency". It shows per user the number of sessions per GameBus category (cognitive, physical and social activities).

```
"Total Frequency": {
    "Cognitive": [
        {User: 100019, Frequency: 10},
        {User: 100020, Frequency: 1}
    ],
    "Physical": [
        {User: 100019, Frequency: 12},
    ],
    "Social": [
        {User: 100008, Frequency: 3},
        {User: 100020, Frequency: 2}
    ],
}
```

LISTING 4.1: Example of hierarchy of measurements with labels.

The depth level of the hierarchical data always stays the same. It goes from measurement of behavior to multiple labels to individual users. To show the change per label we go with a Barchart. However, we combine the Barchart with a Treemap to show the contribution for every user. Hence, we choose a Treemap-barchart [28] for this visualization. However, the Treemap-barchart is not enough for visualizing the labels. We know that the behavior is done in the selected time period but we can still not say anything about how data is divided over the time period. Looking at Fogg's behavior grid [14] from Chapter 2, we must show the duration of the behavior to know if it was "Dot", "Span" or "Path" behavior. To show this, we combine a Treemap-barchart with a Linegraph (see Figure 4.7 (C)). The Linegraph shows for every label the change per day over the selected time period. If all the data is gathered in one day it can be seen in this Linegraph.

We cannot simultaneously show a Treemap-barchart and Linegraph for every group and every time period because this will take up too much space. Therefore, we have chosen to only compare one time period of a group with another time period

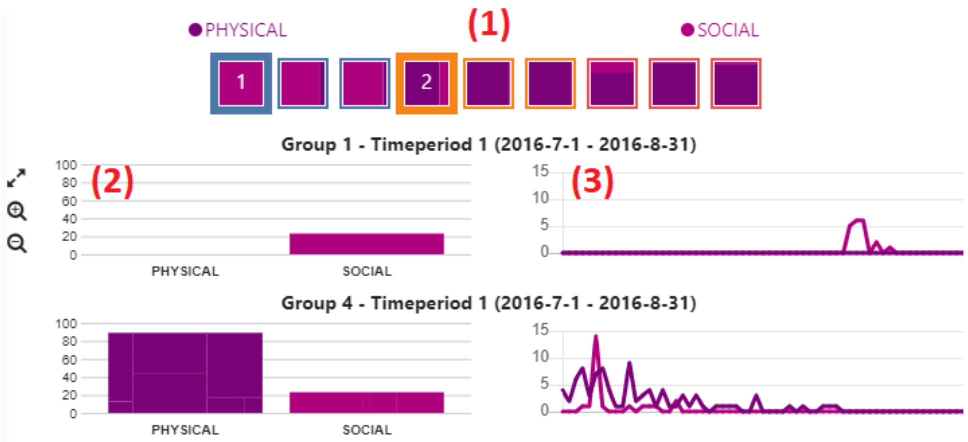


FIGURE 4.13: Visualization of labels of a measurement. It exists of 3 parts: (1) Small Treemaps to show per group and time period the distribution between labels. (2) Treemap-barchart to compare the labels in more detail. (3) Linegraph that shows how data of the labels is divided over the time period. In this figure we compare the total frequency of activities between group 1 and group 2 in time period 1. The labels for this measurement are "Physical" and "Social" activities.

of a group. This makes it possible to compare different time periods within the same group or with different groups. Figure 4.13 shows how we visualized this. Three different visualizations can be spotted: (1) small Treemaps, (2) Treemap-barcharts, and (3) Linegraphs.

We have chosen to compare a time period of a group with another time period of a group. We must first show an overview of the behavior of all the groups and time periods to show the data analyst which time periods could be interesting. This is what we will do with the small Treemaps. We will show a small Treemap for every time period and every group, see Figure 4.14. These Treemaps can be selected to give more detail into the behavior of that group and time period.

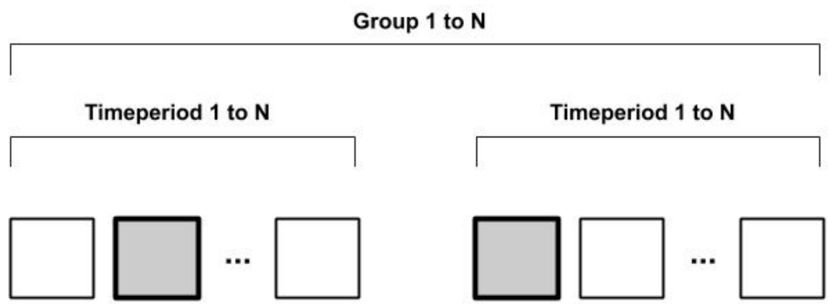


FIGURE 4.14: Concept of small Treemaps. Every square is a Treemap that shows the distribution between the labels of a measurement. It scales with the number of time periods and the number of groups. The number of Treemaps is equal to the number of groups multiplied by the number of time periods. Two Treemaps are selected which is indicated by the gray color.

The implemented small Treemaps can be seen in more detail in Figure 4.15. The Treemaps show the distribution between the labels of a measurement. The labels within the Treemap are indicated by color as informed by the legend. The color purple was chosen because this color is not used by any of the groups. In Figure 4.15, you see a total

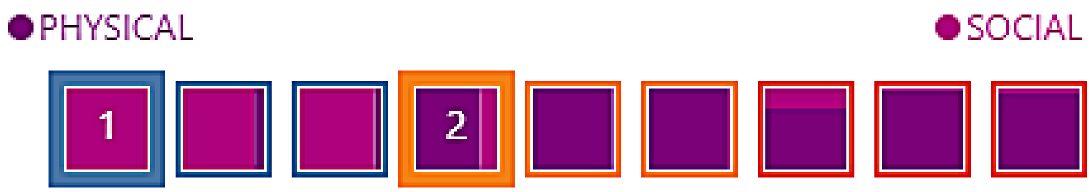


FIGURE 4.15: Zoom in of small Treemaps for three groups (blue, orange, red) and three time periods. Treemaps show the distribution between different labels. In this case the distribution between physical and social activities. Time period 1 of group 1 and time period 1 of group 2 are selected. The numbers show in which order the Treemaps are selected.

of 9 Treemaps. Treemaps are made for every group and every time period. In this case there are three time periods selected for three created groups. Groups are indicated by the color of the outline of the Treemaps. In Figure 4.15, time period 1 of group 1 and time period 1 of group 2 are selected. This is visualized by increasing the border width and showing the selection number within the Treemap. The selection number is the order in which the Treemap-barcharts and Linegraphs are visualized.

We show a Treemap-barchart and a Linegraph of the two selected Treemaps, see Figure 4.13 (2) and (3). The Treemap-barchart shows a Treemap for every label. A Treemap is represented as a bar which makes it possible to also see the total value per label. The Treemaps of every label exists of all the users within the group to show the contribution per individual user. The Linegraph shows for every label the change per day over the selected time period. If all the data is gathered in one day it can be seen in this Linegraph. The Linegraph is also a start for analyzing weekly patterns (Task **T5**).

Users can also be selected in the Treemap. We use squarified Treemaps to make the selection of users easier. Squarified Treemaps make the squares within the Treemap as big as possible. Figure 4.16 shows one selected user in the Treemap. It shows the contribution for the selected user per label (Requirement **R7**). Selected user(s) can be excluded or put into a new group in the same way as in the Boxplot.

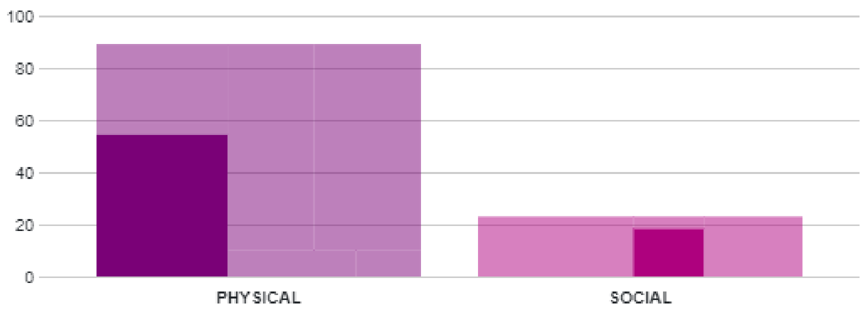


FIGURE 4.16: Treemap with one user selected. It shows the contribution of that user for physical and social activities.

The Treemap-barchart also provides a way to filter further (Requirement **R9**). Sessions of a label can be removed by pressing on the label below the Treemap-barchart. This could help you to remove dirty or dominant data and explore other patterns in the GameBus dataset.

The Linegraph shows weekly or daily patterns but it is insufficient. By pressing on the magnifying glass next to the Treemap-barchart, you change the appearance of the Treemap-barchart (see Figure 4.13). It can then show how data is gathered on a weekly (Figure 4.17 (a)) or daily basis (Figure 4.17 (b)) (Requirement **R12**). These Treemap-barcharts show on which days and hours the group are most and least active. Days and hours can be filtered out by clicking on them (Requirement **R9**). Friday and the hour "19:00" are filtered out which can be seen by the red color.

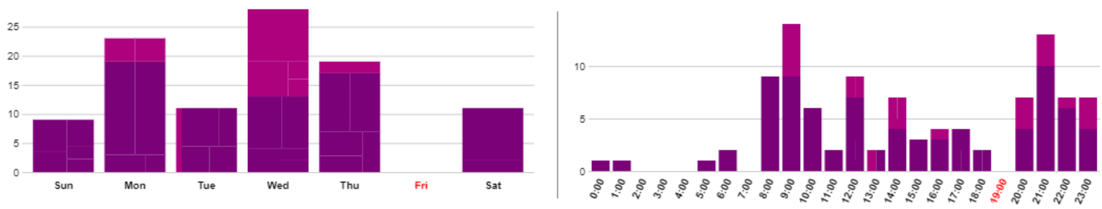


FIGURE 4.17: Other appearances of the Treemap-barchart to examine daily or weekly patterns per label. From left to right: (a) Treemap-barchart per weekday to see on which days people are most active. Friday is disabled which can be seen by the red color. (b) Treemap-barchart per hour to see on which period per day people are most active. "19:00" is disabled which can be seen by the red color.

4.5.2 Specialized visualizations

In this section we describe the second part of the Focuser component. The goal of this part is to show specialized visualizations of behavior (Task **T7**). These visualizations should give more information about the opportunity and motivation element of behavior. This can give us more insight in why behavior has changed. There was only one requirement for this section:

1. Show Specialized visualizations of behavior which can be extended (**R13**).

We build two specific visualizations. The first one is for the opportunity element which is influenced by the location of the users. The second one is for the motivation element. In particular, we will show the Relatedness factor of motivation from the self-determination theory by visualizing social interactions.

Geographic Location

As said in Related work, opportunity is a key element for influencing behavior. Location has a lot of influence on the opportunity element. If you live near a gym you might be more interested in exercising. Showing change in location could be a change for the opportunity element and an explanation why behavior has changed. Some activities of GameBus log the users location. To show the locations of the users we use a map. The idea of this visualization is to show the amount of activity of a group spread around the world. We choose to show a map per time period (Juxtaposition). On each map we show all the groups which are divided by color (Superposition). Only showing colored dots on a map is not enough. Most campaigns of GameBus run in one location and this will result in a lot of overlapping points. This problem is tackled by clustering the points. Each cluster shows how many times it is visited by the users.

We had the following requirements for showing the clusters on a map:

1. Zoom in and out functionality on the map (Additional requirement).
2. Clusters should separate when zooming in and merge when zooming out (Additional requirement).
3. Clusters should show the contribution of each group (Additional requirement).

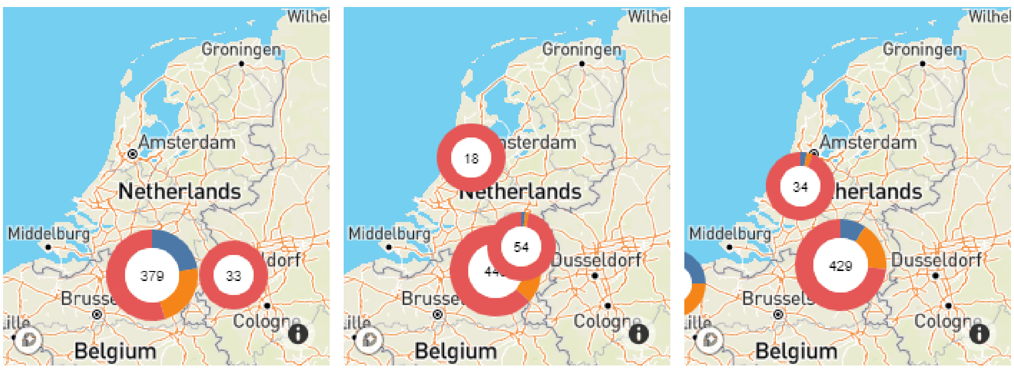


FIGURE 4.18: Implemented visualization to show the location of users from GameBus. In this example, we show the location of three groups (blue, orange, red) within the Netherlands for three different time periods.

To show the maps and clusters we use the JavaScript library MapBox [29]. We also use MapBox to cluster the data points. It shows clusters according to the zoom level of the map. Clusters are made with the algorithm Hierarchical greedy clustering.

In addition to the clusters, a donut chart was added to show the contribution per group (see Figure 4.18). We have chosen to make the radius of the donut chart dependent on the number of visits. This will make it easier to see where people have done the most activities. We created three thresholds for the radius of the donut chart. These thresholds are chosen such that different types of clusters can easily be distinguished. The thresholds can be seen in Table 4.1.

Threshold	**Radius**
Number of visits > 100	20 pixels
Number of visits > 15	18 pixels
Number of visits > 0	16 pixels

TABLE 4.1: Thresholds used for calculating the radius of the donut chart.

By default, the maps are initialized on the Netherlands, but you can move around to look at the rest of the world.

Social interactions

Another key element for influencing behavior is motivation. Behavior of a group could be changed because users are motivating each other (Relatedness). The data of social interactions was created with the help of the sessions. The creation of this data can be read in the chapter Implementation. An example of how the created data looks can be read in Table 4.2.

Id	**Player**	**Player2**	**Created_at**	**Session**	**Session2**
1	100019	100008	2017-05-06 13:50:00	220123	220124

TABLE 4.2: Example of created data for social interaction. It shows that users 100019 and 100008 have done an activity together on 2017-05-06

We would like to visualize this data to show how much users interact with each other. We have chosen a graph to visualize the social interaction (see Figure 4.19). Figure 4.19 shows how graphs are presented within the application. Every node in the graph represents a user. Only the users that have done some interactions with other members are included in the graph. The color of the nodes is representative of the group. The thickness of the edge is dependent on the number of sessions that were done between two users.

As the data analyst you can select a group, and two time periods that you want to compare with the help of drop-down menus. You can only compare time periods of the same group because they have the same users. In the visualization tool we show 3 graphs. The first two graphs are for the first and second selected time period

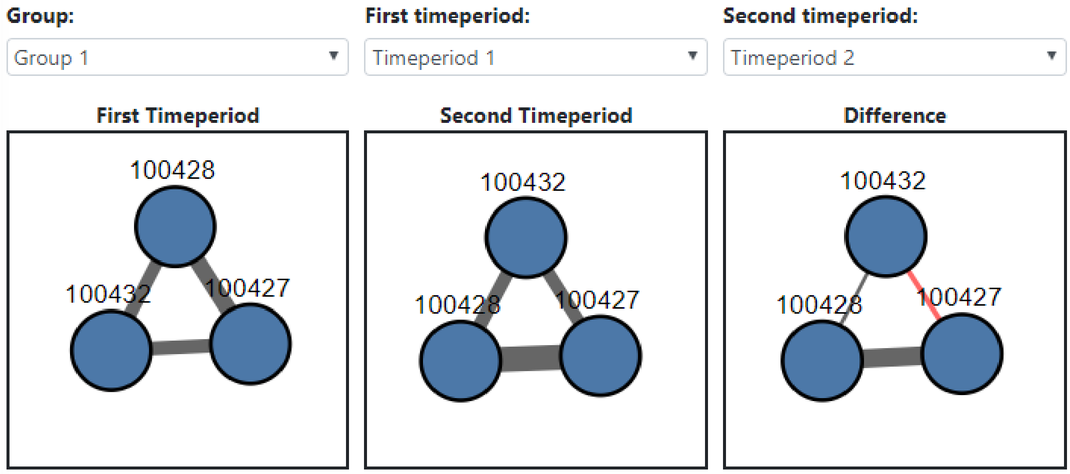


FIGURE 4.19: From left to right: (a) Graph of the social interactions between players of group 1 in time period 1. (b) Graph of social interactions between players of group 1 in time period 2. (c) Graph that shows the difference between (a) and (b)

which represent the interaction in that time period. The third graph represents the difference between the number of interactions in time period one and time period two (Explicit encoding).

Comparing graphs is hard even when the number of nodes are few. Therefore, we added the third graph to make change more easily visible. The red edges show a decrease in interaction and the black edges an increase. For large groups the box around the graph quickly becomes too small to show all the users. A zoom and panning function was added to tackle this problem.

4.6 View

The Focuser component makes it possible to filter certain data. In particular, this is done by the Treemap-barchart in the Focuser Component. Data can be filtered by day, by hour or by session. The goal of the view component is to show what you have filtered out. By storing these settings we can save an interesting view of the data which you can import to look at it again.

The View component has no influence on the created groups or the chosen measurements. If you want to recreate the exact same visualizations you need the same groups, same measurements, same time periods and the same filters. The view component holds the chosen time periods and the chosen filters. The advantage of this is that you can also look at a view with a different set of groups or with a different set of measurements. For this component we came up with the following requirements:

1. Should show dates of selected time periods (Additional requirement).
2. Should show which data is filtered out (Additional requirement).
3. Should be able to reset filters (Additional requirement).
4. Import/export views (**R11**).

The result of the view component can be seen in Figure 4.20. This component shows the selected time periods and the filtered out days, hours and sessions. The garbage can resets a property within the view. A view can be imported and exported in the same way as groups (Requirement **R11**).

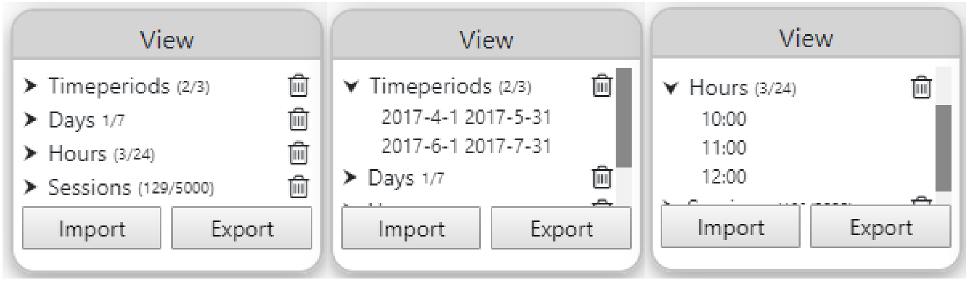


FIGURE 4.20: Three screenshots of the view component. (a) All properties of the view component. (b) Property time periods expanded which shows the start date and end date of two time periods. (c) Hours expanded which shows that the hours "10:00", "11:00" and "12:00" are filtered out.

Chapter 5

Implementation

This chapter describes the implementation phase of the project. During this stage the stated techniques and methods from the design chapter were tested and implemented. Plugins and extension that were used during this stage were: ReactJS, Chart.js, MapBox, d3 and Cube.js. The installation guide of the application can be found in Appendix E.

5.1 Application structure

ReactJS was used as framework to create the client-side of the application and to satisfy requirement **R1**. ReactJS is a component-based framework. In Figure 5.1 you see the structure of the application with the help of a component diagram. It can be seen that every component in the dashboard (users, group creator, view, timeline and focuser from Figure 4.1) has its own ReactJS component. The important part about the component diagram is the stack order. This identifies the update loop. The structure

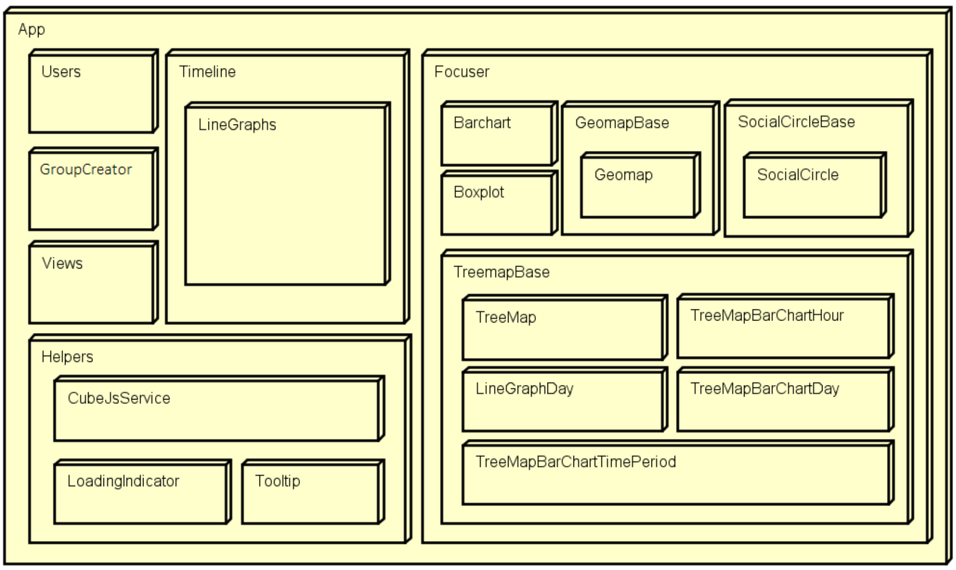


FIGURE 5.1: Component diagram of client-side of visualization tool

of the application can be seen as a tree with the app component as the root. The app component triggers the design components (Users, GroupCreator, Views, Timeline and Focuser) when they need to update. The design components then update their children and so on. In the sequence diagram in Figure 5.2 you see an example of an update. A new group has been made by dragging some players into the Group creator component. The Group creator component tells the App component that

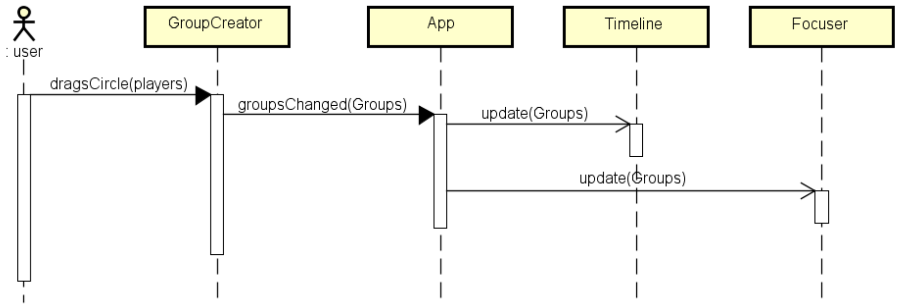


FIGURE 5.2: Sequence diagram of what happens when a new group has been added

the groups have changed. The app component will communicate with the Timeline and Focuser asynchronously to update. Both the Timeline and Focuser use the CubeJsService component to gather data from the server. The Timeline and Focuser will then restructure the data and send the updated data to their components (e.g. Barchart, LineGraphs) such that the visualizations can update.

5.2 Measurements of behavior

Working in the browser comes with challenges (Requirement **R1**). In Chapter 3, we concluded that we should visualize behavior with multiple measurements. These measurements should be configurable to make the tool future proof. In conclusion, the application should be responsive while working with big data and yet flexible for different measurements of behavior. A standard server for retrieving the data from the database is not enough, because it is not fast enough. We need to aggregate the data and also use caching mechanisms to increase the speed. We use the open source

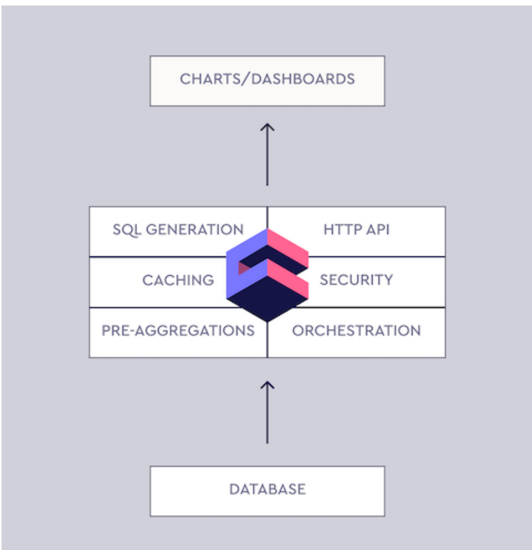


FIGURE 5.3: Cube.js architecture [30]

```
cube("CubeName", {
  sql: "SELECT
        FROM
        WHERE",

  measures: {
    value: {
      type: "sum",
      sql: "value"
    }
  },

  dimensions: {
    player: {
       type: "string",
       sql: "player"
    }
  }
});
```

LISTING 5.1: Structure of a standard cube in Cube.js framework Cube.js [30] to deal with this problem. Cube.js serves as an analytics back-end and communicates with the client (visualization tool) and the database, see Figure 5.3. As a client you can create connections with a cube. A cube represent a table of data and is used to retrieve data. The cube retrieves the data from cache or from the database. The cube will aggregate the data according to his configuration. In the next section we describe the structure of a cube.

5.2.1 Cubes

The visualization tool makes connections with a cube to retrieve the data. A standard cube defines itself by 4 properties: a name, an SQL statement, measures and dimensions which can be seen in Listing 5.1. The SQL statement tells the cube which data it should access from the database. The measures parameter consist of a list of measures. A measure is a column from the SQL statement on which you want to aggregate. In the example we aggregate on the column "value" and sum all the values. The type of the aggregation could be changed to count. The aggregation will work as we have described in the formulas 3.3 and 3.4 in Chapter 3. The dimensions parameters consist of a list of dimensions. A dimension controls how the data is grouped while aggregating. In the example we group by player which means that we sum the value from measures per player. For every measurement of behavior we use a cube (see Figure 5.4). These cubes can be altered to change the measurements but there are some requirements to make the visualization tool work. An example for a required dimension is "Label". The "Label" dimension is the labeling of the measurement as described in Chapter 2 and 3 to get more insight in the behavior. These labels could be an existing column but it can also be an extra column that you

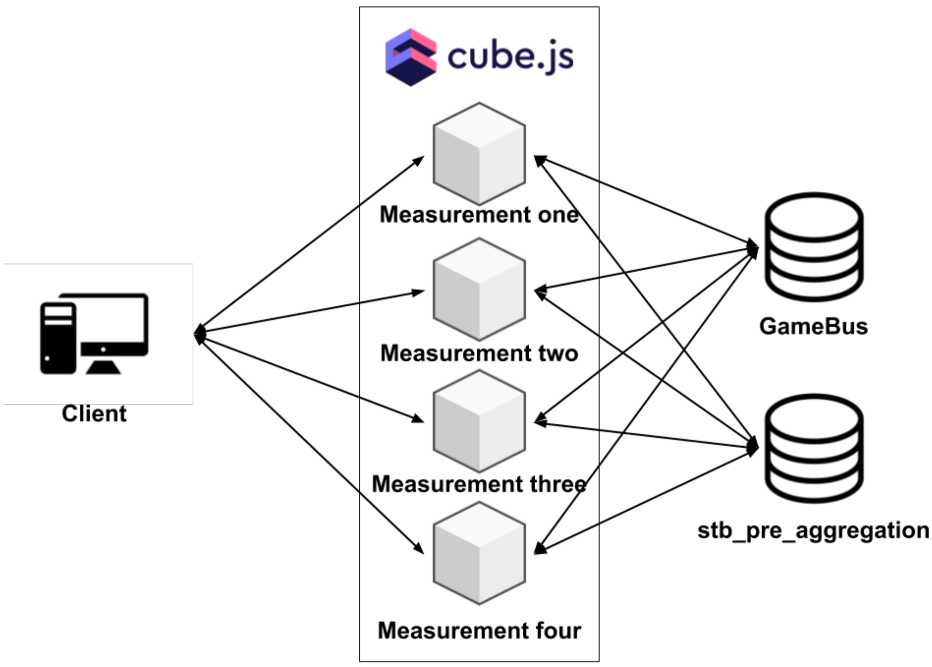


FIGURE 5.4: Usage of Cube.js within GameBus Insight. Every measurement is a cube which retrieves the data from the database or from cache and sends it back to the client. To increase performance, every measurement is pre-aggregated by Cube.js. These pre-aggregations are saved in the database "stb_pre_aggregations".

created in the SQL statement. Further requirements for creating or altering a cube can be found in Appendix E.

5.2.2 Aggregation

To increase the performance of the application the data is aggregated. This is done by Cube.js when sending a request to a cube but also beforehand. The most important component that needs aggregated data is the timeline component because it should show an overview. Every measurement in the timeline component is aggregated beforehand with the help of an extra property in the cube: "pre-aggregation". Cube.js creates two tables per measurement with the aggregated data. One table aggregates the data per user per month and the other table aggregates per user per week. These tables are saved in the database "stb_pre_aggregations" which can be seen in Figure 5.4.

5.3 Database

5.3.1 Indexes

Cube.js queries the database to retrieve data. Cube.js can then aggregate the data to the desired form. The performance of the application relies a lot on the speed of the database query. Therefore, the structure of the query is important but also indexes on the database can help a lot. Two indexes were created during this project to increase the performance which can be found in Appendix D. The cubes (queries) are configurable to change the measurements of behavior. For future use, it is important to look at which indexes the database is using when running a query if the visualization tool is lacking performance.

5.3.2 Dirty data

From the data analysis in chapter 2, it was already clear that there were some anomalies in the data. While implementing and testing the visualization tool we found some more. The "where"-statement of the cube and the filter of the visualization tool can be used to filter some of these out. If values become too large, cube.js can no longer aggregate the data and it crashes. This happened only once when creating a cube for the number of steps. In Table 5.1 you see 5 sessions with their linked number of steps. As clarification, these steps are registered for a single session. If we take the first session as example with 1,099,862,293,713,632 steps, this would result in 838,095,067,810 kilometers that a person would have walked in one session. By further analysing these sessions it became clear that these sessions were done by only two users. These users were filtered out for the number of steps with the help of the "where"-statement in the cube.

A similar issue happened for the total duration of activities. For implementing and testing purposes, we created a cube with a measurement that shows the total duration of physical activities. We then noticed that some activities were registered with a duration over 100 days. For this measurement, we decided to ignore all duration values that are over 1 day.

Session id	Number of steps
2305048	1099862293713632
2305047	868980254538734
2305049	789109741821258
2305013	751344163395424
2305056	692821413047191

TABLE 5.1: Example of sessions that have a large number of steps

Because data comes from different types of applications (e.g. FitBit, GoogleFit) not all data is saved in the same way. An example of this is the location which was sometimes saved comma-separated ("51.1797383,5.16184079") and sometimes semicolon-separated ("51.1797383;5.16184079"). Also a lot of NULL values were encountered. No data was removed from the database. The cubes were used to filter out or correct some of the ambiguous data records. The data analyst can decide for every measurement if he would like to ignore certain data or not.

5.3.3 Metabolic value

In Chapter 3 we mentioned that we would add the metabolic value to the database. An extra table was added in the database for the metabolic value. The value table of the GameBus database holds which type of physical activity a user has done. Unfortunately, ever app (like FitBit and GoogleFit) gives its own description of these types. There are around 300 types of physical activities because types like "Walking" are described in more than one way. Table 5.2 shows the first 15 types of physical activities ordered by their occurrence. We use the compendium of physical activities

Physical activity type	# of occurences
walking	1819685
Biking	189905
Running	90894
Rowing	11480
Andere Beweegsessie 30min	2205
Skating	1475
Other	1192
hiking	748
Bike	489
Sportuurtje	323
Cycling	188
Run	169
Deskbike	146
Sport	108
Wandelen	106

TABLE 5.2: First 15 types of physical activities ordered by their occurrence.

[31] to link the most occurring types of physical activities to their MET-value. Not every type of activity can be linked to a MET-value. Types like "Other" are to vague to be linked. The MET-values of the compendium [31] are very specific. An example

that they provide for walking is "hiking or walking at a normal pace through fields and hillsides, MET-value: 5.3". For the sake of simplicity, we leave out the speed (km/h) and the environment of the activity and link every physical activity to a standard MET-value which is also provided by the compendium [31]. These types of activities are stored in a new table in the GameBus database with their according MET-value. This table can be seen in Table 5.3. We will use the metabolic value as a measurement in the next chapter.

Physical activity type	**MET-value**
Bike	7.3
Biking	7.3
Cycling	7.3
Deskbike	7.3
Fietsen	7.3
Fitness	5.0
hiking	5.3
Lopen	3.5
Rowing	5.8
Run	9.0
Running	9.0
Stepcount	3.5
Swimming	6.0
Walk	3.5
walking	3.5
wandelen	3.5

TABLE 5.3: Table metabolic_value that is created for the GameBus database. It exists of the most occurring physical activities with their according metabolic value.

5.3.4 Social circle

Also social interactions between people were added to the database. We created a new table called SocialCircle in the GameBus database which is filled with all the social interactions between people. A social interaction is defined as two sessions that are done by two different people for the same type of activity on the same date and time. The time does not have to be exactly the same but within a range of one minute. It can occur that there was no social interaction but two sessions were logged on the same time for the same activity as a coincidence. This does not really matter for our tool. In our tool we show the number of social interactions between players. The chance of two players actually doing activities together becomes higher when the number of stored social interactions becomes higher. The algorithm for finding all the social interactions can be found in Appendix G.

Chapter 6

Results and evaluation

In this chapter we will discuss the results of the visualization tool. In particular, we will visualize the data of a campaign that took place in the city Mol (Belgium) [32]. This campaign took place in 2017 from April to October with every month a new challenge. We will perform three experiments on these challenges. First, we will analyze the behavior of private circles. Second, we will look at competing circles to study what we can conclude on the change in behavior. After that we will look at the buddy system of GameBus.

6.1 Measurements

The measurement of behavior are configurable. We have chosen four measurements that we will use for the results. Each measurement is labeled into multiple categories (Requirement **R3**).

1. Total Frequency:
This measurement shows the number of sessions done by the users. The total registered sessions can tell us something about the number of activities that are done. The total frequency measurement is divided into the labels: "Physical", "Social" and "Cognitive". This gives us more insight in which type of activity users do.

2. Frequency challenge:
This measurement will show us the total number of sessions that were registered for a challenge. In combination with the measurement "Total Frequency" we can see how many activities users do outside the challenges. Frequency challenge is labelled on the activity types that were registered for that session (e.g. "Google fit activity", "Avoid Calories").

3. Physical duration:
Of all the physical activities done by the users, this measurement shows the total duration in hours. All the activities with a duration longer than a day are filtered out. The total duration is further divided by labels into 4 categories: activities with a metabolic value lower than 4 (Light), activities with a metabolic value lower than 6 and higher than 4 (Medium), activities with a metabolic value higher than 6 (High) and activities with an undefined metabolic value (Other).

4. Intensity:
This measurement shows the total expenditure of the users expressed with the help of the metabolic value. The metabolic value was added to the data as described in chapter 5 Implementation. The total expenditure tells us about how active a group of users have been. It is calculated by multiplying the duration of an activity (in hours)

and the according metabolic value of the activity. This measurement is labelled on the different kind of physical activities (e.g. "walking", "cycling").

6.2 Results

6.2.1 Private circles

We will first investigate how the private circles performed within the challenges of "Mol". The private circles are all the users that participated in these challenges. We use the challenges of the User component to create one group with all the users that participated in the challenges of "Mol". There are a total of 149 users. Analyzing the individual users gives us information about how they performed during these challenges. In Figure 6.1 you see the results of the four measurements for all individual circles. There are some interesting points in the data. Looking at "Frequency challenge" it is clear that the challenges started in April and ended in October. Furthermore, you can see that the total frequency of activities increases between April and October. Yet, we can see that activities done for challenges is slightly decreasing after the first month until the last month of these challenges.

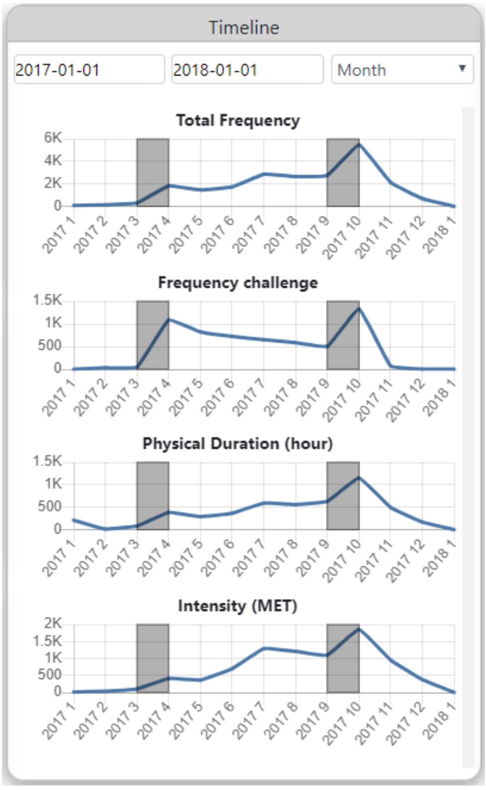


FIGURE 6.1: Timeline component of private circles between 2017 and 2018. The black boxes are the created time periods for the months April and October.

We choose to compare the months April and October which correspond to the first and last challenge of the campaign in "Mol". Figure 6.2 shows the difference between the two time periods for the four chosen measurements. We can see an increase for every measurement. Not every measurement has the same amount of increase. "Total Frequency" has an increase of circa 200% while "Frequency Challenge" only has an increase of circa 20%. This means that the users are doing more activities that are not linked to challenges.

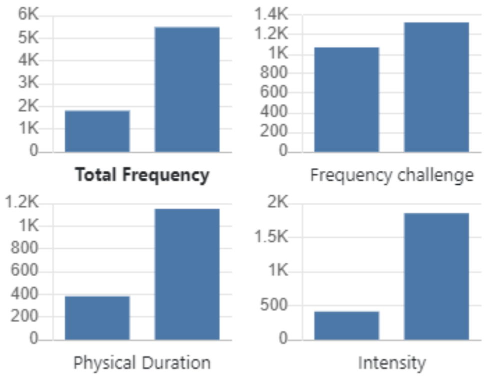


FIGURE 6.2: Barcharts of measurements of behavior. It shows the results of the private circles for the chosen time periods (April and October).

As said in the introduction of this chapter, the "Total Frequency" measurement is divided into the labels "Physical", "Social" and "Cognitive". The Treemap-barchart, in Figure 6.3, shows that most activities in the first time period (April) are physical. This could explain why the "physical duration" measurement and "total frequency" measurement show a similar Linegraph. Figure 6.4 and 6.5 shows the Boxplot of the total frequency measurement. These Boxplots show the results of the first selected time period (month April) and the second selected time period (month October). It can be seen that people are more active in the second time period by looking at the individual points. It can also be seen by the average which has grown. The average is visualized with the blue triangle. The Boxplot shows us that the biggest section of the group is not doing much activities.

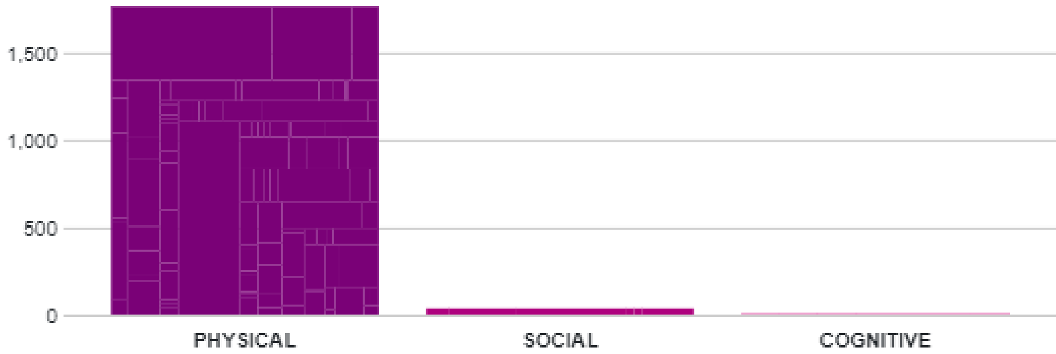


FIGURE 6.3: Treemap of the measurement total frequency of the first time period (April). Total frequency is labeled on the type of the activity: Physical, social or cognitive). It shows that almost all activities within April were physical.

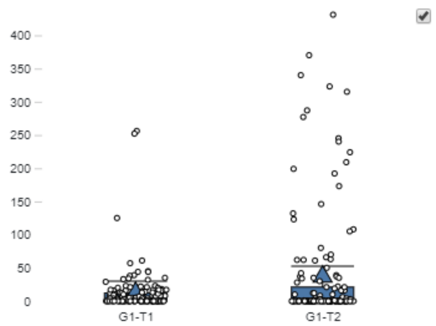


FIGURE 6.4: Boxplot of total frequency for the time periods April (left) and October (right). It shows how the individual users have changed between the two time periods.

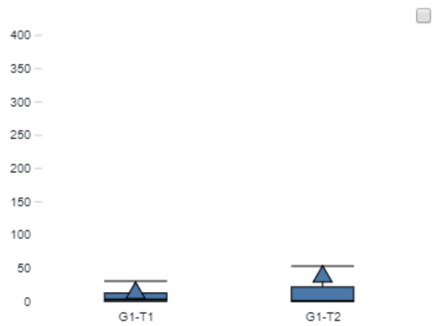


FIGURE 6.5: The same Boxplot as in Figure 6.4 but with the individual users disabled. This shows us that the average has grown over the two time periods.

6.2.2 Competing circles

Now that we have seen some results of how all the users performed we look at the behavior of the circles that competed against each other. In this section we will show the behavior of 3 circles. These circles were randomly chosen in the visualization tool to create one group per circle. We will examine these groups and see what we can conclude on their behavior. Figure 6.6 shows the timeline of the three groups which are indicated by color. Group 1 is blue, group 2 is yellow and group 3 is red. It can be seen that the measurements are having peaks in the same time periods. This could be the case because it is a start of a new challenge. The measurement "Frequency challenge" peaks at the beginning of the "Mol" challenges for all groups. Three time periods were chosen which can be seen in the timeline component. We have chosen the first month, the last month and a month in the middle of all the "Mol" challenges. In each of these months there is a different challenge. Figure 6.7 shows the Barchart and Boxplot of the selected time periods. The Boxplot is shown for the measurement "Total frequency" which can be seen by the bold label in the Barchart. The blue group consists of 7 users, the yellow group of 13 users and the red group of 11 users. The Boxplot shows that every group has one or two outliers in at least one time period. We have selected these users in Figure 6.7. If we exclude these users we get the Barchart and Boxplot from Figure 6.8. What is interesting to see in Figure 6.8 is that in the first time period almost every user is doing some activities but it becomes less

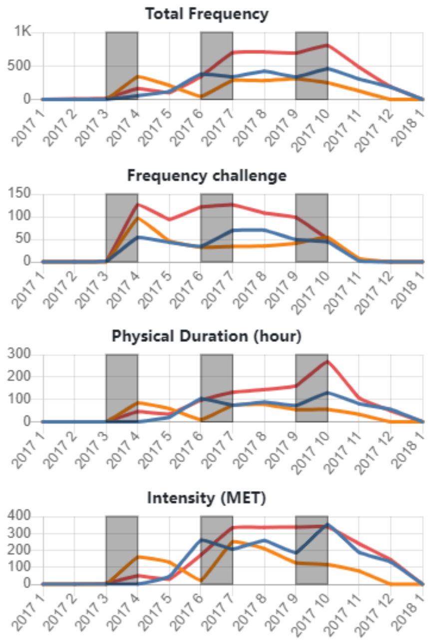


FIGURE 6.6: Timeline component of three competing circles (blue, orange and red) within the challenges of "Mol". Time periods are created for the months: April, July and October.

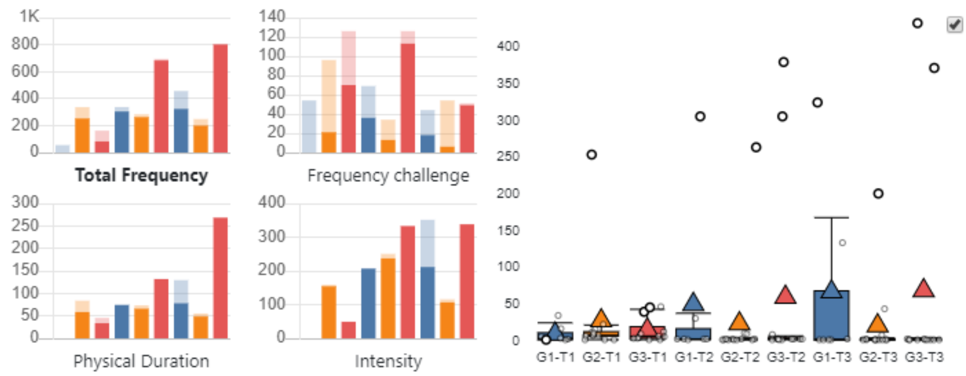


FIGURE 6.7: Barchart and Boxplot of the three created groups competing in the Mol challenges. Groups are indicated by color. The Barcharts show a bar for every group and every time period (3 groups * 3 time periods = 9 bars). The same holds for the Boxplot. The Boxplot is shown for the total frequency measurement which can be seen by the bold label in the Barcharts. In the Boxplot the outliers are selected. Users that are selected are bold and users that are not selected have a lower opacity. Contribution of selected users is shown within the Barcharts.

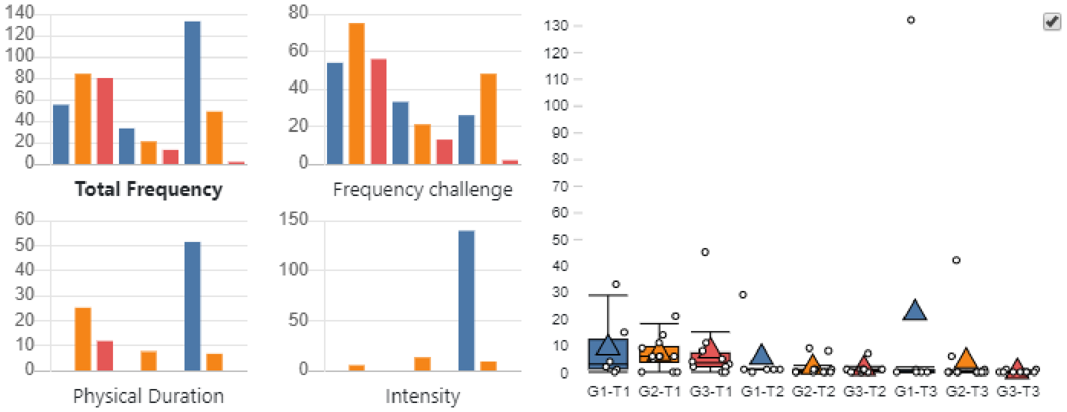


FIGURE 6.8: Barchart and Boxplot of the same time periods and groups as in Figure 6.7 but with the outliers from Figure 6.7 excluded. The boxplot is shown for the measurement "Total frequency".

in the next two time periods. This could be explained by the novelty effect. The novelty of the app is gone and people lose interest. Although most users become less active, when looking at Figure 6.7 the total frequency grows. The reason for that is because the outliers become more active. When the outliers are excluded, you see that the physical duration and intensity drops immensely. For certain time periods there is no data for these measurements. With the private circles we concluded that almost all activities were physical. So one could wonder why do we not have physical duration for the done activities? The reason for that is because not every physical activity has a registered duration. Physical activities that are missing their duration cannot be used for calculating the total duration or total intensity.

Figure 6.7 shows an increase in physical duration between time period two and three for the red group. Yet, the intensity is about the same. We will zoom in on the red group with the outliers included and compare the physical duration of these two time periods with the Treemap-barchart in figure 6.9. One of the outliers is selected to show the growth of this individual user. The Treemap-barchart shows per label of

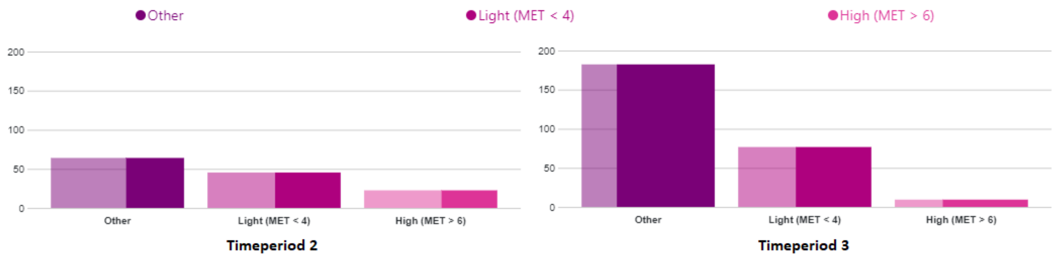


FIGURE 6.9: Treemap-barchart of the red group for the measurement physical duration. We show the Treemap-barchart for time period 2 (July) and time period 3 (October). One of the two outliers is selected to see his contribution.

a measurement the amount. The label with the highest amount of total duration in hours is "Other". This means that most physical activities that are done do not have an assigned metabolic value. The label "High (MET > 6)" is decreasing between the two time periods and the label "Light (MET < 4)" is increasing. This means that the users in the red group are doing more activities with a lower metabolic value, which is also the reason why the intensity level stays about the same. The total duration of physical activities is increasing but it is invested in activities that require less energy. Figure 6.10 shows the Treemap-barchart of the intensity measurement of the red

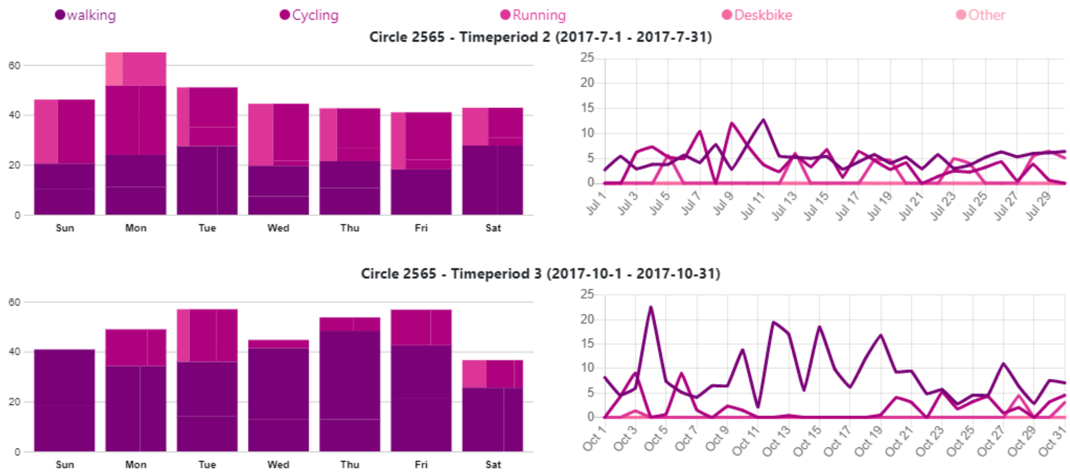


FIGURE 6.10: Intensity Treemap-barchart per day for the red group for time period 2 (July) and 3 (October). The intensity is calculated by multiplying the metabolic value with the number of hours an activity is done.

group for time period 2 and 3. It can indeed be seen that the users are walking more in time period 3 and do less cycling and running. It can also be seen that the users of the red group no longer cycle on Sundays. The Linegraphs show that the users have been active during the whole time period. Figure 6.11 shows the social graph of the red group of time period 2 and 3 and the difference between these two graphs. Looking at the Boxplot from Figure 6.7, we have selected the two outliers from the

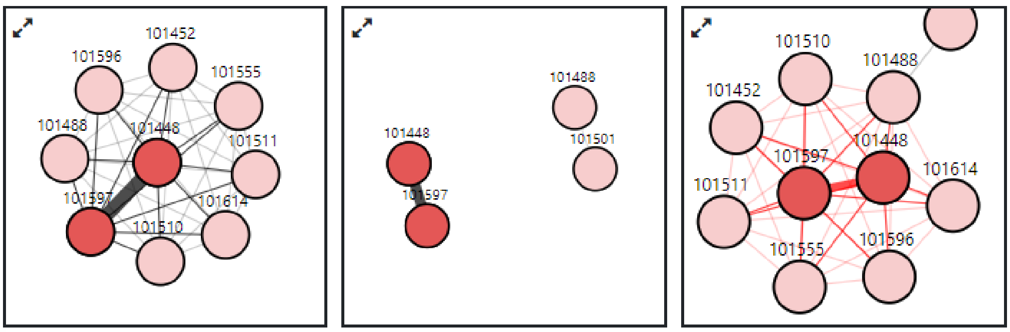


FIGURE 6.11: Social graphs of the red group that show interactions between users. The two outliers are selected which can be seen by the opaque color. Red edges indicate a decrease in social interactions. From left to right: (a) Social graph of time period 2 (July). (b) Social graph of time period 3 (October). (c) Social graph of the difference of (a) and (b).

red group from time period 2 and time period 3. In the social graph we can see that these two users often do activities together. This can be seen by the thickness of the line between the two users. Looking at the difference graph, the number of activities between these two users decreases. Yet, when looking at the Barcharts and Boxplot from Figure 6.7, the total number of activities for both users has increased. A reason why the number of social interactions decreases between these two people could be explained with the location. We have selected both users individually to see where they have done their activities in the third time period (see Figure 6.12 and 6.13). It can be seen that one user has done a lot of activities outside of Belgium, which could be the reason for the decrease in social interaction.

FIGURE 6.12: Location of user 101448 of the red group in time period 3.

FIGURE 6.13: Location of user 101597 of the red group in time period 3.

6.2.3 Buddy system

In this section we will look at the buddy system within the challenges of "Mol". The buddy system was a strategy introduced in these challenges to pair a passive and an active user with each other. The goal of the buddy system is to increase the activity of the passive user. We will investigate if this had any impact on the behavior of users. There are a total of 170 circles existing of two users that participated in the challenges of "Mol". We assume that if a circle exists of two users that it is a "buddy" circle. We must investigate all the circles individually to investigate whether a buddy system has worked or not. The visualization tool GameBus Insight cannot show more than 5 groups at a time so it took some time to find interesting circles. In the visualization tool we have looked at around 20 buddy circles that provided data during the whole campaign of "Mol". We selected only one time period that covered all the challenges of "Mol" (April till October) to investigate the data per user. A result of this time period for the total frequency can be seen in the Boxplot in Figure 6.14. It shows that one user per buddy circle is really active and the other user is really passive. This was something that holds true for a lot of buddy circles. There were also some interesting buddy circles that show that the buddy system worked for them. We have selected one buddy circle to show what we found.

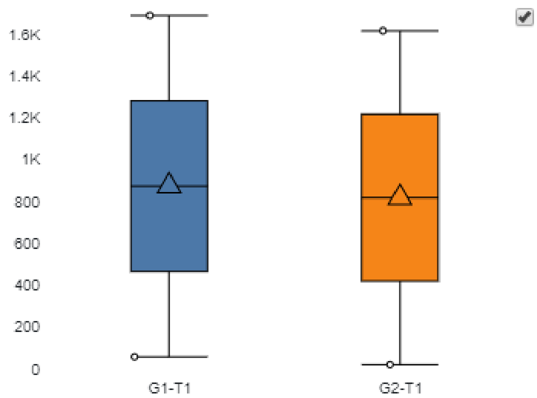


FIGURE 6.14: Boxplot of total frequency for two buddy circles. The time period that was chosen was the whole campaign of "Mol" (from April to October). It shows for both groups one really active user and one really passive user.

Figure 6.15 and 6.16 show together the first 6 months of the challenges in Mol. Every month is a time period. It can be seen that in the first month one user is not doing activities at all. In the second month the user started to do some activities

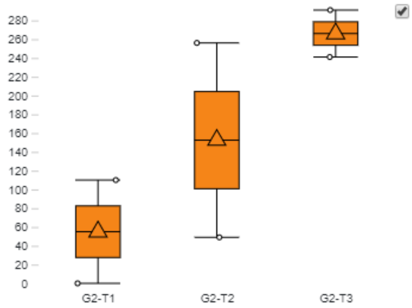


FIGURE 6.15: Boxplot of Total frequency of buddy circle of the first three months of Mol challenges

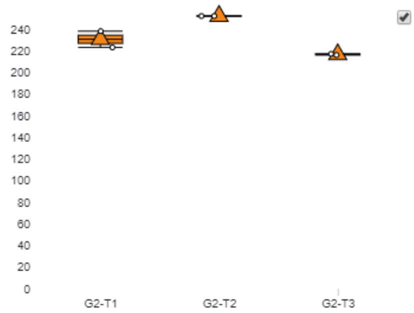


FIGURE 6.16: Boxplot of Total frequency of buddy circle of the next three months of Mol challenges

and after a few months the user is doing the same number of activities as the other user. The social graph also shows that these users are doing more activities together. In the first period there was no social interaction, therefore we compare time period 2 (May) and time period 3 (June) in Figure 6.17. The first social graph shows that these users did a total of 47 activities together. The passive user only did 49 activities in the second time period which means that they did almost all activities together. This grows in the third time period which can be seen by the right social graph. This shows the growth between time period 2 and time period 3. The active user has a positive effect on the user that was passive in the beginning. Figure

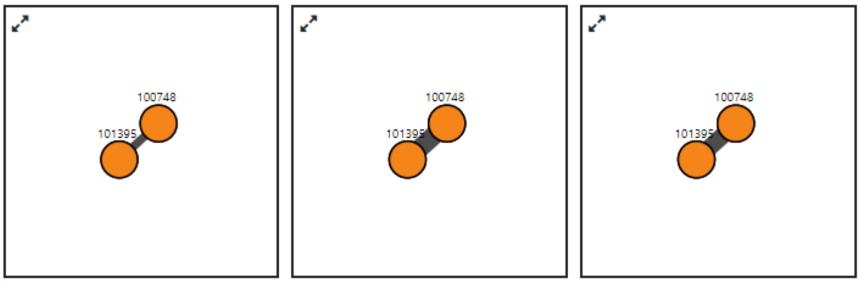


FIGURE 6.17: Social graphs of the buddy circle that show interactions between the two users. From left to right: (a) Social graph of time period 2 (May). (b) Social graph of time period 3 (June). (c) Social graph of the difference of (a) and (b).

6.18 shows where the activities have happened in the Netherlands per time period. It is interesting to see that from the second to third time period, activities are now more centralized to Mol, the city of the challenges. It could be the case that the users changed their sport location such that they can do more activities together.

FIGURE 6.18: Locations of the buddy circle within time period 1 (April), time period 2 (May) and time period 3 (June).

6.3 Evaluation

To verify that our tool works, we did a user evaluation study. Two end-users (data analysts), who are already familiar with the data, were invited to do the evaluation.

6.3.1 Setup

The user evaluation existed of two parts. In the first part the users will do several assignments with the tool while thinking out loud. The assignments were based on Task T1 up to T8 from Chapter 3. The user will have to make conclusions about the change in behavior between certain group with the help of these tasks. In this assignment we look at the outliers of the created groups, put them together and analyzed the outliers further. In depth questions were asked about the change in behavior to see if the user could perform the right actions to also see this change. The second part of the evaluation consisted of a questionnaire about the design of the application. The total evaluation took around 40 minutes. The template for the user evaluation can be found in Appendix F.

6.3.2 Results

The visualization tool has a steep learning curve. It helped that the users were already familiar with the data. Some assignments could be answered with multiple visualizations, but the right visualization could show it more easily or in more detail. Because it was the first time that the users worked with the tool a more suitable visualization could sometimes have been used. All assignments were answered correctly except for one. In this assignment you have to count the number of outliers. This can be done with the Boxplot visualization. Yet, the Boxplot shows data per time period. This means that users are drawn for each time period and are thus shown multiple times. When creating a new group it shows how many users you have actually selected. The answer was then corrected by the data analyst.

Both users had to be interrupted one time during the assignments. This was during the selection of time periods because the wrong months were selected. The selection of the month May can be seen in Figure 6.19. To select the month May you must select the space between April and May.

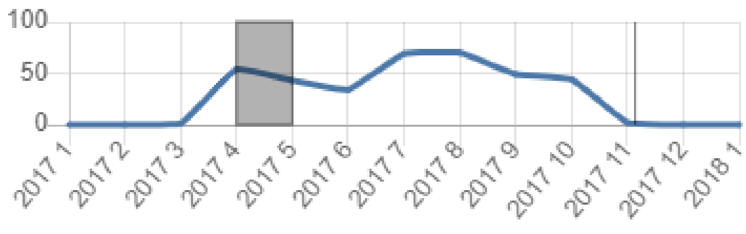


FIGURE 6.19: Linegraph of a measurement of behavior for one group between 2017 and 2018. The month May is selected as time period.

The last question of the assignment was if the users could find other conclusions using the tool. In this assignment one user went to see if he could find out where two users were doing activities together. He selected two users in the social graph and put them both in separate groups. The other groups were disabled by the data analyst to zoom in on these two groups. The map showed where both users did activities and a conclusion could be made were they did activities together. It became clear during this exercise that the data analyst saw the potential impact of the tool.

Table 6.1 displays the result of the questionnaire. We display the data analysts by two different colors. Questions and statements are rated according to the Likert scale [33]. Overall the data analysts were very pleased with the application. One

Questions:	**(Bad) 1**	**2**	**3**	**4**	**5 (Good)**
What is your general impression of the tool?				x	x
What do you think of how groups are selected?			x	x	
What do you think of how time periods are selected?			x	x	
	(Not at all) 1	**2**	**3**	**4**	**5 (Completely)**
Does the design meet your expectations?				x	x
	(Disagree) 1	**2**	**3**	**4**	**5 (Agree)**
The design is intuitive.			x		x

TABLE 6.1: Results of the questionnaire. We display the answers of the two data analysts by using two different colors.

data analyst was missing a search bar when selecting users which is why the question "What do you think of how groups are selected?" has scored less. Although circles and challenges are ordered on their id, if the number of circles and challenges become very large it still takes some time to find them. A search bar is a nice feature to add in the future.

A feature that is lacking according to one of the data analysts is to zoom in on the activities of the social graph. It would be helpful if you could filter on the sessions of the social graph to see which activities users did together. It is possible to show this within the tool by creating a measurement of behavior that only shows the activities of social interactions. However, it cannot be done without changing the measurements.

Another feature that the visualization tool lacked according to both data analysts was selecting more than 5 groups. When creating groups or subgroups there is a max of 5 groups. Looking back at the results with the buddy system, it could sometimes be handy to have more than 5 groups. Therefore we added a small fix in the application. In the configuration file of the application you can enter the colors of the groups. The default is the 5 colors given from Chapter 4. If you would like to add more groups you will need to add colors. The number of groups that you can make is dependent on the number of colors. The total number of colors that you can use is still limited. To distinguish groups, colors must be diverse within the visualization tool. A well-known tool for creating a color palette is ColorBrewer [34]. In the paper "ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps" [35] they explain that they limited the number of colors for their color palette to 12. According to ColorBrewer, this is the limit for the number of colors that can still be distinguished easily.

Chapter 7

Conclusion and Future work

7.1 Conclusion

We build a visualization tool for comparing behavior of groups existing of GameBus users. This thesis describes the visualization tool "GameBus Insight" that visualizes the event logs of GameBus. It was designed to show change in behavior between different time periods between (or within) groups. In this tool we express behavior with measurements. These measurements can be selected by the user outside of the visualization tool. With configurable measurements, we allow the tool to be flexible and ready for the future. Furthermore, this allows applications with a similar data structure as GameBus to also be investigated with GameBus Insight. The power of the tool is dependent on the chosen measurements to show different types of behavior that are available in the dataset. We have shown in the chapter Results that our tool can show change in behavior in multiple ways.

7.2 Limitations

This section describes the limitations of GameBus Insight:

- The number of groups is limited to 5. However, changing the number of colors in the configuration file will allow the user to create more groups. When the number of groups becomes too large visualizations will overlap.
- The number of time periods that can be created is limited to 3. This is done because the visualizations scale with the number of groups and time periods (groups * time periods).
- The Treemap-barchart allows to filter out sessions. These sessions are removed from the visualizations. The max number of sessions that you can remove in the visualization tool is set to 5000. This is limited because it has a great impact on performance. If you want to filter out more then you should alter the measurement cubes and filter with a where-statement.

7.3 Future work

In order to further improve the created visualization tool described in this thesis, one should focus on the following tasks:

- Build a search in the user component. This search should make it easier to search challenges, circles and users by ID.

- Zoom in on sessions of social graph to get more insight in what kind of activities users do together.
- Extend visualizations. The pagination in the Focuser component makes extensions in the visualization tool possible.
- Change the way how time periods are selected. In the evaluation we noticed that selecting time periods is not intuitive and that it should be changed.
- Show selected time periods in the Barchart. To reduce space we chose to remove the labels from the x-axis. When looking at the Barcharts is not immediately clear that there are multiple time periods.

The following items are more related to the dataset of GameBus than to the visualization tool:

- Change measurements. Explore the data by changing the measurements. Also new ways of measuring behavior are possible when new applications are added to GameBus. These measurements could show a different aspect of behavior.
- Update metabolic value and social interactions tables. These tables are created for the dataset that was used during this project. These tables need to be updated for new datasets.
- Metabolic values are now linked to a type of an activity. It would be better to link a metabolic value to a session. This would allow to take the speed of the activity into account and this will give a more precise intensity.
- Take location into account for social interactions. Again, this would provide more precise data.
- Fill NULL values of duration. Some activities have their duration filled in the activity name (e.g. "60 minute walk"). The property that should hold the duration is empty but can be generated automatically based on the activity name.

Bibliography

[1] A. Hruby and F. Hu, “The epidemiology of obesity: A big picture,” *PharmacoEconomics*, vol. 33, December 2014.

[2] M. Mack, “What drives rising health-care costs?” *Government Finance Review*, vol. 32, no. 4, pp. 27–32, Augustus 2016.

[3] “Fitbit,” 2019, https://www.fitbit.com/nl/app, last accessed in April 2019.

[4] World Health Organisation, “What is the WHO definition of health?” 2019, https://www.who.int/about/who-we-are/frequently-asked-questions, last accessed in April 2019.

[5] “Gamebus,” 2019, industrial Engineering & Innovation Sciences department of the TU/e, https://www.gamebus.eu, last accessed in June 2019.

[6] P. V. Gorp, “Gamebus, social health games for the entire family,” 2016, https://www.eitdigital.eu/fileadmin/files/CLC-Ehv/Gamebus.pdf, last accessed in June 2019.

[7] “Griddlers,” 2019, https://www.griddlers.net/nl/home, last accessed in April 2019.

[8] “Google fit,” 2019, https://www.google.com/fit/, last accessed in April 2019.

[9] A. Tizkar and N. M. Tabatabaei, “Rapid prototyping for software projects with user interface,” *Scientific Bulletin of University of PITESTI, Electronics and Computer Science Series*, vol. 2, p. 85, October 2009.

[10] G. Hemakumara and R. Ruslan, “Spatial behaviour modelling of unauthorised housing in colombo, sri lanka,” *KEMANUSIAAN*, vol. 25, pp. 91–107, September 2018.

[11] S. Michie, M. van Stralen, and R. West, “The behaviour change wheel: a new method for characterising and designing behaviour change interventions,” *Implementation science : IS*, vol. 6, p. 42, April 2011.

[12] L. Legault, *Self-Determination Theory*. Springer International Publishing, June 2017, pp. 1–9.

[13] K. Nowack, “Facilitating successful behavior change: Beyond goal setting to goal flourishing,” *Consulting Psychology Journal: Practice and Research*, vol. 69, April 2017.

[14] B. Fogg and J. Hreha, “Behavior wizard: A method for matching target behaviors with solutions,” June 2010, pp. 117–131.

[15] M. Gleicher, “Considerations for visualizing comparison,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, pp. 413–423, January 2018.

[16] M. Gleicher, D. Albers Szafir, R. Walker, I. Jusufi, C. Hansen, and J. Roberts, "Visual comparison for information visualization," *Information Visualization*, vol. 10, pp. 289–309, November 2011.

[17] "Tableau," 2019, https://www.tableau.com/, last accessed in June 2019.

[18] "Power bi," 2019, https://powerbi.microsoft.com/en-us/, last accessed in June 2019.

[19] T. Baranowski, J. Baranowski, D. Thompson, R. Buday, R. Jago, M. Griffith, N. Islam, N. Nguyen, and K. Watson, "Video game play, child diet, and physical activity behavior change," *American journal of preventive medicine*, vol. 40, pp. 33–38, January 2011.

[20] M. Campbell, I. Tessaro, B. DeVellis, S. Benedict, K. Kelsey, L. Belton, and A. Sanhueza, "Effects of a tailored health promotion program for female blue-collar workers: Health works for women," *Preventive medicine*, vol. 34, pp. 313–323, March 2002.

[21] J. Dzator, D. Hendrie, V. Burke, N. Gianguilio, H. Gillam, L. Beilin, and S. Houghton, "A randomized trial of interactive group sessions achieved greater improvements in nutrition and physical activity at a tiny increase in cost," *Journal of clinical epidemiology*, vol. 57, pp. 610–619, July 2004.

[22] C. Silva, D. Fassnacht, K. Ali, S. Gonçalves, E. Conceição, A. Vaz, R. Crosby, and P. Machado, "Promoting health behaviour in portuguese children via short message service: The efficacy of a text-messaging programme," *Journal of health psychology*, vol. 20, pp. 806–815, June 2015.

[23] J. Allen, J. Stephens, C. Dennison Himmelfarb, K. Stewart, and S. Hauck, "Randomized controlled pilot study testing use of smartphone technology for obesity treatment," *Journal of obesity*, vol. 2013, p. 151597, December 2013.

[24] V. Burke, N. Giangiulio, H. Gillam, L. Beilin, and S. Houghton, "Burke v, giangiulio n, gillam hf, beilin lj, houghton s. physical activity and nutrition programs for couples: a randomized controlled trial," *Journal of clinical epidemiology*, vol. 56, pp. 421–432, June 2003.

[25] S. Venu, G. Lolla, and L. Hoberock, "On selecting the number of bins for a histogram," *The seventh international conference on data mining, Las Vegas*, November 2019.

[26] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," in *Proceedings 1996 IEEE Symposium on Visual Languages*, September 1996, pp. 336–343.

[27] R. McGill, J. W. Tukey, and W. A. Larsen, "Variations of box plots," *The American Statistician*, vol. 32, no. 1, pp. 12–16, 1978.

[28] R. Vliegen, J. Wijk, and E.-J. van der Linden, "Visualizing business data with generalized treemaps," *IEEE transactions on visualization and computer graphics*, vol. 12, pp. 789–796, September 2006.

[29] "Mapbox," 2019, https://www.mapbox.com, last accessed in December 2019.

[30] "Cube.js," 2019, https://cube.dev/, last accessed in November 2019.

[31] B. Ainsworth, W. Haskell, S. Herrmann, N. Meckes, D. Bassett Jr, C. Tudor-Locke, J. Greer, J. Vezina, M. WhittGlover, and A. Leon, "The compendium of physical activities tracking guide. healthy lifestyles research center, college of nursing & health innovation, arizona state university," 2019, https://sites.google.com/site/compendiumofphysicalactivities/, last accessed in November 2019.

[32] "Games for health europe," 2017, website about the Mol Campaign in 2017, https://www.gamesforhealtheurope.org/speaker/pieter-van-gorp/, last accessed in January 2020.

[33] R. Likert, "A technique for measurement of attitudes," *Archives of Psychology*, vol. 22, January 1932.

[34] "Colorbrewer," 2019, http://colorbrewer2.org/, last accessed in December 2019.

[35] M. Harrower and C. Brewer, "Colorbrewer.org: An online tool for selecting colour schemes for maps," *Cartographic Journal The*, vol. 40, pp. 27–37, June 2003.

Appendix A

Measurements of behavior

Below you find all the measurements of behavior that were found in the examined papers during the Literature review. Measurements including their unit of measure are registered per paper. Measurements about food were filtered out. Every paper was examined by two people. In the perfect scenario, every measurements would be filled in twice, once per student. These double registrations are also filtered out. It is still possible that some double measurements can still be found because of different descriptions.

Paper	Measurement	Unit
Allen-et-al-2013	Weight	Count (= integer)
Allen-et-al-2013	Waist circumference	Count (= integer)
Allen-et-al-2013	Physical activity	Duration: minutes
Allen-et-al-2013	physical activity	Duration: minutes
Allen-et-al-2013	Body Mass Index based on weight and height measurements in light clothing using a stadiometer and balance scale	kg/m2
Allen-et-al-2013	Waist circumference	Length: centimeters
Allen-et-al-2013	Physical activity derived from Stanford 7-Day Physical Activity Recall	nan
Allman-Farinelli-et-al-2016	Change and frequency of physical activity	Count (= integer)
Allman-Farinelli-et-al-2016	Change in weight (measured by dietitian during intervention self-measured during maintenance period)	Weight: kilograms
Allman-Farinelli-et-al-2016	Change in BMI	kg/m2
Allman-Farinelli-et-al-2016	Frequency of physical activity	Count (= integer)
Allman-Farinelli-et-al-2016	Duration of physical activity	Duration: minutes
Baranowski-et-al-2011	Physical activity	Count (= integer)
Baranowski-et-al-2011	Physical activity sedentary (1-100 counts per minute of accelerometer)	Duration: minutes
Baranowski-et-al-2011	Physical activity light (101-2999 counts per minute of accelerometer)	Duration: minutes
Baranowski-et-al-2011	Physical activity moderate (bigger or equal than 3000 counts per minute of accelerometer)	Duration: minutes

Baranowski-et-al-2011	Physical activity mean counts per minute	Duration: minutes
Baranowski-et-al-2011	BMI percentile	BMI percentile
Baranowski-et-al-2011	BMI z-score	z-score
Baranowski-et-al-2011	Triceps measurement	Distance: millimeter
Baranowski-et-al-2011	Waist circumference	Distance: centimeter
Baranowski-et-al-2011	Height	cm
Baranowski-et-al-2011	Weight	kg
Baranowski-et-al-2011	Moderate to vigorous physical activity [SR]	Duration: minutes
Bauer-et-al-2010	Healthy exercise behavior [SR]	None
Bauer-et-al-2010	Positive emotions to SMSMT	None
Bauer-et-al-2010	Change in children's BMI-SDS at stages in the treatment	BMI-SDS difference
Bauer-et-al-2010	Physical activity per week	Count (= integer)
Bennett-et-al-2010	Body weight at 12 weeks	kg
Bennett-et-al-2010	BMI	kg/m^2
Bennett-et-al-2010	Blood pressure control	mm HG
Bennett-et-al-2010	Waist circumference	cm
Bennett-et-al-2010	Website logins	Count (= integer)
Burke-et-al-2003	Weight & Height	nan
Burke-et-al-2003	Waist circumference	nan
Burke-et-al-2003	Total cholesterol and high-density lipoprotein cholesterol	nan
Burke-et-al-2003	Physical fitness was measured as the Physical Work Capacity at 75% of maximum heart rate (PWC75) using submaximal testing on bicycle ergometers	nan
Burke-et-al-2003	Physical activity was assessed at baseline at the end of intervention and at 12-month follow-up using a 7-day recall and a 14-day recall	nan
Campbell-et-al-2002	Physical activity	Count (= integer)
Campbell-et-al-2002	Smoking	Count (= integer)
Campbell-et-al-2002	Cancer screening	Count (= integer)
Campbell-et-al-2002	health behaviors	Count (= integer)
Campbell-et-al-2002	Body Mass index	Count (= integer)
Campbell-et-al-2002	Exercise yes no	Percentage
Campbell-et-al-2002	aerobic physical activity (walking jogging swimming biking aerobic dancing other dancing aerobic exercise classes)	MET-hours per week
Campbell-et-al-2002	strengthening and flexibility activities (lifting weights/strength exercises stretching/flexibility exercises and other exercise)	MET-hours per week
Doyle-et-al-2008	BMI-Z	Count (= integer)
Doyle-et-al-2008	BMI	Count (= integer)

Doyle-et-al-2008	Weight	Count (= integer)
Doyle-et-al-2008	Weight concern	Count (= integer)
Doyle-et-al-2008	Shape concern	Count (= integer)
Dzator-et-al-2004	Physical activity	Count (= integer)
Dzator-et-al-2004	Costs	Money
Dzator-et-al-2004	Consequences	Euro
Dzator-et-al-2004	Physical activity accumulation on the most days	Duration: minutes
Dzator-et-al-2004	BMI	BMI
Dzator-et-al-2004	Exercise days	Days per week
Dzator-et-al-2004	Fitness adjusted for body weight	W/kg
Elliot-et-al-2007	Healthy dietary behavior	Count (= integer)
Elliot-et-al-2007	Positive dietary social support	Count (= integer)
Elliot-et-al-2007	Peak oxygen uptake	ml/kg/min
Elliot-et-al-2007	Sit-ups in 1 min	Count (= integer)
Elliot-et-al-2007	Healthy physical activity behavior	Count (= integer)
Elliot-et-al-2007	Physical activity beliefs and understanding	Count (= integer)
Elliot-et-al-2007	Positive physical activity social support	Count (= integer)
Elliot-et-al-2007	Body weight	Lbs
Elliot-et-al-2007	Body mass index	BMI
Elliot-et-al-2007	Overall well-being	Count (= integer)
Elliot-et-al-2007	Weight	kg
Elliot-et-al-2007	Sit-ups	Count/minute
Emmons-et-al-1999	Participation in regular exercise (self-reported)	nan
Emmons-et-al-1999	Long-term smoking cessation was measured as self-report of abstinence from cigarettes for the 6 months prior to the final survey	
Fassnacht-et-al-2015	Extent of reporting behavior discussed below	nan
Fassnacht-et-al-2015	Daily time spend on average participating in activities such as fast walking swimming ball games	Duration: hours
Fassnacht-et-al-2015	Duration spend in front of a screen	Duration: minutes
Fassnacht-et-al-2015	Satisfaction questionnaire	-
Fassnacht-et-al-2015	Submitting of required SMS	Percentage
Fjeldsoe-et-al-2016	Body weight	kg
Fjeldsoe-et-al-2016	Waist circumference	cm
Fjeldsoe-et-al-2016	Participation in moderate-vigorous physical activity (MVPA)>30minutes per week [SR]	Count (= integer)
Fjeldsoe-et-al-2016	FFBQ questionnaire [SR]	Count (= integer)
Hallberg-Ranerup-Kjellgren-2015	Patients self-reported experience of using the self-management system for 8 weeks regarding	

Hallberg-Ranerup-Kjellgren-2015	Daily answers on self-report questions concerning lifestyle well-being symptoms medication intake and side effects	Qualitative
Hallberg-Ranerup-Kjellgren-2015	Results of home blood-pressure measurements	Qualitative
Hallberg-Ranerup-Kjellgren-2015	Reminders and motivational messages	Qualitative
Hallberg-Ranerup-Kjellgren-2015	Access to a web-based platform for visualization of the self-reports	Qualitative
Hartman-et-al-2016	Weight reduction	kg
Hartman-et-al-2016	Increase in physical activity	nan
Hartman-et-al-2016	Engaging in MVPA	Hours/week
Jensen-et-al-2016	Body weight	lbs
Jensen-et-al-2016	standardized BMI	BMI
Kim-et-al-2015	Weight	Kg
Kim-et-al-2015	Percent body fat	percent
Kim-et-al-2015	Physical activity level	MET-minutes/week
Kim-et-al-2015	Obesity-related QOL (Quality of life)	Count (= integer)
Kim-et-al-2015	Weight loss after the treatment	kg
Kim-et-al-2015	Change of percent body fat	Percentage points
Kim-et-al-2015	Obesity-related quality of life	Unknown
Kim-et-al-2015	Korean version of the International Physical Activity Questionnaire-Short Form	Unknown
Kim-et-al-2015	Self reported satisfaction and acceptance of the text message program using a 5-point Likert rating	strongly agree to strongly disagree
Leahey-et-al-2014	weight loss at the end of the 3-month program	kg
Leahey-et-al-2014	Percentage of individuals who achieved a 5% weight loss	percentage
Leahey-et-al-2014	Cost-effectiveness of interventions	cost of intervention/mean weight loss in kg
Leahey-et-al-2014	Self monitored diet and PA	Weight control practices questionnaire
Leahey-et-al-2014	Counting calories and fat grams	Weight control practices questionnaire
Leahey-et-al-2014	Self weighing	Weight control practices questionnaire
Majumdar-et-al-2013	"A 41- item online instrument called Eat-Move was developed to measure frequency and amount of the targeted behaviors and the demographic variables."	Count (= integer)
Majumdar-et-al-2013	completion of level	Count (= integer)
Majumdar-et-al-2013	Physical activity	nan
Pagoto-et-al-2015	Weight	Weight: kilograms

Pagoto-et-al-2015	Social support for weight loss using the Weight Management Support Inventory	nan
Piatt-et-al-2013	Change in weight	kg
Piatt-et-al-2013	Changes in CVD risk factors including glucose triglyceride levels waist circumference blood pressure values and HDL cholesterol	multiple
Rodearmel-et-al-2006	Weight	Kilograms
Rodearmel-et-al-2006	BMI (based on weight height and age)	kg/m2
Rodearmel-et-al-2006	percentage body fat estimates were calculated from skinfold thickness measurements taken using Lange calipers	nan
Rodearmel-et-al-2006	Steps per day	Count (= integer)
Shapiro-et-al-2008	Acceptability	Count (= integer)
Shapiro-et-al-2008	Attrition	Count (= integer)
Shapiro-et-al-2008	Adherence	Count (= integer)
Shapiro-et-al-2008	Minutes spent on excersising on average over the past week for each day	Count (= integer)
Shapiro-et-al-2008	SSB consumption on average over the past week for each day	Count (= integer)
Shapiro-et-al-2008	Minutes watched TV on average over the past week for each day	Count (= integer)
Shapiro-et-al-2008	Number on your pedometer	Count (= integer)
Shapiro-et-al-2008	How many SSB per day	Count (= integer)
Shapiro-et-al-2008	Screen time per day	Count (= integer)
Shuger-et-al-2011	Body weight	KG
Shuger-et-al-2011	BMI	KG/(m^2)
Shuger-et-al-2011	Body fat	Percent
Shuger-et-al-2011	Physical activity	Kcals/day
Shuger-et-al-2011	Physical activty	Count (= integer)
Silva-et-al-2015	Fruit/vegetable consumption	Daily portions
Silva-et-al-2015	Physical activity	hours per day
Silva-et-al-2015	Screen time	hours per day
Silva-et-al-2015	Satisfaction questionnaire	score
Silva-et-al-2015	Messages sent	Count (= integer)
Sze-et-al-2015	Weight	lb
Sze-et-al-2015	Dietary measures	
Sze-et-al-2015	Self-reported effectiveness on the helpfulness usability ease of use of the implementation of the web-based system	nan
Sze-et-al-2015	Daily consumed calories	kcal
Sze-et-al-2015	Self reported rating of the event's valence salience arousal frequency and vividness	Likert scale
Tate-Jackvony-Wing-2006	Body weight	Weight: Kilograms

Tate-Jackvony-Wing-2006	Height	Centimeters
Tate-Jackvony-Wing-2006	Physical activity was measured using the activity questionnaire of Paffenbarger et al	nan
Tate-Jackvony-Wing-2006	Mean Weight Loss	KG
Tate-Jackvony-Wing-2006	Physical activity	KCAL/week
Tate-Jackvony-Wing-2006	Number of logins	Count (= integer)
Tate-Jackvony-Wing-2006	Waist circumference	cm
Tate-Jackvony-Wing-2006	Sufficient physical activity [SR]	kcals burned/week
Thomas-Leahey-Wing-2015	Weight loss after 3 and 6 months	kg
Thomas-Leahey-Wing-2015	Number of weeks participants logged in to the website	Count (= integer)
Thomas-Leahey-Wing-2015	Sufficient physical activity [SR]	min/week
Turner-McGrievy-et-al-2009	Weigth + Height	nan
Turner-McGrievy-et-al-2009	knowledge of weight-loss topics	nan
Turner-McGrievy-et-al-2009	physical activity (short Inter- national Physical Activity Questionnaire)	nan
Turner-McGrievy-et-al-2009	user control	
Van-Drongelen-et-al-2014	Fatigue measured using the 20-item Checklist Individual Strength	nan
Van-Drongelen-et-al-2014	Sleep quality measured using the Jenkins Sleep Scale	nan
Van-Drongelen-et-al-2014	General perceived health self-reported	nan
Van-Drongelen-et-al-2014	Nutritional behavior self-reported	nan
Van-Drongelen-et-al-2014	Hydration self-reported	nan
Van-Drongelen-et-al-2014	Amount of physical activity per week self-reported	nan
Winett-et-al-2007	Percent kcal from fat	Percent
Winett-et-al-2007	Weight	Kg
Winett-et-al-2007	Nutrition self-regulation	nan
Winett-et-al-2007	Planning and tracking physical activity	nan
Winett-et-al-2007	Fitting physical activity into daily routine	nan
Winett-et-al-2007	Changes in Body weight	kg

Winett-et-al-2007	Changes in nutrition behavior	nan
Winett-et-al-2007	Changes in physical activity behavior	-
Winett-et-al-2007	Steps per day	Count (= integer)
Wing-et-al-2006	Weight (weekly)	Kilograms
Wing-et-al-2006	Weight change	Kg
Wing-et-al-2006	Amount of weight lost	kg
Wing-et-al-2006	Calories expended in physical activity	kcal
Wing-et-al-2006	Self-weighing	Count (= integer)
Wing-et-al-2006	Adherence to prescribed behavioral strategies	Unknown

Appendix B

Model to database model

Figure B.1 shows the most important tables in the GameBus database. This diagram was simplified to our model to make Chapter 3 easier to read. In our model we change the name of *game_session* to *session*, *game_descriptor* to *activityType* and *property_instance* to *value*. Furthermore, we removed a the tables *conditions*, *challengerules*, *pointmappings*, *challenge*, *participations*, *personal_point* and *data_provider* to simplify the model even further.

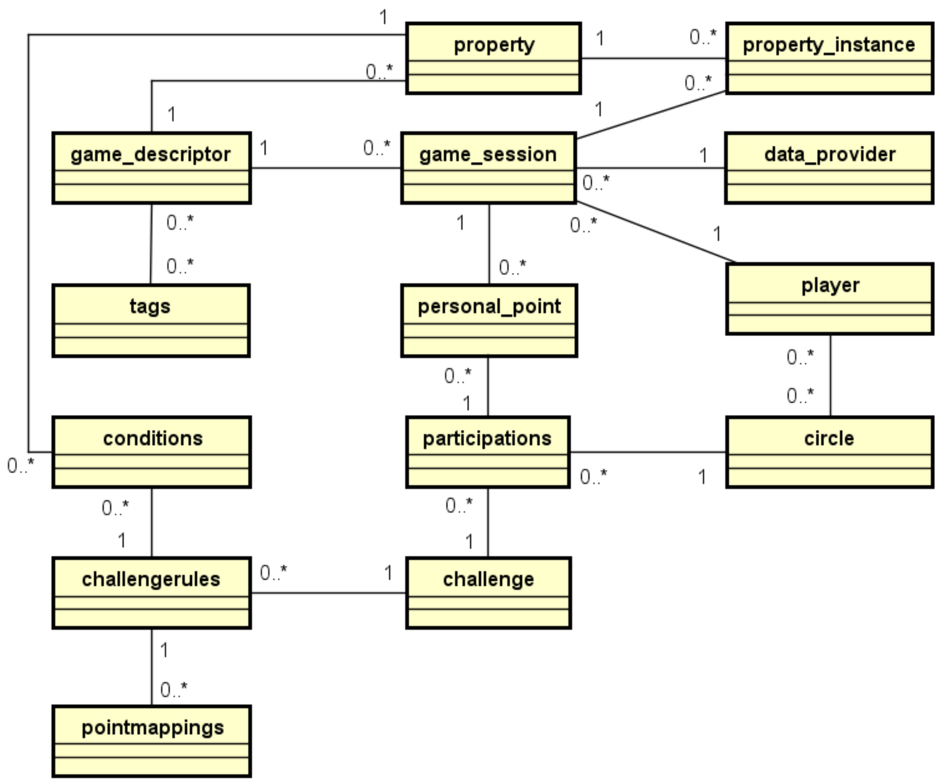


FIGURE B.1: Database diagram with most important tables of GameBus

Appendix C

Frequency of properties

Below you find all the 128 properties that can be found in the given GameBus dataset ordered by the frequency that they occur. The two extremely active users from the data analysis are filtered out because we want to see what property is used most by the average user. The "Total linked activity types" column shows how many activity types can give data for this property. The "Activity types that gave data" column shows how many activity types actually gave data. With activity type we hint to the term *game_descriptor* as described in Appendix B.

	Property name	Frequency	Total linked activity types	Activity types that gave data
1	total time (seconds)	94544	11	6
2	when	72456	7	3
3	location	50873	6	5
4	distance (meters)	38462	4	3
5	calories burned	28421	5	4
6	activity type	25285	6	2
7	score	19732	7	7
8	start time	17800	2	1
9	steps	17720	6	2
10	entry mode	17193	1	1
11	tracking mode	17193	1	1
12	tracking source	17193	1	1
13	runkeeper live	16922	1	1
14	equipment	16922	1	1
15	activity group	16607	1	1
16	description	16172	8	7
17	sports type	14291	1	1
18	bonus	13780	5	5
19	unlock	13608	5	5
20	level	13600	5	5
21	missed	13347	4	4
22	background	13345	4	4
23	wrong	12924	4	4
24	group size	12000	1	1
25	number of generations	11981	1	1
26	selfie code	11627	1	1
27	duration (seconds)	7384	4	4
28	notes	6359	5	4
29	penalty	5338	1	1

30	solving time (seconds)	5338	1	1
31	puzzle ID	5338	1	1
32	type	5337	1	1
33	end location	4470	3	3
34	start location	4467	3	3
35	sports type (run; bike; ...)	3612	2	1
36	Primary hashtag	3560	2	2
37	URL of Post	3549	3	2
38	maximum speed (km/h)	3243	2	1
39	ID of other player	3076	2	2
40	avg heartbeat per minute	3008	2	1
41	max heartbeat	2939	2	1
42	is team member?	2726	1	1
43	age of other player	2724	1	1
44	step count	2580	2	1
45	image location	1983	1	1
46	note	1283	2	2
47	avg heartrate per minute	929	1	2
48	amount of water (ml)	875	1	1
49	rest heart rate	819	1	1
50	weight (kg)	786	1	1
51	Runkeeper user ID	750	1	1
52	elevation (meters)	750	1	1
53	evidence URI (computer)	750	1	1
54	evidence URL (human)	750	1	1
55	Public Post?	696	2	1
56	hours slept	670	1	1
57	member age	254	1	2
58	blood sugar (mmol/l)	207	1	1
59	fat percentage	202	1	1
60	is new topic?	197	1	1
61	what did you do?	128	3	3
62	for father?	114	1	1
63	for kids?	114	1	1
64	for mother?	114	1	1
65	for pets?	114	1	1
66	for grandparents?	84	1	1
67	blood pressure (systolic)	83	1	1
68	for yourself?	82	1	1
69	blood pressure (diastolic)	81	1	1
70	4. TEAMWORK	51	1	1
71	1. PERSONAL VALUE	51	1	1
72	6. LEADERSHIP	51	1	1
73	3. COMMUNICATION	51	1	1
74	5. PROBLEM SOLVING	51	1	1
75	2. SELF MANAGEMENT	50	1	1
76	BMI (kg/m2)	48	1	1
77	temperature (celcius)	40	1	1

78	did you win?	31	1	1
79	voluntarily?	22	1	1
80	amount contributed	17	1	1
81	website	16	1	1
82	Out of bed time	12	1	1
83	Overall sleep efficiency	12	5	1
84	Post	12	2	1
85	Total sleep duration (minutes)	12	5	1
86	Wake up time	12	5	1
87	caught new Pokemon?	12	1	1
88	got medal?	12	1	1
89	leveled up?	12	1	1
90	2.2 initiative	4	1	1
91	5.3 manage conflict	3	1	1
92	2.3 resilient	3	1	1
93	4.1 respectful	3	1	1
94	5.4 negotiation	3	1	1
95	1.1 integrity	3	1	1
96	2.4 persistence	3	1	1
97	4.2 assertive	3	1	1
98	1.2 trustworthy	3	1	1
99	4.3 accept criticism	3	1	1
100	6.1 coaching	3	1	1
101	1.3 people focused	3	1	1
102	3.1 listening	3	1	1
103	4.4 facilitation	3	1	1
104	6.2 strategic	3	1	1
105	1.4 business focused	3	1	1
106	3.2 storytelling	3	1	1
107	6.3 empathic	3	1	1
108	3.3 sociable	3	1	1
109	5.1 conceptual thinking	3	1	1
110	6.4 delegation	3	1	1
111	2.1 self learning	3	1	1
112	3.4 written communication	3	1	1
113	5.2 creativity	3	1	1
114	Bed entry time	0	4	0
115	Bed exit time	0	4	0
116	Caption	0	1	0
117	contacts added	0	1	0
118	end time	0	1	0
119	Fall asleep time	0	4	0
120	influences added	0	1	0
121	Latency after wake-up (minutes)	0	4	0
122	Latency before sleep (minutes)	0	4	0
123	more information	0	1	0
124	Number of naps	0	4	0
125	start date	0	1	0

126	Tags	0	1	0
127	themes added	0	1	0
128	Total nap duration (minutes)	0	4	0

Appendix D

Database indexes

Two extra non-clustered indexes were created on the database of GameBus to improve the performance of the application. These indexes can be seen in Listing D.1.

```
create index property_instance_game_session ON property_instance (property,
    game_session, value);

create index game_session_game_descriptor ON game_session (game_descriptor,
    player, created_at);
```

LISTING D.1: Indexes that were put on the database

Appendix E

Installation guide

E.1 File structure

in Figure E.1 you see a simplified file structure of the visualization tool. The base file exists of three folders: client, server and extra data. The server folder contains

```
Visualization tool
├─ client
│  ├─ public
│  │  └─ index.html
│  └─ src
│     ├─ Component
│     │  ├─ CSS
│     │  └─ JS
│     ├─ Data
│     ├─ config.js
│     ├─ index.js
│     └─ index.css
├─ server
│  ├─ schema
│  └─ index.js
└─ extra data
   ├─ MetabolicValueTable.sql
   ├─ SocialCircleTable.sql
   └─ socialgroups.py
```

FIGURE E.1: Simplified file structure of the application

the file *index.js* which starts up the cube.js server. The schema folder contains all the different cubes that can be used as different measurements. The most important part about the client folder is the component folder. In here you find for every component a separate CSS and JavaScript file. The extra data folder exists of files for adding the extra data (Metabolic value and social interactions) to your database.

E.2 Run visualization tool

Before you can start the tool you need to install the required packages. We used for this the JavaScript development tool *npm* (`https://www.npmjs.com/`). Run the command *npm install* within the base, client and server folder.

In the server folder, edit the hidden file ".env" to have your MySQL database credentials.

Next, add the extra data to the database (Metabolic value and social interactions). Metabolic value is not necessary if you change the measurements. Social interactions table is necessary for showing the graph. If you use a different dataset than this table should be updated. In your database, run the SQL from the following files: "MetabolicValueTable.sql", "SocialCircleTable.sql" which can be found in the folder "extra data". If you want to update the socialcircle table with new game sessions, look at the file "socialgroups.py". This is a python script that creates data for the table socialcircle according to the given game sessions.

To run the application we go to the base folder and we run the command *npm start*. This will start up the server and the client concurrently. The web page will start after the command is done. If the web page is not displaying, go to `http://localhost:3006`.

E.3 Changing cubes

Cubes can be found in the Schema folder of the server. You can add new files or alter cubes that are already there. Cubes exists of the variables *name*, *sql*, *measures* and *dimensions*. The name is important for the client for creating a connection. We come back to that in "Measurement_settings file". With the sql you can join the tables that you want from your dataset. A cube has the following requirements: *measures* has the property called *total* and that these dimensions are always in the *dimensions* property:

```
dimensions: {
    createdAt: {
       type: ‘time‘,
       sql: ‘created_at‘
    },
    user: {
      type: ‘string‘,
      sql: ‘player‘
    },
    dateActivity: {
      type: ‘string‘,
      sql: ‘dateActivity‘
    },
    dayOfWeek: {
      type: ‘number‘,
      sql: ‘dayOfWeek‘
```

```
    },
    hourOfDay: {
      type: ‘number‘,
      sql: ‘hourOfDay‘
    },
    type: {
      type: ‘string‘,
      sql: ‘tagname‘
    },
    gamesession: {
      type: ‘number‘,
      sql: ‘gs_id‘
    }
}
```

LISTING E.1: Required dimensions

The "type" dimension is the labelling of a measurement. The client-side expects the data to have a "type" which is used for the Treemaps.

If you create a cube with a new name you need to change this is the *measurement_settings* file which can be found in the Data folder of the client.

E.4 Measurement_settings file

In this file you can change the cubes, group colors and initial start-date and end-date of the application. Changing group colors influences the amount of groups you can make. Cubes are represented by a *CubeName* which is needed for communication with CubeJs and a *Name* which is used in the application above the linegraph. *CubeName* should be the same as the name from the cube in "server/schema".

Appendix F

User evaluation form

Date: **Start-time:** **End-time:**

Assignment:

1. Create 3 different groups for circles: 2565, 2573, 2575.
2. Change the names of the groups accordingly. (name: "Circle " + circle id)
3. Select two time periods of two months. The first time period existing of the months April and May, the second period existing of the months June and July.
4. Which group(s) have overall a negative change in behavior between the two time periods?
5. How many outliers are there for the measurement total frequency? Select the outliers and create a new group. (note: creating groups can be done by right-clicking a user)
6. For the outlier group. Which user (user-id) has grown the most physically?
7. For the outlier group. The total duration of light intensive activities (metabolic value smaller than 4) has increased between the two time periods. On which day(s) did this have the most impact?
8. For the outlier group. One user stops using the app for almost a month in between challenges. Which user is this? (note: use daily Linegraph)
9. Are the outliers doing activities together? If yes, has this increased between the two time periods?
10. For the outliers, which users have done the most activities together and from which circle are they? (note: users can be selected in the graph)
11. Open question. What other conclusions can you make from the selected groups and time periods?

Questions:

What is your general impression of the tool?	(bad) 1 2 3 4 5 (good)
Does the tool meet your expectations?	(not at all) 1 2 3 4 5 (completely)
What do you think of how groups are selected?	(bad) 1 2 3 4 5 (good)
What do you think of how time periods are selected?	(bad) 1 2 3 4 5 (perfect)
The design is intuitive.	(disagree) 1 2 3 4 5 (agree)
Are you planning to use the tool in the future?	Yes / No

Which visualization is most important to you?

What features does the tool lack?

Comments/Notes:

Appendix G

Social circle script

Below you find the created script that was used for finding the social interactions between users. It takes the gamesesions as csv as input. It is important that the dates of the sessions are sorted in ascending order else the algorithm will not work. The output of the script is a json file with all social interactions.

```
import csv
import json
import pandas as pd
from datetime import datetime, timedelta

df = pd.read_csv('gamesessions.csv')

records = []

for index, row in df.iterrows():
    start_date = datetime.strptime(row['created_at'], '%Y-%m-%d %H:%M:%S')

    start_date_minus_one_min = start_date - timedelta(minutes = 1)
    start_date_plus_one_min = start_date + timedelta(minutes = 1)

    for index2, row2 in df.iloc[index + 1:].iterrows():
        date2 = datetime.strptime(row2['created_at'], '%Y-%m-%d %H:%M:%S')

        if start_date_minus_one_min <= date2 and start_date_plus_one_min >=
            date2 and row['game_descriptor'] == row2['game_descriptor'] and
            row['player'] != row2['player']:
            rec = {}
            rec['player'] = int(row['player'])
            rec['player2'] = int(row2['player'])
            rec['created_at'] = row['created_at']
            rec['created_at2'] = row2['created_at']
            rec['game_session'] = int(row['id'])
            rec['game_session2'] = int(row2['id'])
            records.append(rec)
        else:
            break

out = json.dumps(records)
f = open('socialcombinations.json', 'w')
f.write(out)
```

LISTING G.1: Python script for creating social circles with gamessesions as input.