BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

EXPLORATORY ANALYSIS OF BIG DATA IN "JMENOMESTO"

EXPLORATIVNÍ ANALÝZA VELKÝCH DAT VE HŘE "JMENOMESTO"

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **VÁCLAV TRAMPEŠKA**
AUTOR PRÁCE

SUPERVISOR **FEDERICO NICOLÁS LANDINI**
VEDOUCÍ PRÁCE

BRNO 2021

Brno University of Technology
Faculty of Information Technology

Department of Computer Graphics and Multimedia (DCGM) Academic year 2020/2021

Bachelor's Thesis Specification

Student: **Trampeška Václav**
Programme: Information Technology
Title: **Exploratory Analysis of Big Data in "jmenomesto"**
Category: Data Mining

Assignment:

1. Get acquainted with the game "jmenoměsto", the database of the game and the information stored in it.
2. Get acquainted with statistical tests.
3. Perform simple queries and analyses to gain insight into the behavior of players.
4. With the aid of available data processing toolkits or libraries, create a software that can perform more complex queries.
5. Perform analyses focusing on one or more of the following aspects: a. cross-lingual analyses of words and/or categories, b. geocultural analyses of words and/or categories, c. evolution of players across time
6. Extend the software to present the results in a visually engaging manner relevant to the choice from the previous point.
7. Extend the software with a tool that can help the game owners to improve the game in terms of fun, playability, revenue or all.

Recommended literature:

- Statistical Analysis Handbook, Michael John de Smith. Winchelsea Press, 2018

Requirements for the first semester:

- Items 1, 2 and 3. Item 4 should be in progress.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor: **Landini Federico Nicolás**
Head of Department: Černocký Jan, doc. Dr. Ing.
Beginning of work: November 1, 2020
Submission deadline: May 12, 2021
Approval date: October 30, 2020

Abstract

This thesis is concerned with the analysis of the database of the online game *Jméno, město*. In this game, players are tasked with adequately answering given categories with answers beginning with a given letter. The thesis analyzes the evolution of player behaviour over the lifetime of the game and the behaviour of players in different countries and cultures within the same and different languages based on the popularity of different answers. A web application was developed to facilitate the execution of these analyses, allowing easy-to-use data collection and data visualisation in charts without requiring knowledge of the database structure and a query language. The results of the proposed analysis methods are compared with Google Trends data to identify the similarities between the data observed in the game and internet searches. The comparison shows that the proposed methods can give meaningful results, and hence the database can be suitable for performing further specific analyses. Furthermore, partly based on the analysis results, the thesis proposes changes to improve the game in player experience and revenue generation.

Abstrakt

Tato práce se zabývá analýzou databáze online hry *Jméno, město*. V této hře mají hráči za úkol adekvátně odpovídat na zadané kategorie odpovědmi začínající na zadané písmeno. Práce analyzuje evoluci chování hráčů během existence hry a chování hráčů v různých zemích a kulturách v rámci stejného i jiného jazyka na základě popularity různých odpovědí. Pro zjednodušení vykonávání těchto analýz byla vyvinuta webová aplikace umožňující snadnou kolekci a vizualizaci dat v grafech bez nutnosti znalosti databázové struktury a dotazovacího jazyka. Výsledky navržených analyzačních metod jsou porovnávány s daty služby Google Trends za účelem zjištění podobností mezi daty sledovanými ve hře a internetovým vyhledáváním. Z toho porovnání vyplývá, že navržené metody mohou dávat smysluplné výsledky a databáze tedy může být vhodnou pro vykonávání dalších konkrétně zaměřených analýz. Práce dále, i na základě výsledků analýz, navrhuje změny pro zlepšení hry v rámci hráčského prožitku a výdělečnosti.

Keywords

Big data, data analysis, online game, world cultures, player behaviour, web application

Klíčová slova

Velká data, datová analýza, online hra, světové kultury, chování hráčů, webová aplikace

Reference

TRAMPEŠKA, Václav. *Exploratory Analysis of Big Data in "jmenomesto"*. Brno, 2021. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Federico Nicolás Landini

Rozšířený abstrakt

Online hra Jméno, město vychází v pravidlech ze stejnojmenné deskové hry. V této hře dostanou hráči zadané počáteční písmeno a seznam kategorií, na které v určitém časovém limitu musí vymyslet odpovědi začínající na dané písmeno. Pro práci byla poskytnuta kopie databáze obsahující záznamy o hrách, odehraných kategoriích a vyplněných odpovědích ze všech jazykových verzí hry. Těchto odpovědí se v databázi nachází ve 46 různých jazycích celkově téměř jedna miliarda. Jedná se o vzorek nebývalých rozměrů a je tak vhodným kandidátem na explorativní analýzu odpovědí a chování hráčů.

Jedním z cílů práce je prozkoumat tuto databázi. Tento proces se skládá z několika kroků. Nejprve je nutné zvolit typ neboli metodu analýzy, který určuje, co bude předmětem analýzy a jakým způsobem může být vykonána. Tato práce navrhuje typů hned několik. Jmenovitě se jedná o analýzu změny počtu hráčů v čase, mezijazykovou analýzu odpovědí a kategorií, geokulturní analýzu odpovědí a kategorií a analýzu trendů oblíbenosti odpovědí v čase. Každý typ analýzy má své výhody stejně tak jako svá úskalí, která je nutné při vykonávání analýzy uvažovat. Dalším krokem procesu je sběr dat adekvátní vybrané metodě. Tato data potom jsou potom interpretována pomocí popisu a grafické vizualizace a následně vyhodnocena. Práce aplikuje každou z navrhovaných metod. U metod porovnávajících odpovědi hráčů jsou výsledky analýz porovnány s údaji o počtech vyhledávání na vyhledávači Google za pomocí služby Google Trends. Na základě výsledků porovnání s těmito údaji je nakonec vyhodnocena i míra použitelnosti zvolené metody. Ta je ovlivněna nejen správností návrhu metody a postupu, ale i kvalitou dat obsažených v databázi.

Každá z vykonávaných analýz se zaměřuje na konkrétní problém či premisu. Analýza změny počtu hráčů v čase zjišťuje, že pro rozšíření hráčské základy v jednotlivých jazykových verzích je klíčové rozšíření povědomí o hře, v čemž nejčastěji hrály klíčovou roli videa na platformě YouTube s miliony zhlédnutí obsahujících záznamy ze hraní hry a živá vysílání na platformě Twitch. Do počtu hráčů se také výrazně promítla světová pandemie choroby covid-19 v roce 2020, kdy byl zaznamenán nárůst počtu her napříč všemi jazykovými verzemi ve stovkách až tisících procent. Geokulturní analýza se zabývá rozdíly v kulturách různých anglicky mluvících zemích. Pro tuto analýzu byly vybrány Spojené státy americké, Kanada, Spojené království a Austrálie. Na základě odpovědí hráčů z těchto zemí je závěrem analýzy konstatování, že kultury v těchto zemích jsou v dnešní době již velmi podobné, ale přesto jsou v nich obvykle nalezitelné rozdíly. Čím více jsou země geograficky vzdálené, tím více jsou tyto rozdíly patrné. Mezijazyková analýza řeší rozdíly mezi populárními odpověďmi v češtině a slovenštině. Ty jsou velmi často podobné, avšak stejně jako u předchozí analýzy je možné najít odlišnosti. Z dat vyplývá, že slovensky mluvící hráči mají povědomí o odpovědích populárních v češtině, ne nutně tomu ale je i naopak. Nakonec analýza trendů oblíbenosti odpovědí v čase zkoumá výrazy související s pandemií v roce 2020 a popularitu aplikace TikTok. Výsledky analýz ve většině případů korelovaly s údaji ze služby Google Trends. Všechny metody se tedy ukázaly jako schopné přinášet smysluplné výsledky. Tyto metody by tím pádem mohly být využity pro další analýzy.

Další část práce se zaměřuje na vývoj aplikace sloužící pro zjednodušení procesu sběru dat a vykonávání analýz. Tato aplikace pomocí jednoduchého a přehledného uživatelského prostředí umožňuje uživateli zvolit, jaká data chce z databáze vybrat, a poté tyto data zobrazuje v grafu, který je možné dále upravovat a exportovat. Stejně tak lze volitelně zobrazit i tabulku s nalezenými daty, které je možné také exportovat. Aplikace umožňuje sbírat data nutná pro vykonávání analýz a vygenerované grafy především ulehčují evaluaci kvality dat pro vybranou analýzu. Aplikace byla důležitým nástrojem pro zkvalitnění a zrychlení procesu provádění výše zmíněných analýz.

Posledním cílem práce je rozšířit aplikaci o nástroj, který by zlepšil hru samotnou v oblasti hráčského zážitku a výdělečnosti. Ke hře ale nakonec nebyl udělen žádný přístup, vyvinutá aplikace funguje plně samostatně. Tento cíl tedy nebylo možné v plné míře splnit. Místo rozšíření aplikace tedy práce přináší návrhy pro zlepšení zmíněných oblastí. Tyto návrhy byly sestaveny na základě osobních zkušeností s hraním hry a výsledku analýzy změny počtu hráčů. Práce navrhuje změny v designu uživatelského prostředí a v hratelnosti, přidání podpory integrace diváků živých vysílání na platformě Twitch do hraní hry a doporučuje celkově větší zaměření na tuto platformu. Pro možné zvýšení výnosů přichází s návrhem vytvoření sponzorovaných kategorií a marketingové kampaně zaměřené na anglicky mluvící hráče, u kterých povědomí o hře zatím není příliš rozšířené.

Exploratory Analysis of Big Data in "jmenomesto"

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Federico Nicolás Landini. Supplementary information was provided by Ing. Jan Poul, one of the owners and the creator of the online game. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Václav Trampeška
May 10, 2021

Acknowledgements

I would like to thank my supervisor Mr. Landini for his thorough guidance, valuable advice and time spent on consultations.

Contents

Chapter 1

Introduction

Jméno, město is an online party game in which players propose an answer starting with a given initial letter to each category in each round of the game. Players come from all different parts of the world, from different cultures and speak different languages. It can be assumed that depending on the player's origin, the most frequent answers to the categories also change. These differences can then be searched for and used to compare life in different countries or cultures. With time, various new trends also emerge globally, and these are also reflected in player behaviour.

A copy of the database containing records of games played, categories played, and answers filled in from all the 46 languages of the game was provided for the thesis. The database contains almost one billion player responses, which is a sample of unprecedented size and is thus a good candidate for exploratory analysis of player responses and behaviour.

The thesis has several objectives. The first is to examine the provided database using various analysis methods and evaluate the methods' results and validity. Another objective of the thesis is to develop an application to simplify the process of collecting data and performing analyses. The last objective of the thesis is to propose improvements to the game in terms of player experience and revenue generation.

Chapter 2 explains the rules of the game *Jméno, město*. First, the original board game rules, then the adaptation and necessary gameplay changes for the online version of the game. Chapter 3 describes the game database and its contents. In Chapter 4, the theoretical foundations of the performed analyses are laid. It covers data collection methods, analysis, visualization and evaluation. Chapter 5 performs specific analyses and evaluates their results. The development of the application from the initial analysis of requirements to the implementation is described in Chapter 6. The last chapter, Chapter 7, describes suggestions for game improvement.

Chapter 2

Principles of the Game Jméno, město

In order to understand the content of this thesis, it is necessary to start with its very foundation – the game *Jméno, město*. This chapter covers an explanation of the rules of the game *Jméno, město* and the adaptation of the rules to the online version[1] of the game.

Rules and Gameplay of the Board Game

Jméno, město is a creative-thinking party game for two or more players. Players are provided with a letter and a set of categories to answer with words that start with the given letter. Points are assigned to each player according to their correct answers to the categories and their uniqueness. The player with the highest score after a certain amount of rounds wins the game.

Before the game can be started, players have to agree on the set of categories and on what will terminate the round. The two most common ways of ending a round are either setting up a time limit, which ends the round after a timer runs out, or playing until one of the players says the word "STOP". Once this rule is settled amongst the players, the game can start.

The round begins with one player saying the first letter of the alphabet loudly and then reciting the rest of the alphabet in their mind until a different player says the word "STOP". The letter being silently recited at that moment will be the letter all of the answers of the round will have to start with. The player tells the starting letter to other players and the round is started. Each player attempts to think of and write down a word that fits each of the categories until the round is finished.

Afterwards, all the answers are checked. The correctness of the answers (valid terms for the categories) is agreed upon amongst the players, and incorrect answers are crossed out. If the category is answered solely by a single player, that player earns 15 points. If the category is answered by more than one player, players who give an answer that nobody else chose get 10 points each. Players who choose an identical answer get 5 points each. After a selected amount of rounds is finished, each player's points are counted, and the winner of the game is determined.

[1] https://jmenomesto.net

Figure 2.1: Answering categories in the game on jmenomesto.net.

Changes to Rules and Gameplay in the Online Version

The gameplay of the browser version of the game stays more or less faithful to the original. However, due to the absence of players physically being in the same room and communicating with each other, some changes were necessary to make.

The decision over the used categories, amount of rounds and round termination is not discussed within the group of players but rather handed to a single player who selects these options in the game settings. The draw of the initial letter has to be handled differently as well, and the letter is therefore selected randomly by the game engine. The set of letters present in the random draw can be altered before the start of the game, too.

Figure 2.1 shows the screen during a round. The players can see what round is being played, how many rounds are going to be played in total, the initial letter and the timer. Beneath that, category names and text fields for answers are situated. Finally, the stop button that finishes the round is placed at the very bottom of the page.

Another part of the game that is more difficult to replicate online is the validation of answers and the separation of answers that fit the category from answers that do not. This is handled by letting the players vote on the answers after the end of a round. Players can either check or uncheck an answer, and if an answer gets less than a half of positive votes

(checks) out of all votes, it is ruled out and not taken into account for point evaluation. Finally, the points that each player gained in the rounds are counted, and the winner is announced.

After a game finishes, the players have the option of playing another game with the same options and categories. This pre-selection removes the need to fill in all settings again and allows players to play multiple games in sequence without unnecessary delays.

Chapter 3

Game Database Overview

The game stores its data into a PostgreSQL database. This chapter explains what such a database is, what data are stored by the game, and how the game database is structured.

3.1 Relational Model

A relational model is a method for storing data[22]. The most popular programming language for implementing a relational model is the Structured Query Language (SQL).

The relational model separates data into tables with columns and rows. In relational model terminology, such a table is called a relation, a column is called an attribute, and the rows of the table are called records or tuples. Each relation represents an entity relevant to the application, and each record of a relation represents an instance of its corresponding relation. A relation is an unordered collection of data items, therefore re-arranging the records in any order does not change the relation. The collection of all relations is called a relational database.

Each record consists of a set of values called properties representing attributes of the relation by which this record is modelled. An attribute that each record has to contain is its own unique key. This key is called the primary key (PK) and guarantees that each record of a relation is distinguishable from the rest of the records in the relation. Connections between two different relations are implemented by a record being linked to a record from the other relation. This link is accomplished by adding a foreign key (FK) attribute to the relation. A foreign key is a primary key on a different relation and represents either a one-to-one or a one-to-many relationship. Many-to-many relationships are resolved by creating a new table with attributes being two primary keys from both of the relations. The combination of both keys creates a new primary key, thus making the relationship an entity.

3.2 SQL

SQL[12, 23] is a standardized query language used in relational databases. SQL input consists of a sequence of commands which are composed of statements. Some of the most common statements are listed below[1].

- The statement `SELECT` is used to extract a collection of attributes from a table.

[1] A complete list of statements with explanations can be found at: https://www.postgresql.org/docs/10/sql-commands.html

- The `FROM` statement specifies a table.
- The statement `WHERE` filters the records to be yielded by the command by a condition.
- The statement `ORDER BY` controls the order in which records appear.
- The statement `LIMIT` trims the number of records returned by a specified amount.

An example of a SQL query command can be seen in Listing 3.1.

```
SELECT
    id,
    name,
    games_played
FROM
    category
WHERE
    id_lang='en'
ORDER BY
    games_played DESC
LIMIT 10;
```

Listing 3.1: SQL query returning ten most played categories in English.

PostgreSQL

PostgreSQL[11] is an object-relational database system distributed for free via open-source. PostgreSQL extends the SQL language and is designed to be extensible. For example, custom data types, functions, functional languages, and index types can be defined or build. A vast majority of the standard SQL features is supported, although with occasional differences in syntax or function.

Figure 3.1: Entity relationship diagram of the game database.

3.3 Game Database

Due to the fact that the actual game database contains sensitive information, the game owners provided a trimmed copy of the database that excludes such information. The schema of the provided database is presented in Figure 3.1. Some of the relations contain redundant information or could be modelled differently for a clearer and more well-structured database, but that analysis is out of the scope of this thesis.

The integral parts of the database are the relations "game", "round", "category", "word", "usr", "game_player", and "lang".

You can find the description of the main relations and their most relevant attributes below. The full list of relations with all of their attributes is described in Appendix A.

Game

This relation stores data about each game of *Jméno, město*. The "game" relation has a many-to-many relationship with the "category" relation. This means that a game can have multiple categories, and at the same time, a category can be used in multiple games. The "game" relation also has a one-to-many relationship with the "round" relation. This means that a game can consist of multiple rounds.
The most important attributes for the analysis are:

- `id` – [PK] unique game record identifier
- `id_usr` – [FK] identifier of the user who created the game
- `id_lang` – [FK] identifier of the language the game is played in

Round

This relation stores data about each round of a single game. The "round" relation has a one-to-many relationship with the "word" relation. This means that there are multiple words answered within a round.
The most important attributes for the analysis are:

- `id` – [PK] unique round record identifier
- `starts_with` – initial letter that all answers have to start with in a round
- `id_game` – [FK] identifier of the game a round is a part of

Category

This relation stores data about each category in the whole game. The "category" relation has a many-to-many relationship with the "game" relation. This means that a category can be used in multiple games, and at the same time, a game can have multiple categories. The "category" relation also has a one-to-many relationship with the "word" relation. This means that there can be multiple answers to a category.
The most important attributes for the analysis are:

- `id` – [PK] unique category record identifier
- `name` – name of the category

- `id_usr` – [FK] identifier of the user who created the category
- `games_played` – amount of games the category had been selected in
- `id_lang` – [FK] identifier of the language the category was created in

Word

This relation stores data about each answer to a category in the whole game.
The most important attributes for the analysis are:

- `id` – [PK] unique answer record identifier
- `value` – string containing the submitted answer
- `id_round` – [FK] identifier of the round the word was answered in
- `date_cr` – timestamp of the submission
- `id_category` – [FK] identifier of the category the word was used as an answer for
- `id_game_player` – [FK] identifier of the player who sent the answer

Usr

This relation stores data about each user of the site that has played at least one game of *Jméno, město*. The "usr" relation has one-to-many relationships with the "game_player", "game" and "category" relations. This means that a user can play multiple games and also create multiple new games and new categories.
The most important attributes for the analysis are:

- `id` – [PK] unique user record identifier
- `lang` – language the user is playing in
- `country_code` – country in which the user is located

Game Player

This relation stores data about each player of a single game of *Jméno, město*. The "game_player" relation has a one-to-many relationship with the "word" relation. This means that a player can send multiple answers.
The most important attributes for the analysis are:

- `id` – [PK] unique game player record identifier
- `id_round` – [FK] identifier of the game the player is a part of
- `id_usr` – [FK] identifier of the "usr" table record this player belongs to
- `country_code` – country in which the user is located
- `flag_robot` – flag that states whether the user is a real person or a robot
- `lang` – language the user is playing in

Lang

This relation stores data about each of the supported languages. The "lang" relation has one-to-many relationships with the "category" and "game" relations. This means that a language could be used in multiple games and categories.
The most important attributes for the analysis are:

- `default_alphabet` – altered alphabet of the language that does not contain letters too challenging to play with
- `show_name` – name of the language in the respective language
- `id` – [PK] unique two-character language code

Chapter 4

Analysis Process and Visualisation

The analysis of the game database aims to gain insight into the behaviour of players. The database provides an opportunity to analyse nearly a billion answers from players from numerous countries and cultures. These answers can be compared, and the differences may provide new findings regarding different cultural environments.

This chapter explains the process of the analysis and the visualisation of data.

4.1 Evaluation of Useful Data

Statistics[17] is a science field that focuses on collecting, analysing, and reporting information from the world around us. It involves the analysis of data, which are obtained by a measurement process. The process of measurement ought to warrant that the results are consistent, accurate, representative and reproducible. Some factors that influence the quality of measurements are:

- framework – producing measurements can be divided into two types of processes: technical and philosophical. The former includes the tools and procedures that are used to collect and store data. The same dataset may result in varying quality of data depending on the technical framework. The latter is the representation of these data in a meaningful way.
- temporal effects – measurements that are made in different circumstances at different times are bound to yield different results. A generally constant process with differences observed as seemingly random fluctuations can be described as being *stationary*. A process where a trend exists is called *non-stationary*. The copy of the game database provided for the analysis only contains records as of a certain date. Even so, it is subject to temporary influences.

The measurement results come in values, such as counts or sets of decimal values, that are of different types or scales. Some of the fundamental scales are:

- nominal – this term refers to data values that do not hold any numerical meaning. The values are assigned to named classes but without the possibility of arithmetic calculations or sorting. An example of this scale in the game database would be the categories.
- ordinal – these are data values that can be ordered in the sense that Class 1 may be better than Class 2. There is no indication of a numerical interval between the

two classes, therefore arithmetic operations do not make sense. For example, when comparing different candy bars, a person may like some more than the others, but there is no way of knowing how big the difference exactly is.

- interval – numerical data that can be ordered and for which the interval between any two objects on the scale can be measured.

What is unique to this game database is that the frequency of answers per category and letter is influenced by the frequency of the other answers. A higher frequency of an answer to one initial letter does not necessarily mean that this entry is more popular than an answer to the same category with a different initial letter. For example, the answer "elephant" for the category "Animals" is the most frequent answer for this category overall. However, this information cannot be interpreted to mean that the elephant is the most popular animal. The main reason for this answer being so common is that it is difficult to think of another animal beginning with the letter "E". The occurrence of an answer is also influenced by how often the category and letter are played. Therefore, answers must be compared within the same context, which means the same language, category and initial letter.

Before starting the analysis, it is essential to consider whether the data sample to be analysed is large enough. If the selection of the language, the country of origin, the category or the initial letter returns a small sample, the results of such an analysis are prone to be unrepresentative.

4.2 Data Collection

Data collection was performed using SQL database scripts over the copy of the game database. The analysis was performed over a database that was further modified for the purpose of the proposed application, which is explained in detail in Section 6.3. The most notable modification for the purpose of the analysis is that the responses from the bot players were filtered out.

When counting the occurrences of words, the case of the letters of the word was not taken into account; all words were converted to lower case. No further changes were made to the data, which can lead to inconsistent results in some languages.

Some letters appear multiple times with different diacritics within a language's alphabet. An example is the Czech alphabet with letters such as "e", "é", and "ě". The players often ignore the diacritics and send the answers without them, which leads to the inconsistency of answer occurrences. Unfortunately, ignoring the diacritics during data collection is not a solution to this problem due to the fact that the spelling of some words of different meanings differs only by the diacritics. Thus, it would not make sense to ignore the diacritics and combine the occurrences of all kinds of word spellings into one.

Another issue that affects the quality of the data collection results is that some words are more difficult for players to spell correctly, so misspelt answers are often found in the database. These errors then negatively affect the frequency of occurrence of correct word spellings.

Some answers also have the same meaning, although they are written differently, which further degrades the quality of the analysed data. Another situation that can arise is that one word has different meanings in different countries within the same language. For

example, the word “football” in answer to the category “Sport” conveys a different sport in the US than it does in the UK.

Together, the above effects can cause significant discrepancies in the results. As an example of these effects, Figures 4.1 and 4.2 show the most frequent answers for the category “Jméno”, starting with “V”. This category requires players to answer a first name. In Czech, as in many languages, the same first names have many other nicknames and variants, which is reflected in the results. Figure 4.1 shows the results precisely as they appear in the database. However, these results do not correctly reflect which names are the most popular among players, as they appear multiple times in different forms. When semantically identical answers are aggregated, as shown in Figure 4.2, the results change drastically. For example, the answer “Vojta” suddenly appears in fourth place instead of its original eighth place, and the occurrence of the answer “Václav” has doubled. New items have also entered the top fifteen answers due to the deletion of the same-meaning responses.

Nevertheless, the unification of the items does not solve all the problems. Some responses cannot be reliably merged together. For example, the answer “Vláďa” and variants with different diacritics, which together account for 0.61% of the responses, can be associated with both the name “Vladimír” and the name “Vladislav”. It is not clear which of these names, if any, were intended by which player and therefore the occurrence of this response cannot be unified in any way. Some variants of the same answer even begin with different letters, such as the names “Josef” and “Pepa”, which further devalues the usefulness of unifying answers.

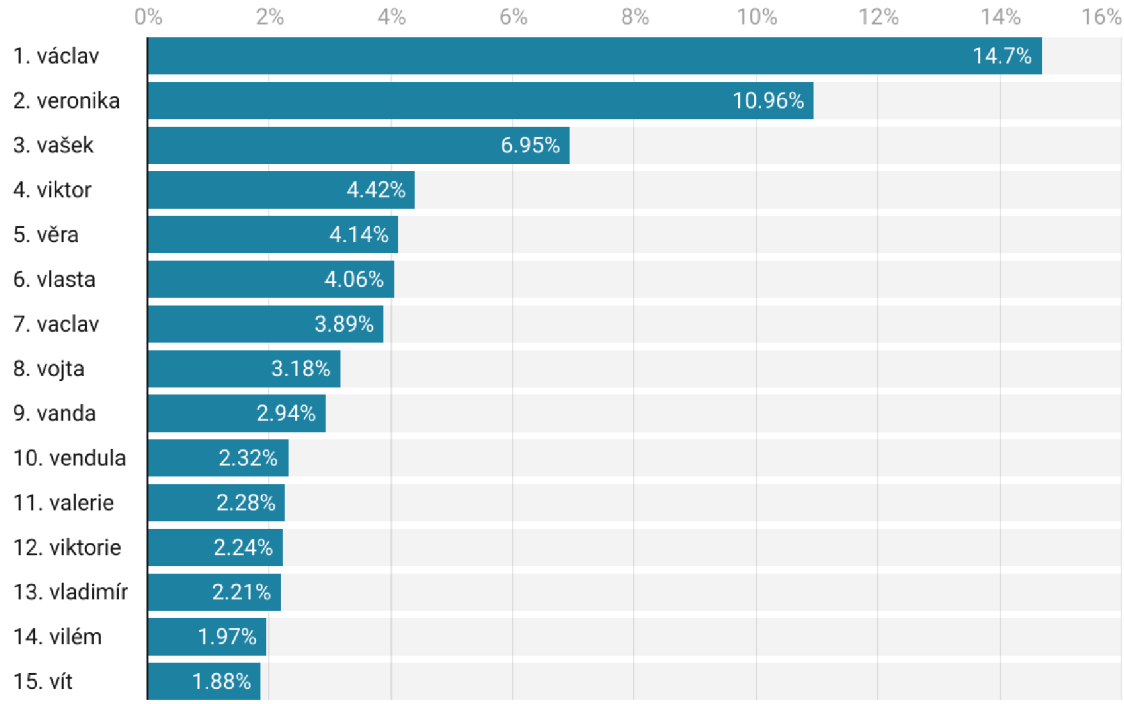

Figure 4.1: The 15 most popular answers in Czech to the category “Jméno” starting with “V” as they are in the database.

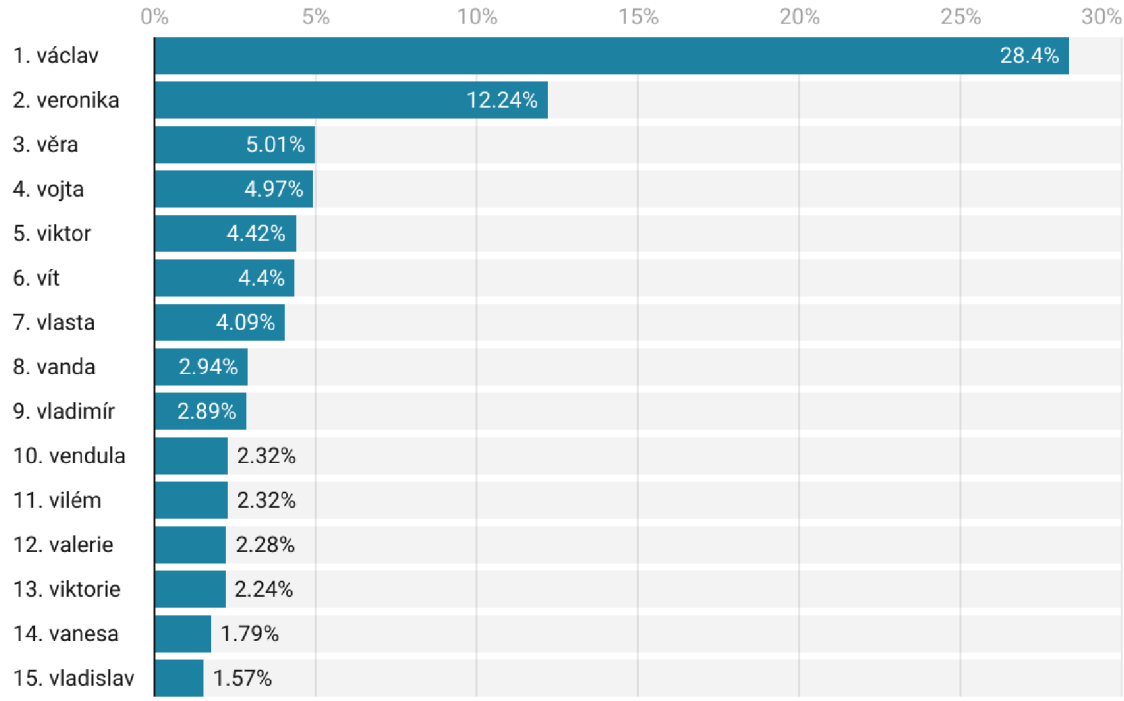

Figure 4.2: The 15 most popular answers in Czech to the category "Jméno" starting with "V" after all name and spelling variations have been unified.

For the above reasons and because the resulting problems could invalidate the results of the analyses, the answers are not modified in any way during the analyses and are taken as they are in the database.

4.3 Data Analysis Methods

Data analysis can be carried out in different ways using different analysis methods. This section presents the proposed analysis methods and their range of applications.

Evolution of Players Across Time

As time passed and new language versions of the game were introduced, the number of players has changed over time. A large number of external factors influences fluctuations in the number of players. It is essential to recognise these factors and thus put the changes in the number of players into context.

Geocultural Analysis of Answers and Categories

This method fully allows a pure comparison of answer occurrences, provided that the answers belong to the same language, category and initial letter. Thus, the popularity of answers from players from different countries can be compared. This comparison provides insight into differences in player behaviour in different geocultural environments.

Cross-lingual Analysis of Answers and Categories

As mentioned in Section 4.1, the frequency of the answers is significantly influenced by the frequency of other correct answers for a given category and letter. Thus, the indication of the occurrence of an answer without proper context holds little to no value. An example could most clearly explain this problem. In the Czech and Slovak equivalents of the "food" category, the answer corresponding to the word "banana" has a 9% share among Czech answers and 18% among Slovak. However, this does not mean that bananas are more popular with the Slovak players. Other answers in Czech include popular answers such as the words for potatoes (brambory) or blueberries (borůvky), which do not start with "B" in Slovak. Because of such words, "banana" has a smaller share of answers in this example. This problem invalidates any direct comparison of the popularity of the answers. It would be extremely difficult to calculate an exact value suitable for comparison, and therefore the answers are not directly compared by percentage. A possible way in which such an analysis can be done to some extent is by finding the most frequent responses in one language and then finding the translations of those responses in the other language. If such a response is typical in one language and rare in the other, then it might indicate that it is a response known only to players of one language. However, even in such an analysis, the risks of the inaccuracy of the analysis must be considered, and its conclusions should be treated with caution. This effect makes cross-lingual analysis of answers essentially unfeasible. Not only is a comparison of the occurrences of equivalent answers in different languages not objective, the translation of an answer from one language to another is often not entirely accurate. The answer may have many other meanings in the language, and it is impossible to know which player meant which one.

Comparing the number of games with a particular category is also highly biased. The choice of categories is heavily influenced by the fact that the game only offers a few basic categories on the main game creation page, with the rest hidden behind a button. Only players who are registered and logged in can choose from these additional categories or create a completely new one. A large number of players will therefore choose from these pre-made categories. Moreover, these selected categories differ in each language version of the game, so comparing how often what categories are played in different languages is *a priori* meaningless.

Answer Popularity Trends Across Time

Just as the number of players changes over time, so does the popularity of the answers. Due to the different total number of answers on different days, the popularity of answers must be considered as a percentage out of the total number of answers.

Soon after the deployment of a new language version of the game, the percentage of answer occurrences undergoes large fluctuations due to the small total number of answers on a given day. However, this effect gradually stabilizes as more games are played, and more answers are submitted. The popularity of answers over time is otherwise influenced solely by external factors. It is most affected by factors that directly affect a large number of players from the region or culture under consideration.

4.4 Data Presentation

Data visualisation and interpretation of results is an indispensable part of any analysis. As data are highly susceptible to manipulation and can be easily misinterpreted or misused during visualisation and presentation, it is essential to remain objective and not overlook factors that influence the results to confirm or refute a premise. The presentation must not be misleading.

When interpreting the results of analyses focusing on the responses of individuals, the demographics should be taken into account, i.e. the age and social groups from which the people who provided responses come.

Graphical Visualisation

To be meaningful, statistical graphs should clearly indicate what the scales are, whether the graph starts with zero or a different value, and how the values were obtained, which mainly involves stating on what dataset and time period the results are based. For graphs that show change over time, great emphasis must be placed on the starting and ending dates. Shifting the dates changes the number of presented values. This change can significantly affect the interpretation of the results without any change in the values themselves.

The primary goal of the graph is to present the data clearly. The analysis of the *Jméno, město* database uses bar and line graphs for this purpose. To maintain a bar graph's clarity, it is preferable to keep the number of bars between 5 and 9, as a large number of bars would be challenging to display and confusing to interpret.

Chapter 5

Data Analysis and Findings

This chapter includes performing analyses over the game database. It introduces the processes of each analysis using the methods described in Section 4.3 and presents their results.

Prior to describing the individual analyses, it is important to keep in mind the players' demographics. As is to be expected for online games, the majority of users are likely to be relatively young, averaging around 30 years old. Presumably, a slight majority of players are male, estimated at around 60%[18, 21]. In terms of social status, the financial situation of the users is good enough to allow them to own a mobile phone or a computer with a stable internet connection. With this in mind, it cannot be assumed that the findings of this chapter apply to the entire discussed population since the player base is only a specific part of that population.

This chapter is divided into four parts: an analysis of the evolution of players across time, a geocultural analysis, an analysis of answer trends and a cross-lingual analysis.

5.1 Evolution of Players Across Time

This section focuses on the analysis of the evolution of the number of players and games played. The number of games can be affected by many internal and external factors. This analysis is concerned with external factors that organically influence the number of games without any intervention by the owners. According to the game owners, the game has never been promoted by any marketing campaign or advertising and it is therefore not considered in the analysis. The only internal factor influencing the numbers is that the game has been unavailable several times due to server downtime.

Figure 5.1 shows the accumulated number of games in the seven most played languages over time. The start date of the graph is December 1, 2014, when the very first game of *Jméno, město* was played. The end date of the graph is September 14, 2020, the date of the copy of the database that was provided for this thesis. The data for this date is incomplete due to the copy not being taken at exactly the end of the day, but this does not substantially affect the analysis of the results.

The highlighted part of the graph marks the first wave of the COVID-19 pandemic in Europe. For the purposes of this graph, March 9, 2020 was chosen as the start date of the highlighted area. On this date, Italy became the first European country to announce

a nationwide lockdown[1]. June 3, 2020, when the Italian lockdown was ended[2], was selected as the end date. By this time, most of the strictest nationwide restrictions in most European countries had already been lifted. This highlighted part is not intended to precisely delimit the duration of the first wave of the pandemic. Given that some of the selected languages have a significant proportion of players outside Europe and the course of the pandemic was different in each country, it would be difficult to identify this time period globally.

As illustrated by the graph, German has been the most played language for most of the game's existence. It was overtaken by Spanish during the first wave of the pandemic on April 28, 2020. The global pandemic has caused a significant increase in games played in all languages.

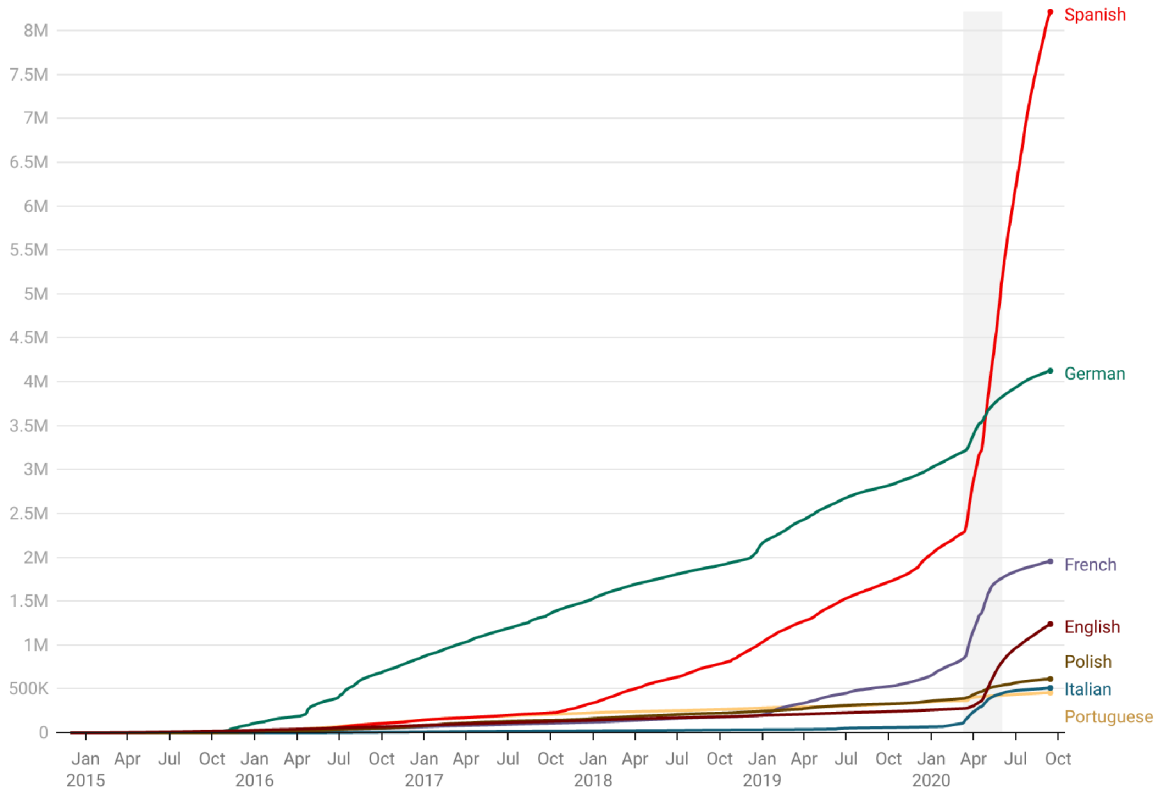

Figure 5.1: Cumulative amounts of games played in the seven most popular language versions over time.

Table 5.1 compares the peak average number of games per day during the pandemic with the average number of games per day before the pandemic outbreak in Europe. Different countries had restrictions at different times, and therefore the week with the highest number of games played was selected to compare the average number of games played during the pandemic. As the table shows, the increase in the number of games per language is in the hundreds to thousands of percentage points. This increase is most likely due to the fact

[1] https://www.theguardian.com/world/2020/mar/09/coronavirus-italy-prime-minister-country-lockdown

[2] https://www.huffpost.com/entry/italy-europe-coronavirus-travel-restrictions_n_5ed772c0c5b6fafa9260d526

Language	Average games/day before pandemic	Peak week	Average games/day during the peak week	Change of games/day
German	2,721	23 - 29 March 2020	12,806	+371 %
Spanish	3,459	25 - 31 May 2020	49,446	+1,330 %
French	2,315	17 - 23 March 2020	20,895	+802 %
English	244	30 April - 6 May 2020	12,048	+4,843 %
Polish	401	25 - 31 March 2020	2,931	+631 %
Italian	845	11 - 17 March 2020	6,704	+693 %
Portuguese	114	20 - 26 March 2020	3,850	+3,273 %

Table 5.1: Change in the average number of games before and during the pandemic in the seven most popular language versions. For the average number of games played before the pandemic, four full weeks from February to early March are chosen for this comparison instead of one, making the estimation of the average number of games more robust and still keeping the portion of weekends and weekdays equal. The reason is that weekends generally have a slightly higher number of games played than the weekdays.

that lockdowns have required people to reduce any physical social contact to an absolute minimum. People, therefore, spent much more time at home on the internet and, in order not to lose contact with friends and loved ones altogether, often held video chats. Such a video call could include playing a simple and accessible multiplayer online game like *Jméno, město*. Although the game is playable on paper, it is unlikely that such a dramatic increase in the number of games is due to players of the paper version moving to the online version. These are most likely brand new players who were interested in doing an activity that they could do online in groups.

In order for there to be an increase in the number of games of the kind that took place during the 2020 pandemic at all, such a game needs to be already known to a considerable number of players. As shown in the graph, all of the mentioned language versions of the game had a stable player base before the pandemic began. As a result, the popularity of the game was able to experience such growth. Then, what helped build these player bases?

Because *Jméno, město* is based on a globally popular board game, it can be assumed that people would search for an online version of the game. Since the board game has a different name in most languages, the online version of *Jméno, město* also has a tailored name and web address for every language version of the game, making it easily searchable. Most of them are also listed first in Google searches[3] ahead of similar games such as *StopotS*[4] and *Stadt Land Fluss*[5].

The game owes much of its rise in popularity to its high profile on YouTube. Popular YouTube videos can be traced back to a large number of significant jumps in popularity. It is important to note that YouTube videos may not have an impact purely in the days after their release. The YouTube algorithm sometimes starts to recommend watching an almost forgotten video to many people at once, making it popular years after the video's release. This effect may have occurred for some of the videos mentioned, causing an increase in the number of games at a seemingly unrelated time period. This effect could be analysed by checking the timeline of views of popular videos to see if the video gained an unusually high number of views some time after publication. Although YouTube keeps track of

[3] As of April 18, 2021
[4] https://stopots.com
[5] https://stadtlandfluss.cool

this statistic, it is not publicly available. Therefore an analysis of this effect is not feasible without gaining access to the accounts of individual creators, which is why any examination of this statistic could not be performed. All the following mentioned video view counts are correct as of April 18, 2021.

German was the first language to develop a large, stable fan base. By October 21, 2015, the number of games per day fluctuated steadily between 70 and 120. The day after came a turning point that increased these numbers dramatically, which can be clearly seen on the above graph. On this day, the German YouTuber PietSmiet[6] released the first[7] of a series of videos of him playing the German version of the game with his friends. This video has amassed a total of 564,642 views. The average number of games over the next three days climbed to 2,065, essentially a twenty-fold increase. The following days, the average per day stabilized at between 600 and 1000 games until the 2nd video[8] was released on November 7. This video became even more popular with 695,983 views and raised the average to 3,572 games per day for the next three days. This creator made a total of 13 videos[9] in the series by the end of 2016, some of them already deleted, with each having at least 200,000 to 300,000 views.

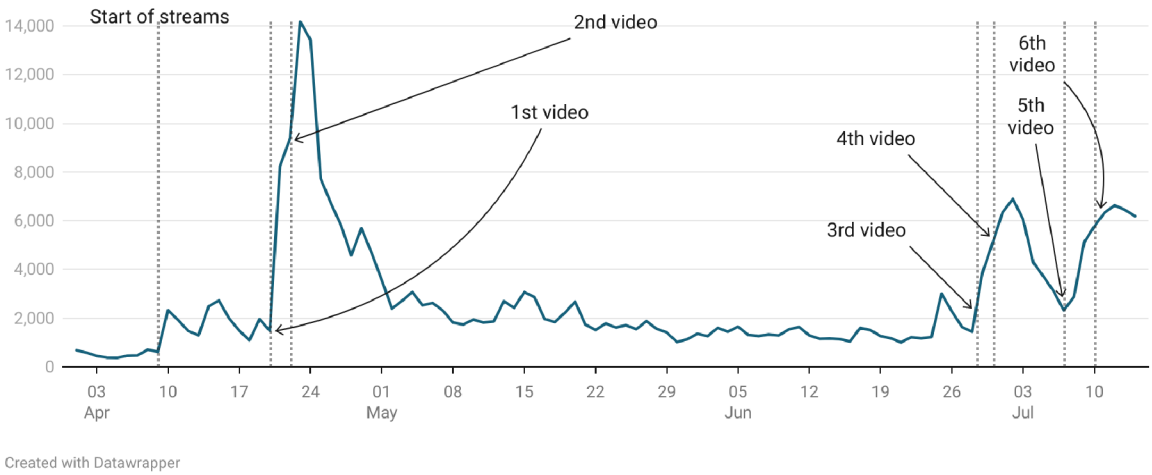

Figure 5.2: The impact of Gronkh's streams and videos on the number of games per day in German in 2016.

Other YouTube creators have also picked up on this successful concept. YouTuber Gronkh[10] released the first two of his series of videos from the game on April 21[11] and 23[12], 2016. The videos were a compilation of the best moments from playing the game, which they have streamed with up to 26,000 live viewers[13] on Twitch in the previous days. Figure 5.2 illustrates the number of games in German between April 1 and July 14, 2016. Before these streams, the average ranged from 400 to 700 games per day. In the period after the

[6] https://www.youtube.com/channel/UCqwGaUvq_l0RKszeHhZ51eA
[7] https://www.youtube.com/watch?v=NJYnkkZr9Gw
[8] https://www.youtube.com/watch?v=V0YKiM6DQuo
[9] https://www.youtube.com/playlist?list=PL5JK9SjdCJp8XuIuGfShrpKgbNqIMZcmV
[10] https://www.youtube.com/channel/UCYJ61XIK64sp6ZFFS8sctxw
[11] https://www.youtube.com/watch?v=iCwyRaTWD8o
[12] https://www.youtube.com/watch?v=yTrRl35Ra_E
[13] https://sullygnome.com/channel/gronkh/2016april

streams began and before the videos were released, it climbed to 1,500 to 2,000 per day. After the release of the videos, there were up to 14,000 games per day played during peak periods. As in the previous case, this number gradually decreased to 1500 per day until June 29, 2016, the release of the next video[14] in the series, which raised these figures again to the order of six thousand. However, all of these numbers cannot be attributed to Gronkh alone; they played the games with other, albeit less influential, personalities of the German internet, who probably also contributed to promoting the game through their streams and videos.

Another huge jump in the number of games can be seen at the turn of 2018 and 2019. This jump is probably due to videos by a YouTube creator called Arazhul[15]. They released 13 videos[16] between December 3, 2018 and April 30, 2019, which collectively have nearly 16 million views. The most significant influence on the number of games comes from the first video, which introduced the game to the largest number of viewers. With the following videos, which were released regularly a few days after each other during December 2018, February 2019 and April 2019, the majority of the audience was already familiar with the game, and therefore the big jump after the first video was not repeated.

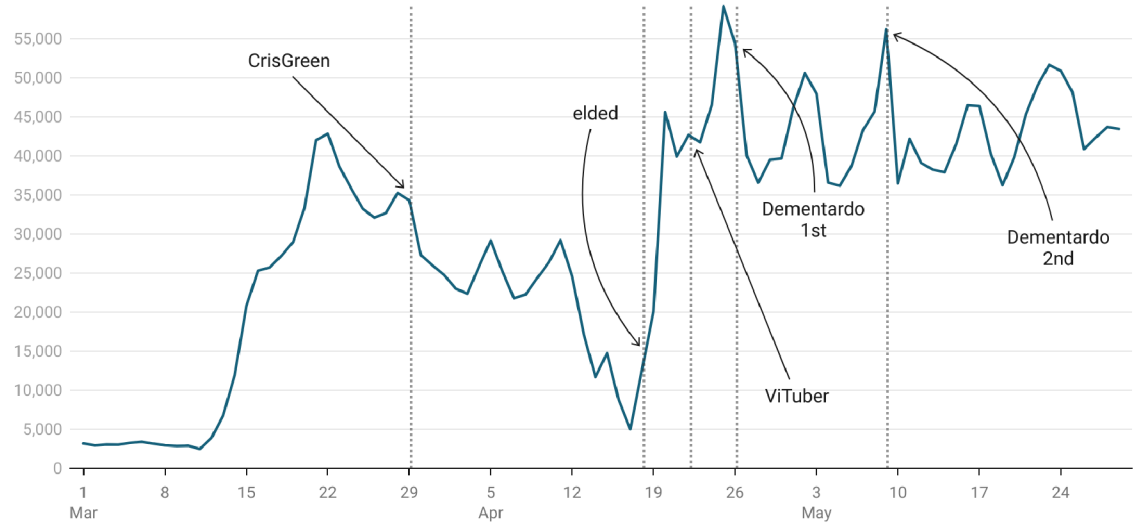

Figure 5.3: Increase of the number of games in Spanish during the spring of 2020 with the release dates of heavily viewed videos.

YouTube videos have also played a role during the first wave of the pandemic in the spring of 2020 when an enormous interest in the game can be observed among Spanish speakers. While the videos in German did not generate a significant amount of attention during this time as none of the prominent creators made any, the opposite happened in Spanish. During this period, two videos[17] were released by YouTuber Dementardo[18] with

[14] https://www.youtube.com/watch?v=g4lBPARzRkM
[15] https://www.youtube.com/channel/UCdo4IIRJlEaln7ZHZHzFNIw
[16] https://www.youtube.com/playlist?list=PL9GbGAd-gY1qjXIVJ6DbTOch-ATt5Dt9T
[17] https://www.youtube.com/watch?v=ztPrJOu-aWs, https://www.youtube.com/watch?v=i8c9-2Yx3qo
[18] https://www.youtube.com/channel/UC7LwVGGOGEhg1qD_wr-MdYg

a total of 5.3 million views, a video[19] by Elded[20] with 2.6 million views, a video[21] by CrisFreen[22] with 800,000 views and a video[23] by ViTuber[24] with 150,000 views. With these popular videos coming from different creators, the final reach was massive. Figure 5.3 displays the increase in the number of players in Spanish and the release dates of the videos. For the most part, video release dates do not directly correlate with the increase in player numbers. This lack of effect could be attributed to changes in the free movement restrictions or the fact that some viewers were familiar with the game already and played it independently of the videos. Around April 18, the game was experiencing server outages, which caused the low number of players at this time. Nevertheless, the videos most likely played a role in the increase in awareness of the game and, combined with the ongoing pandemic, caused the previously mentioned increase in games to nearly quadruple the pre-pandemic total.

Similar analyses of the number of games could be done in other languages with essentially the same results. The most significant influence on game numbers is rooted in the momentary publicity, and this correlates closely with the popularity of the game on Twitch and, most importantly, YouTube.

5.2 Geocultural Analysis of Answers and Categories

This section covers the geocultural analysis of answers and categories. This analysis is performed over answers within the same language. English was chosen as the language to be analysed. The reasons for this are that it is the thesis' language and the 4th most played language version. The relation containing answers in English has a total of 42,783,700 records, which is a sufficiently large sample to perform the analysis successfully.

The analysis compares the popularity of answers in different countries. The US, the UK, Australia and Canada (ordered by the amount of records) were selected for the analysis as English-speaking countries with the highest number of responses. Only 400,000 responses can be attributed to the fifth country in the ranking, New Zealand, which is one-eighth the number of responses from Canada. This number was considered insufficient, and thus only the responses from the four countries mentioned above are included in the analysis.

During the process of analysis, responses are compared on a percentage basis. This percentage is calculated as the number of occurrences of a response divided by the total number of responses in the same country for the same category and initial letter.

Nowadays, the world population is already highly globalised and cultural differences are gradually disappearing. Looking at the four countries mentioned above, all of which are highly developed, it can be said that they already share quite similar cultures. When comparing the responses from these countries, this fact can be clearly observed. The most frequent answers tend to be very similar for almost all categories in each of the countries.

Analyses of responses were conducted over dozens of categories, Table 5.2 presents the 30 most played, but in most of them, no noteworthy differences in responses from different countries were found. The analysis instead focuses on the categories where some differences can be identified.

[19] https://www.youtube.com/watch?v=Zgzx8lB8kas
[20] https://www.youtube.com/channel/UCejggbBHSAIx8wvnc4x7HSg
[21] https://www.youtube.com/watch?v=bKwmsekQ5EQ
[22] https://www.youtube.com/channel/UC7qdNWFCD3lo8toe0TsaeUw
[23] https://www.youtube.com/watch?v=h08UceYggIo
[24] https://www.youtube.com/channel/UCUEMCHt9VVY7qclA5yb23xg

1.	Animals	16.	Ice Cream Flavors
2.	Food	17.	Fruit
3.	Girl's name	18.	Reasons To Quit Your Job
4.	Country	19.	Song titles
5.	Color	20.	Things You Do On A Date
6.	Boy's name	21.	Things you Drink
7.	Movie	22.	Band or Artist
8.	Four letter word	23.	Sport
9.	Body part	24.	Things with tails
10.	TV show	25.	Superhero name
11.	Fruit or vegetable	26.	things you did today
12.	Thing you find in your pockets	27.	Clothing
13.	Pizza Toppings	28.	Games
14.	Things you see in the zoo	29.	Things you do before sleep
15.	excuses for being late	30.	Type of drink

Table 5.2: Thirty most played categories in English.

The more significant differences in the most popular answers are usually either different terms for the same answer, for example "crisps" in British and "chips" in American English, or answers that relate much more to one country than the others. The latter usually corresponds to answers that are commonly known across all countries, but only in some are they the first choice. As an example, while the most popular answers for the "Country" category starting with "A" are either "Argentina" or "America" in the US, the UK and Canada, in Australia, however, the clear winner is obviously the answer "Australia". Most interesting for this analysis, though, are the cases where one of the answers is massively popular in one country but wholly omitted in the rest. Such answers can be confidently considered to be a matter of cultural difference.

Given the categories, the answers are always strongly connected to the ordinary realities of life, that is, to the closest things to a person's country or culture. Thus, popular answers may describe things that are utterly alien to a person from another part of the world. Such a case can be well observed on the "Animal" category, which is the most played category in English. Of the four countries selected, two are in North America, one is in Europe, and one in Oceania. It can be assumed that popular answers would vary considerably depending on the continent, especially in the case of Australia, which is known for a unique fauna not found anywhere else in the world.

Figure 5.4 shows the answers to the "Animal" category that are significantly more frequent in Australia than elsewhere. All of them are animals living in Australia while being extremely rare or absent in the rest of the countries. Some of those species are even endemic to Australia. This fact alone does not warrant such massive differences in popularity; for example, koala, kangaroo, and emu are all frequently answered worldwide. What causes this difference is that these animals, although familiar to Australians, are not well known in other places. The echidna has even appeared on the Australian five-cent coin since 1966[4], which could be the reason of its immense popularity. These responses clearly show the diversity and uniqueness of animal life in Australia.

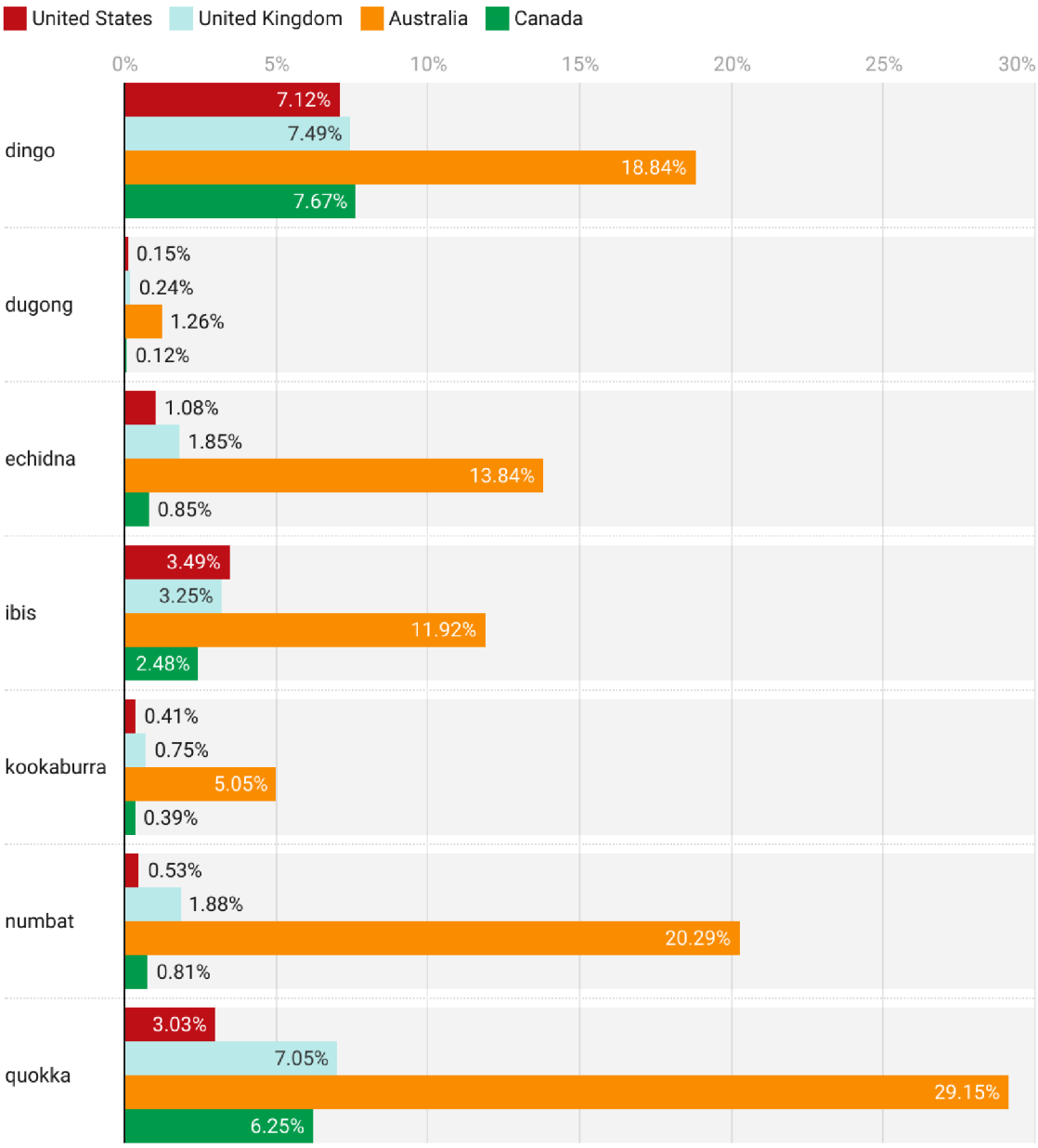

Figure 5.4: Answers for the category "Animal" significantly more prevalent in Australia than in the rest of the countries. Values are a percentage of the number of responses to the total number of responses to the same initial letter.

Similarly, the opposite effect can also be observed, as displayed by the distribution of these responses in Figure 5.5. It is possible to find out what animals live predominantly in North America by investigating responses more often coming from the US or Canada. Such responses include the alligator, cougar, jackrabbit, raccoon and ferret. The newt, which is common in North America and Europe, accounts for half the percentage of responses in Australia than the UK. However, it should be noted that the difference would have probably been slightly narrower was it not for the high frequency of the answer "numbat" in Australia. Further examples of animals dominant in Europe, and therefore the responses

more frequently originating from the UK, are the vole and the vixen, which is a term for a female fox.

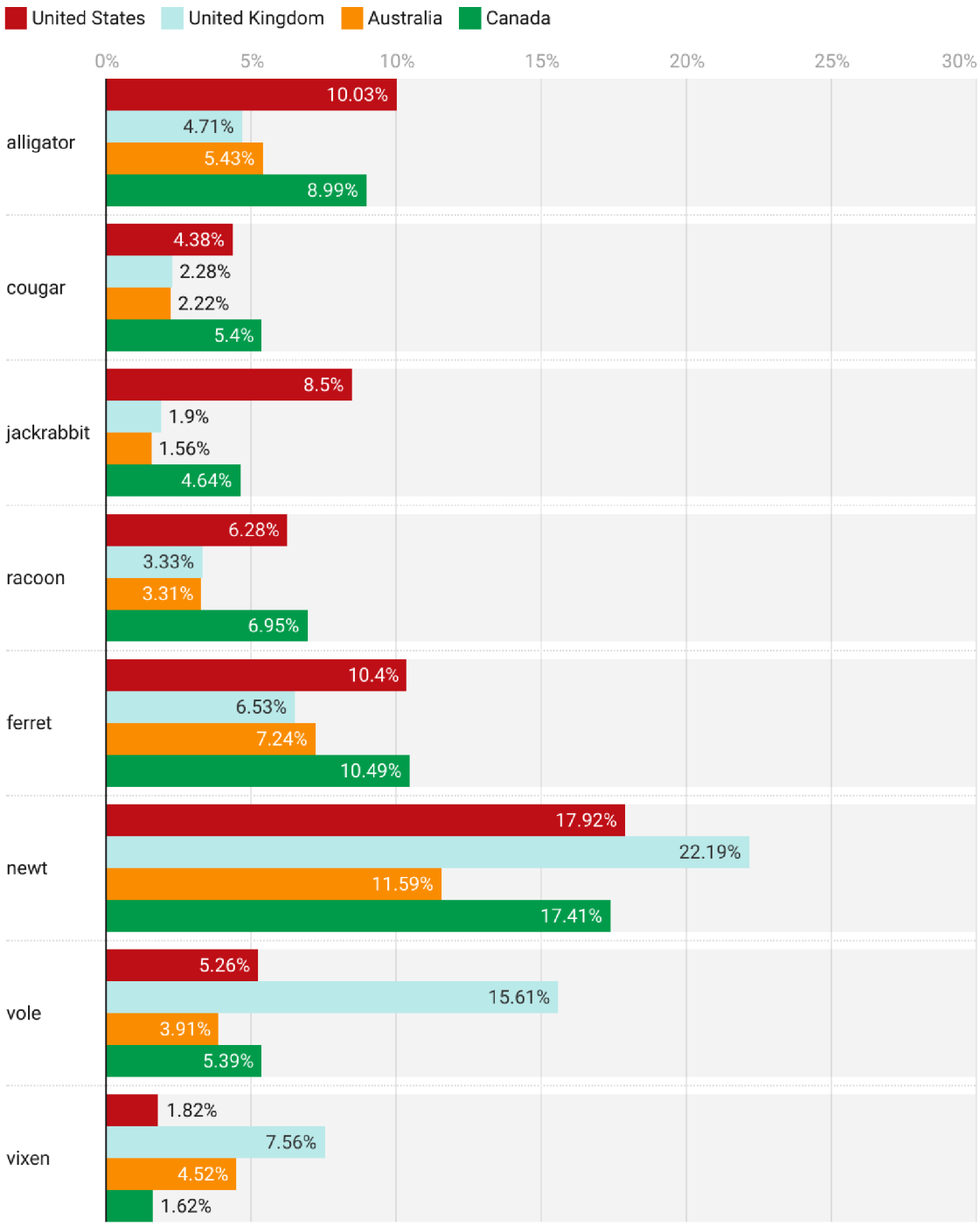

Figure 5.5: Answers for the category "Animal" significantly more prevalent in the North American countries or the UK than in the rest. Values are a percentage of the number of responses to the total number of responses to the same initial letter.

Another category on which differences in culture can be observed is the category "Sport". As Figure 5.6 illustrates, sport is an area of culture that is often vastly different in various countries. The high frequency of answers "equestrian" and "polo" in the UK and, to an extend, Australia indicates a firmly rooted tradition of horsemanship. This assumption is also supported in the "Hobby" category, where "horse riding" is a popular answer in these two countries. The high number of the answer "darts", on the other hand, could be in turn indicative of a modern trend, i.e. a sport that has experienced remarkable commercial growth in the UK[25] in recent years.

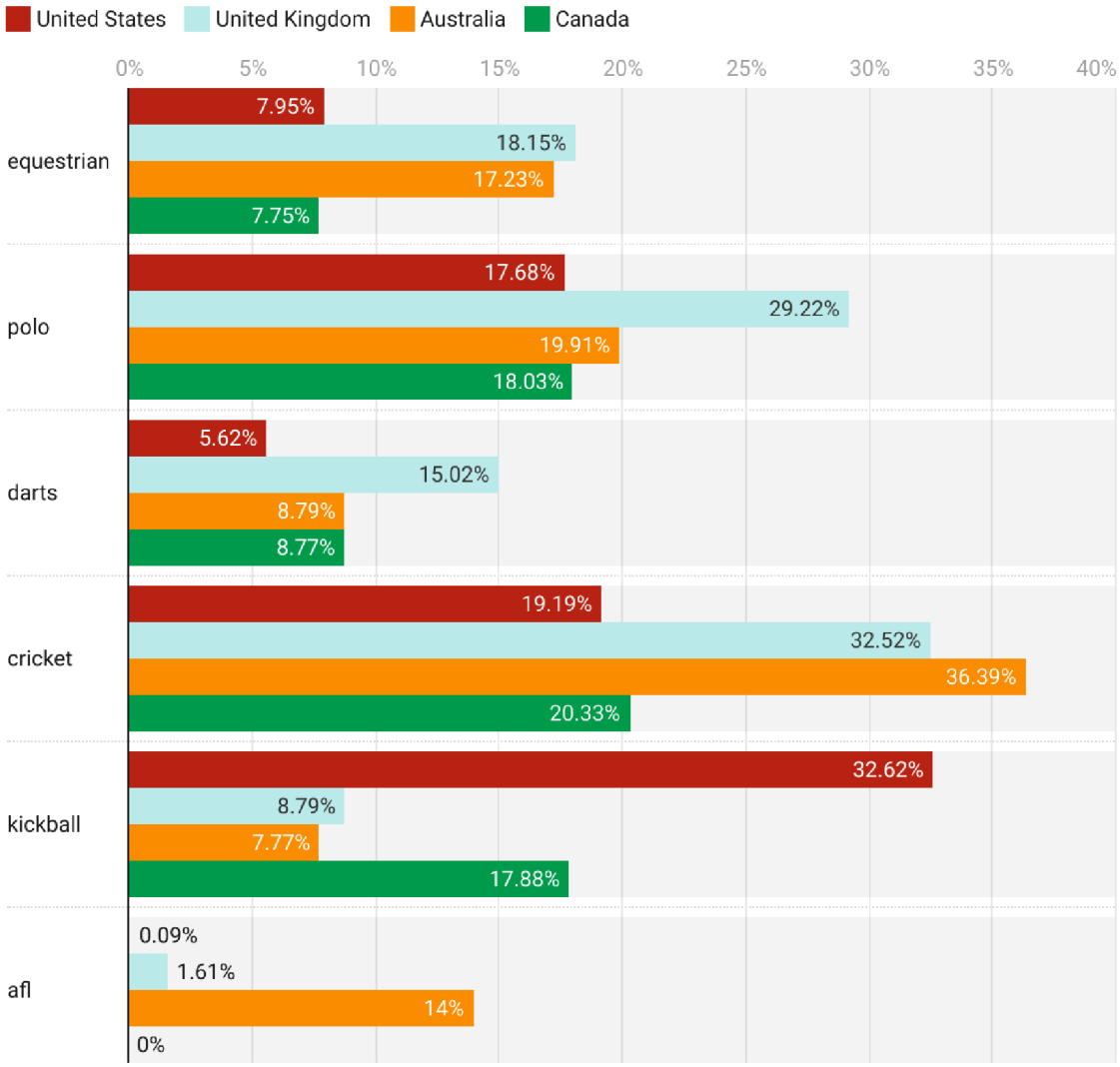

Figure 5.6: Answers for the category "Sport" significantly more prevalent in some countries. Values are a percentage of the number of responses to the total number of responses to the same initial letter.

[25] https://dartsnews.com/news/three-reasons-why-the-popularity-of-darts-skyrocketed

Sport	United States	United Kingdom	Australia	Canada
cricket	2	11	17	2
kickball	100	5	8	16
AFL	< 1	< 1	100	< 1

Table 5.3: Google Trends popularity scores calculated from searches for the sports in question over the period from December 14, 2014 to September 14, 2020.

The remaining sports are compared to Google Trends[26] search popularity scores. Table 5.3 presents the scores for each country and sport. Scores are calculated on a scale of 0 to 100, with a score of 100 representing the location with the highest popularity relative to all searches from that location. A score of 50 represents where the term has half the popularity. A higher score indicates a larger proportion of all searches, not a higher absolute number of searches. The time scale chosen includes the total time range of responses in English within the provided copy of the database.

As Google Trends statistics for cricket[27] show, it is searched the most in South Asia, the Caribbean islands, the Middle East, South Africa, England and Australia. When comparing the values in the figure with the Google Trends score, it can be noticed that the percentages of answer distribution approximately correspond to the score. Cricket is a slightly more popular answer in Australia than in the UK, while the US and Canada, where the popularity is similar, are well behind.

By contrast, kickball, which is similar in rules to baseball, is the most searched in the US, according to Google Trends scores[28], about six times less searched for in Canada and significantly less, but similarly frequently, in the UK and Australia. Similar proportions can be seen in the game's answers, where kickball is also the most popular in the US, less so in Canada, and the least, but similarly frequently answered, in the UK and Australia.

The AFL (Australian Football League) is the top competition of Australian rules football, which is the most attended sport in Australia[16]. Google Trends statistics[29] show that this term is searched for almost exclusively in Australia. Statistics from the gaming database confirm that the sport is popular almost exclusively in Australia, but there is some awareness of it in the UK.

The conclusion of this analysis is that although the cultures of the selected countries are similar, it is still possible to find clear differences between them even in the database of the *Jméno, město* game.

5.3 Answer Popularity Trends Across Time

This section examines trends in the popularity of answers over time. The time distribution of the answers is collected from all the game languages and compared to the global Google Trends search statistics. The number of responses for each day is calculated relative to the total number of responses on that day to represent a value unaffected by the number of players. Since Google Trends reports search popularity scores on a scale of 0 to 100, answer ratios are normalized to the same scale.

[26] https://trends.google.com/trends
[27] https://trends.google.com/trends/explore?date=2014-12-14%202020-09-14&q=%2Fm%2F09xp_
[28] https://trends.google.com/trends/explore?date=2015-12-14%202020-09-14&q=kickball
[29] https://trends.google.com/trends/explore?date=2015-12-14%202020-09-14&q=%2Fm%2F0ckh09

Different trends are analyzed. First, a data collection of trends in the popularity of 2020-pandemic-related terms and a comparison of these trends is performed. Figure 5.7 illustrates the comparison between the search[30] and the game answers occurrences of the term "covid". As the figure shows, while searching the term "covid" was less frequent since mid-March, the occurrence had an upward trend overall in the game. The reasons for the opposite trends can only be guessed. It is possible that at the beginning of the first wave of the pandemic, people searched more for information about the new disease, which then became common knowledge and the need to search for the term became less necessary. At the same time, the topic was still too new and stressful to be mentioned often by players. Over time, the novelty of the topic wore off, it became common, and, perhaps because of this, players were more inclined to use these term in the game. Another possible explanation for the increased occurrence of the term in the game could be a popularization of a category where the term is more likely to be brought up as an answer. This conjecture could be confirmed by finding a similar trend for another term with the same meaning.

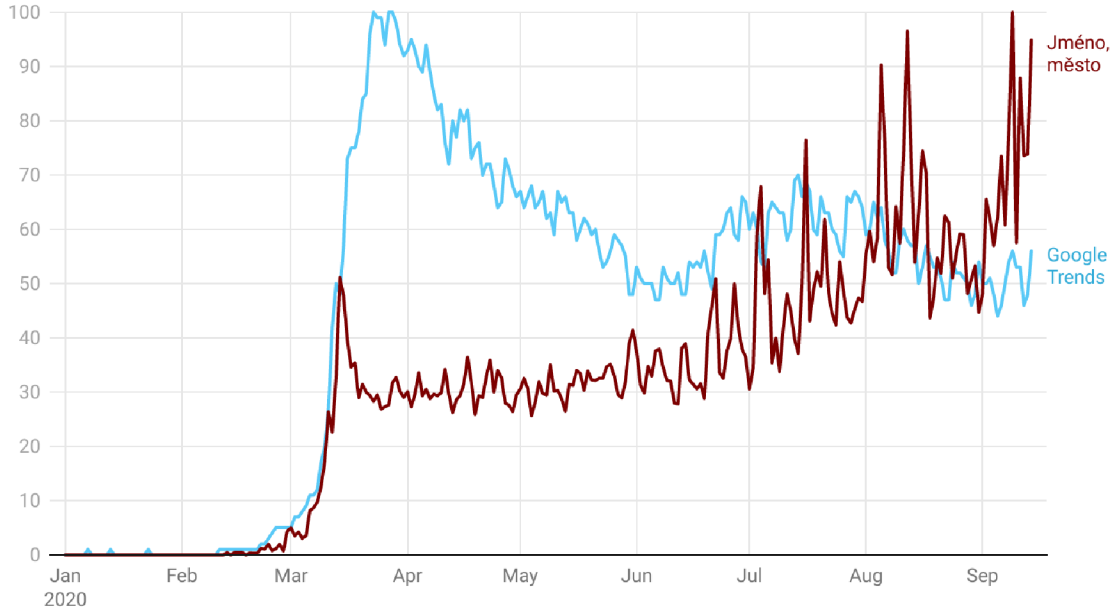

Figure 5.7: Worldwide comparison of the term "covid" in the Google Trends score and the occurrence of that term to the total number of responses within a day in the game database, normalized to a scale of 0 to 100, from January 1, 2020 to September 14, 2020.

Figure 5.8 illustrates the same comparison between search[31] and play occurrences as in Figure 5.7, this time for the term "corona". This term holds the same meaning as the term "covid" and even starts with the same letter, thereby avoiding the influence of other responses. As the graph shows, the trends of occurrences in game and search are similar. Thus, the influence of a newly popular category on the increased occurrence of the term "covid" in the game can be dismissed.

[30] https://trends.google.com/trends/explore?date=2020-01-01%202020-09-14&q=covid
[31] https://trends.google.com/trends/explore?date=2020-01-01%202020-09-14&q=corona

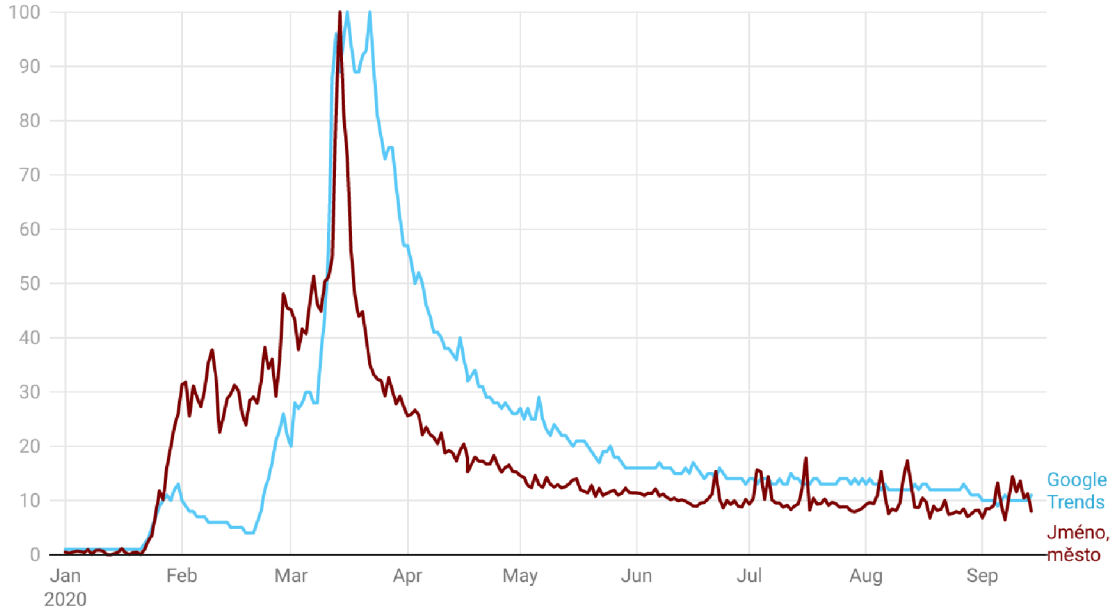

Figure 5.8: Worldwide comparison of the term “corona” in the Google Trends score and the occurrence of that term to the total number of responses within a day in the game database, normalized to a scale of 0 to 100, from January 1, 2020 to September 14, 2020.

In order for trend analysis of responses to be safely applicable over any expression, it must be possible to extract meaningful information from the database consistently. Previous analysis of pandemic-related terms has withstood a comparison with Google Trends. As an additional example, an analysis of the popularity of the app TikTok was conducted. The app was released in September 2017, which is within the time frame of the game database. Figure 5.9 shows the trend on both platforms, which, as one can observe in the graph, is very similar.

It can be said that some terms are as popular in the game as they are in searches in Google. Thus, such an analysis of the game database could give meaningful results when performed for other terms.

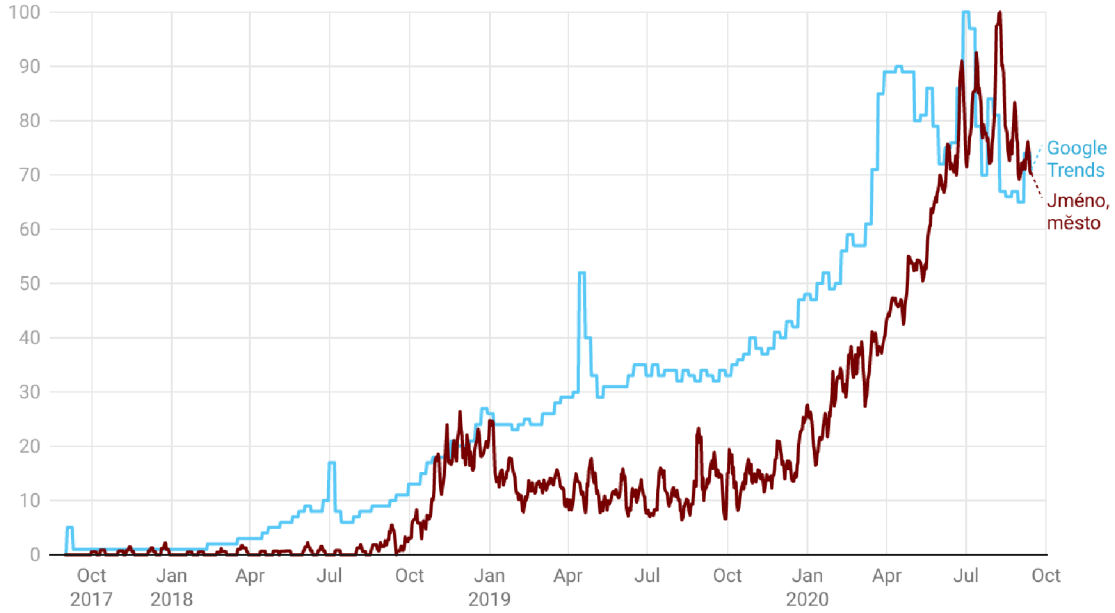

Figure 5.9: Worldwide comparison of the TikTok application's Google Trends[33]score and the occurrence of the terms "tik tok" and "tiktok" combined to the total number of responses within a day in the game database, with a 7-day moving average applied, then normalized to a scale of 0 to 100. The moving average is applied because Google Trends only reports the index value once a week for long time samples, and it also makes the graph more readable. Values from September 1, 2017 (application launch) to September 14, 2020.

5.4 Cross-lingual Analysis of Answers and Categories

This section aims to compare the answers of players in different languages. Although this type of analysis has many limitations, as explained in Section 4.3, which make it difficult to perform such an analysis in depth, it is at least, under certain conditions, feasible in a superficial way. Usually, the number of submissions, even when approximated to percentages, cannot be directly compared across languages.

The languages selected for this analysis are Czech and Slovak. They are very similar grammatically, which increases the accuracy of the results. The analysis is intended to confirm or refute the assumption that the languages and cultures in the Czech Republic and Slovakia are similar and very closely related. Answers in identically given categories in both languages are compared. Unfortunately, there are not many categories usable for this analysis, as only a few categories in Slovak contain a sufficient number of responses. A further issue with the analysis of these two languages is that it is not possible to completely filter out responses from people of non-Czech nationalities among the responses in Czech. Some players from Slovakia probably live in the Czech Republic and play the game in the Czech language version. Therefore, their answers appear in the game database in the same

[33] https://trends.google.com/trends/explore?date=2017-09-01%202020-09-14&q=%2Fg%2F11f555cn81

Czech	Slovak
ananas	uhorka
ementál	jablko
guláš	egreš
chleba	nanuk
nanuk	ryba
cibule	halušky
jablko	ananas
nudle	syr
zelí	egres
dort	jahoda
sýr	zemiaky
fazole	lievance
žemlovka	cesnak
citron	citron
halušky	palacinky
datle	i
brambory	rizoto
okurka	guláš
maso	slanina
jitrnice	ryža

Table 5.4: The 20 most frequent responses for the categories corresponding to the category "Food" in Czech and Slovak.

Czech	Slovak
elektrikář	elektrikar
hasič	elektrikár
doktor	maliar
malíř	doktor
zedník	zubar
lékař	lekar
zubař	automechanik
chemik	zubár
instalatér	traktorista
automechanik	lekár
opravář	rybar
gynekolog	murár
dělník	murar
rybář	rybár
kuchař	j
chirurg	opravar
traktorista	profesor
astronaut	policajt
celník	barman
inženýr	robotnik

Table 5.5: The 20 most frequent responses for the category "Job" in Czech and Slovak.

way as answers from Czechs living in the Czech Republic. To a lesser extent, this problem is also present in the Slovak answers with answers from Czech players.

The analysis was first performed over the "Food" category equivalents. Table 5.4 contains the 20 most frequent answers in both languages as a demonstration of the typical answers. All of the answers for the category in one language were searched for in the other both directly and translated, with or without diacritics. By examining these answers, it was found that the Czech terms "bramborák" and "dalamánek" are present in Slovak responses at most in single-digit figures. This amount of responses is, in a total of 1225 responses for the category, negligible. Similarly, the term "granadír" common in the Slovak answers is almost absent from the Czech ones. It is again found in single-digit figures and, with about 100 times more total responses than in Slovak, this number is also negligible.

Next, an analysis was made of the anwers corresponding to the category "Job", of which the 20 most popular responses in both languages are shown in Table 5.5. No noteworthy differences between the responses in the two languages were found.

Some differences can be observed in the "YouTuber" category. This category is not affected by the aforementioned problems of this method, such as the effect of translations on the frequency of responses. The names of the creators are the same for all languages, so the popularity of the responses can be directly compared by the percentage of responses to the same initial letter. Tables 5.6 shows the most frequent answers in Czech and 5.7 in Slovak. At first glance, it can be noticed that 13 out of the 20 responses are in the top

kovy	natyla
tary	vadak
gogo	gogomantv
hoggy	fallenka
datel	smusa
baxtrix	ikaro
jirka král	rady
zachy	ati
pedro	lucypug
ment	ondra vlček

Table 5.6: The 20 most frequent responses for the "YouTuber" category in Czech.

gogo	selassie
moma	fifqo
gogomantv	datel
tary	jirka kral
hoggy	pppeter
lucypug	vadak
zika	ment
baxtrix	pedro
kovy	pewdiepie
asimister	smusa

Table 5.7: The 20 most frequent responses for the "YouTuber" category in Slovak.

20 in both languages, which already indicates some similarities as well as differences. All the seven remaining answers in Czech are also quite frequent in Slovak. However, of the remaining Slovak answers, four of them do not occur in the Czech answers almost at all.

Figure 5.10 takes a closer look at the most frequent answers in Czech and Slovak for a few selected initial letters. Each time, the first of the two creators of the same initial letter creates content in Czech, the second in Slovak. The differences in the popularity of Czech creators between the languages, although clearly visible, are usually not as pronounced as for Slovak creators. In several cases, the most popular answer of Slovak players is present in a very small quantity among Czech answers. On the other hand, nearly all of the most answered Czech creators, even those not listed in the figure, score very highly in Slovak, too.

From this finding, it can be concluded that although Slovak players know Czech YouTube creators fairly well, the same cannot always be said on the contrary. Generally speaking, it can be said that Slovaks are more aware of what is going on in the Czech Republic than Czechs are of what is going on in Slovakia. Nevertheless, it is usually difficult to find significant differences in the answers of Czechs and Slovaks. This analysis confirms that, as expected, the two countries and cultures are very similar and close to each other.

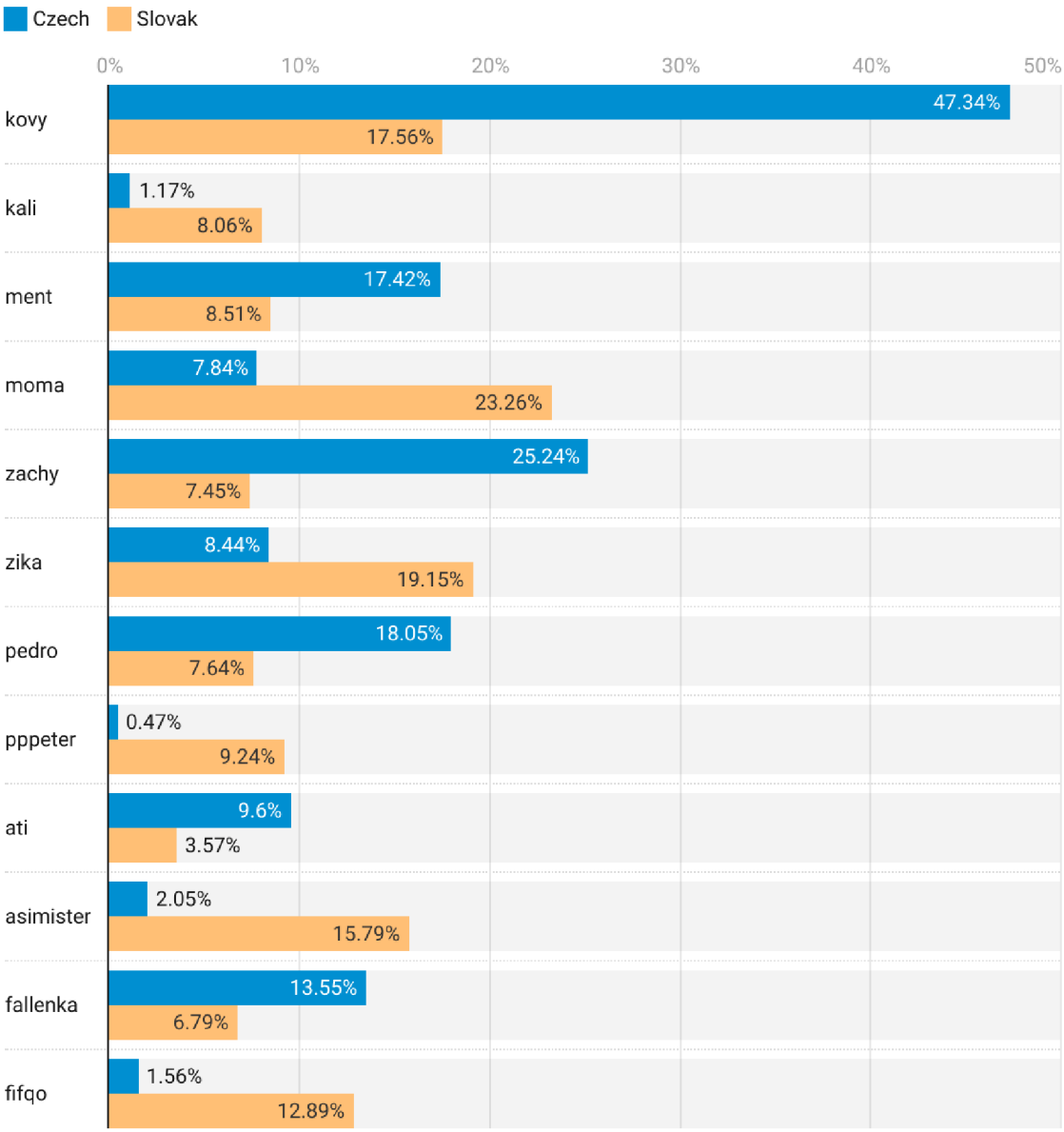

Figure 5.10: Comparison of some of the most popular answers in the category "YouTuber" in Czech and Slovak.

Chapter 6

Analysis Tool Development

The owners of the *Jméno, město* online game raised a request for a software tool that would allow them to visualise and analyse data of the game database more easily. This chapter explains the design and requirements of the tool, what technologies were used, the implementation, and the final product.

6.1 Application Analysis

The first step towards creating any application is the analysis of its requirements. The following development steps are fully dependent on the foundations laid by the analysis. It is therefore important that it is done carefully. This section focuses on the requirements, already existing solutions to these requirements and the functionality design of the application.

Requirements

The aim of the application is to create a simple-to-use software application that would be used for the analysis of answers to various categories on a given initial letter.

The application must allow the user to specify what answers they want to analyse and how. The two types of database selections the application must support are displaying the most popular answers and displaying answer popularity change over time. After the user creates their selection, the application must clearly show the results of the selection in the form of a graph.

For the selection of most popular answers, the user may specify:

- language of the answers
- category the answers belong to
- initial letter
- country of origin of the players
- number of answers to show

For the selection of answer popularity change over time, the user may specify:

- answer

- language of the answer
- category the answer belongs to
- initial letter
- country of origin of the players

Existing Solutions

There are no existing applications that would allow such a complex query creation without requiring the user to have deep knowledge of the game database. Additionally, there are no applications that would convert the results of such queries into a visually pleasing and ready-to-use graph. Currently, the only way of doing so is manually writing an SQL query and then feeding the results into a chart creator application. The proposed application aims to make this process considerable more straightforward and much faster by being specifically tailored to the game database, not requiring any understanding of the database structure or any code writing and automatically visualising results in a graph.

6.2 Design

This section examines the design use cases of the application. Determining and clearly stating what features the application will cover is an integral part of the application design.

Use cases

Use cases of the application are shown as a use case diagram in Figure 6.1.

The two main requirements described in Section 6.1 are portrayed on the diagram as "Get most popular answers" and "Get answer popularity over time". These selections allow the user to specify a category from which to get answers. Presuming the user is not familiar with the game or the game database, the user is not aware of which category name could be specified. To know which categories the user can select from, the use case "Get most popular categories" is needed. To further help the user create the most popular answer selection, the user can find how many answers are there to given criteria as described by the use case "Get total amount of answers".

Once the results of the selections are shown, the user can export the graph visualisation of the selection results as a PDF or a PNG file. The user can also export the raw results as a CSV file.

Before using the application at all, the user has to log in. The user can also log out. This is necessary because some of the data present in the database can be deemed as sensitive and also because using this application could give any player a significant advantage while playing the game.

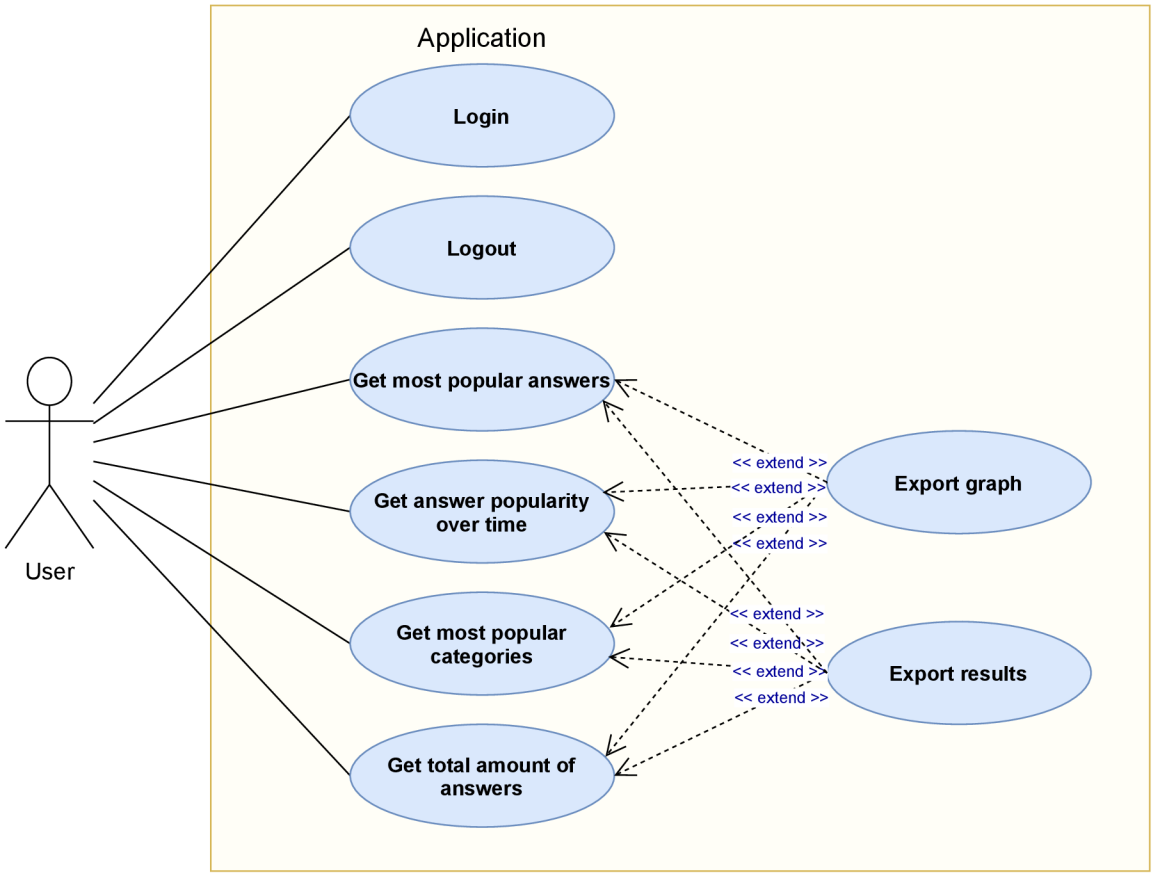

Figure 6.1: Diagram of the application's use cases.

Query selection specifications

This section details the different types of queries the user may create and the options they can specify for each type. Visual representation of the following settings is displayed in Section 6.3.

Most popular answers selection:

- language of the answers (one or all)
- multiple category names answers can belong to
- multiple countries players may be based in
- initial letter
- number of entries to show
- option to alter results to percentage, total counts are shown by default

Answer popularity over time selection:

- multiple character strings and options of “exact match”, “starts with”, “ends with” or “contains”
- language of the answers (one or all)
- multiple category names answers can belong to
- multiple countries players may be based in
- initial letter
- option to alter results to percentages, total counts are shown by default

Most popular categories selection:

- language of the categories (one or all)
- number of entries to show

Total amount of answers selection:

- language of the answers (one or all)
- multiple category names answers can belong to
- multiple countries players may be based in
- initial letter
- number of entries to show

6.3 Implementation

This section covers the process of development of the application based on the analysis presented in Section 6.1. The brief descriptions of the utilized technologies, application database, front-end and back-end are included in this section. The source code of the application is publicly available on GitHub[1].

Used Technologies

Following the analysis of requirements and design, the development of a web-based application was chosen. Web-based technologies provide a series of advantages over a standalone application. They are not reliant on a platform. They do not require installing additional software, only a standard web browser. Finally, they do not require a lot of computer power from the end-user because all performance-heavy logic is typically executed on the server. A slight disadvantage of a web-based application is the need to optimise the front-end of the application for different web-browsers and screen sizes.

This application was developed primarily for standard computer screen size and the Google Chrome browser, although also tested on Mozilla Firefox and Microsoft Edge.

[1] https://github.com/Vennny/IBT-app

PHP

PHP[15] is a popular open-source general-purpose object-oriented scripting language, which is particularly suitable for web application development. PHP code is run on the server, and the result of the executed PHP code is then sent to the client, for example, an HTML website. PHP is a dynamically typed language, meaning that the type of the variable changes according to the value that is stored in it[14]. Versions 5 and 7[10] introduced support for more strict typing, making the language essentially weakly typed if desired by the developer. In a professional environment, where applications are typically large-scaled, a PHP framework is often used instead of a plain PHP, which helps the development be much faster.

PHP is the main programming language of the back-end of this application. The application is programmed in PHP 8.0.1, which was the most recent stable version of PHP at the time of the start of the application development.

Laravel Framework

Laravel[5] is a free, open-source PHP framework designed for web application development. Laravel aims to make PHP-based web development easier and more approachable for developers of any level of experience. Laravel implements many of the features that are needed for most web applications, thus letting the developer focus on more specific work. Laravel is primarily based on the MVC (Model-View-Controller) architecture. A Model is the application data and data-related logic. A View is responsible for the presentation of application data. It represents all of the user interface logic of the application. In the case of Laravel, a View also handles the generation of HTML code. A Controller acts as an interface between the Model and the View. It processes the user's actions using the application's business logic, manipulates data using the Model and interacts with the View to provide the final output[9, 20]. Laravel was used for both the front-end and back-end development.

HTML

HTML[7] is the standard markup language used for creating web pages. By consisting of a series of elements, it describes the structure of a web page, which is later parsed by a web browser and displayed. HTML was utilized for the application's views.

CSS

CSS[3] is the standard style sheet language used for styling web pages. It is the code responsible for making the web page visually appealing. This is achieved by styling individual HTML elements or grouping elements by their tag name or an attribute (most of the time the `class` attribute), therefore applying the styling to multiple elements at once. CSS styles the HTML views of the application.

Bootstrap

Bootstrap[6] is a free, open-source CSS framework for building faster and easier responsive styling. Bootstrap also optionally contains JavaScript-based design templates. Boostrap comes with a large set of CSS rules, that can be applied to an HTML element using the

`class` attribute. In the proposed application, views were structured and styled with the help of Bootstrap.

JavaScript

JavaScript[1] is an interpreted or just-in-time compiled programming language. Javascript's most well-known usage is to program web pages' behaviour, but it is also widely used in non-browser environments. Javascript is a dynamically typed language and supports both a procedural and an object-oriented approach and both an imperative and a declarative programming style. Javascript is responsible for all of the front-end logic of this application.

Jquery

JQuery[8] is a free, open-source JavaScript library. It is designed to simplify HTML document traversal and manipulation, event handling, CSS animation and Ajax with an API that works across most browsers. It is the by far most commonly used JavaScript library[13].

Chart.js

Chart.js[2] is a free (available under the MIT license), open-source JavaScript library. It provides an easy way of data visualization. Chart.js supports eight different chart types: line, bar, radar, pie, polar, bubble, and scatter. In the proposed application, graphs containing the results of selections are created with chart.js.

Database Structure and Data

The database of the application is heavily based on the database copy provided for the analysis. Much of the information contained in the database is not relevant to the features of the application. The information that is relevant is stored in a structure that is not appropriate for the application. Leaving a copy of the database in its original state would lead to very time-consuming database queries. Therefore, it is necessary to create a copy of the original database that is tailored to the application. The disadvantage of this modification is that to keep the analysed data up-to-date, such modified copies have to be created regularly. Although this process is very time consuming, it dramatically speeds up and simplifies the analysis and is therefore worth carrying out.

The only relations needed for the four types of query selections specified in Section 6.2 are relations "category" and "word". The "category" relation remained untouched for the final application database, but the "word" relation proved to be, with the size of 127.5 GiB, too big for regular query selections. Such queries took up to tens of minutes of computer time to execute when run on a regularly powerful computer[2]. Therefore a decision was made to split the "word" relation into multiple smaller relations based on the answer's language. Due to the "word" relation containing answers by not only real players but also by bots, the relations were also filtered only to contain answers from real players.

[2]CPU: Intel® Core™ i7-4710HQ, RAM: 12 GB, SSD: Crucial MX500 1TB

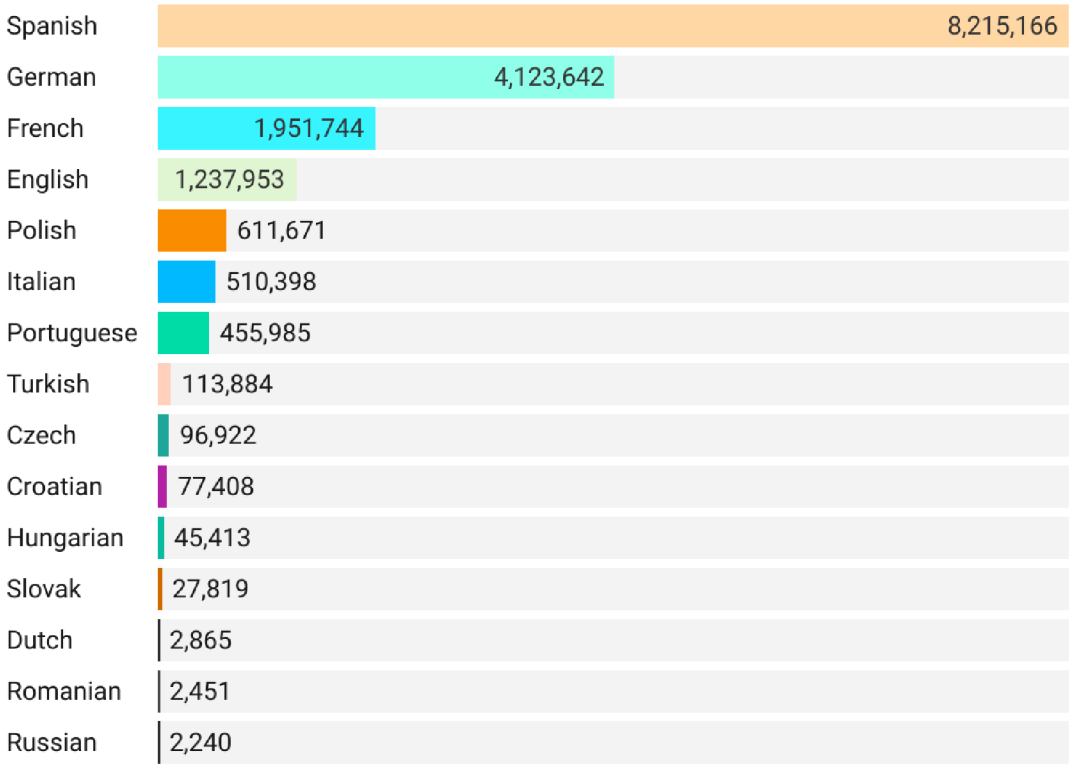

Figure 6.2: Top 15 languages sorted by the number of games played.

Based on the most played languages, shown in Figure 6.2, the relation "word" was split into ten different relations. These relations are the first seven of the most played languages (Spanish, German, French, English, Polish, Italian, Portuguese), native languages of the likely users of the application (Czech, Slovak), and the rest of the languages united into one relation. New relations were created using SQL commands for each language showed in Listing 6.1. New language-specific answer relations additionally include the identifier of a category, the initial letter and the country where the player was based. The last new relation, uniting all of the remaining languages, additionally includes the language code. It is also unnecessary to keep the information about the time of answer submission; only the date is kept. These modifications further speed up the execution. The script does not need to search for this often needed information in tables "category" and "game_player".

Index creation was considered and experimented with, but it did not improve the performance of database queries, quite the opposite.

```
CREATE TABLE word_cs AS(
    SELECT
        word.id AS id,
        word.value AS value,
        DATE(word.date_cr) AS date_cr,
        category.id AS id_category,
        round.starts_with AS letter,
        game_player.country_code AS country_code
    FROM
        word
    INNER JOIN
        category ON word.id_category = category.id AND
        category.id_lang = 'cs'
    INNER JOIN
        game_player ON word.id_game_player = game_player.id AND
        game_player.flag_robot = false
    INNER JOIN
        round ON word.id_round = round.id
)
```

Listing 6.1: Example of an SQL query used to create new answer relations based on the language.

Compared to the time consumption mentioned in the previous paragraph on the same computer specifications, this step significantly increased the speed of the queries. It reduces the amount of time needed for execution to seconds for languages with smaller data pools and a couple of minutes for languages with big data pools like Spanish and German.

The "lang" relation is also included in the application database structure. The languages are listed for the users to choose from when creating a selection.

To fulfil the needs of the application "login" use case, relations "users" and "migrations" were created using the Laravel command `php artisan migrate` later during the development process. Relation "users" stores information like username and password to successfully login the users eligible to use the applications. Relation "migrations" stores the dates of the creations of new migrations, which is required by the Laravel framework.

The final structure of the application database is shown in Figure 6.3.

Figure 6.3: Entity relationship diagram of the application database.

Database Connection

To establish a database connection, first, an Apache HTTP server has to be set up and started. The connection itself is defined in the `.env` file in the root of the Laravel application. A standard way of working with the database tables is through a model, which is then used by the Laravel Eloquent ORM[3]. However, most of the database queries build by the application are too complex for the Eloquent ORM approach. These queries are most efficient when working with the database directly and therefore are proccessed without a model. The only model created is the `user` model, which is used by the Laravel Authentication middleware[4].

User Interface

The directory tree of the application's user interface (UI) covered by this section is shown below.

```
/
├── public
│   ├── css
│   │   └── styles.css
│   └── js
│       ├── form.js
│       ├── graph.js
│       └── loading.js
└── resources
    └── views
        ├── layouts
        │   └── app.blade.php
        └── main
            ├── graph.blade.php
            └── index.blade.php
```

The core of the front-end is the layout view `app.blade.php` and the two main views, `index.blade.php` and `graph.blade.php`. These views are only accessible to the user once they are logged in. Classes and views that resolve logging in were generated using Laravel Authentication commands[5], slightly modified and will not be further described.

The `app.blade.php` file contains HTML code common to both views. This includes the complete `<head>` tag with, among other things, links to common JavaScript code and CSS stylesheets including Bootstrap, which is used for the graphical design. All of the styles that are not generated by a library are included in the file `styles.css`. The layout file also includes the navigation bar placed at the top of the page and handles the presentation of errors that may occur, typically errors from the server-side validation.

The `index.blade.php` view displays the query builder form. The form changes dynamically based on the user's actions using the JavaScript and JQuery code in file `form.js`. There are four base form structures. They correspond to the selection types presented in Section 6.2. An example of a filled-in form of the "Most popular answers" selection is

[3]More information at: https://laravel.com/docs/8.x/eloquent
[4]More information at: https://laravel.com/docs/8.x/authentication
[5]See footnote 4

depicted in Figure 6.4. Similarly, an example of the "Most popular categories" selection is depicted in Figure 6.5, an example of the "Answer popularity over time" selection in Figure 6.6, and an example of the "Total amount of answers" selection in Figure 6.7. When the user presses the "Confirm" button, request is sent to the Application Programming Interface (API). The application's API is described in detail later. Using the JavaScript code in the file `loading.js`, the view displays a loading icon over the page until the request gets resolved. This icon is created in CSS and was generated by loading.io[6].

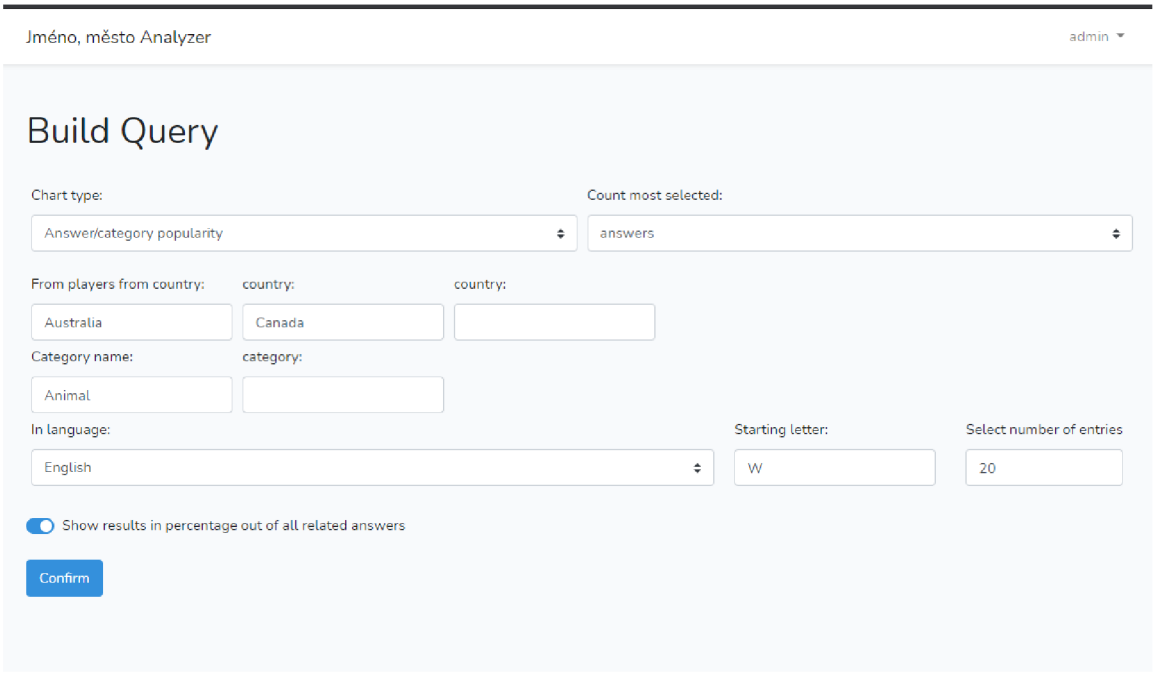

Figure 6.4: Filled in form example of the "Most popular answers" selection.

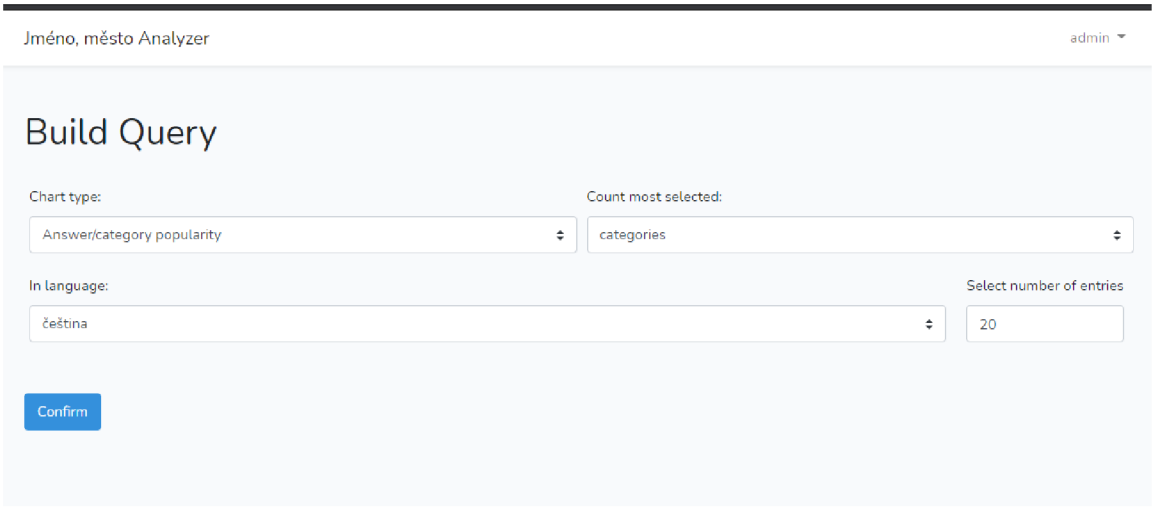

Figure 6.5: Filled in form example of the "Most popular categories" selection.

[6] https://loading.io/

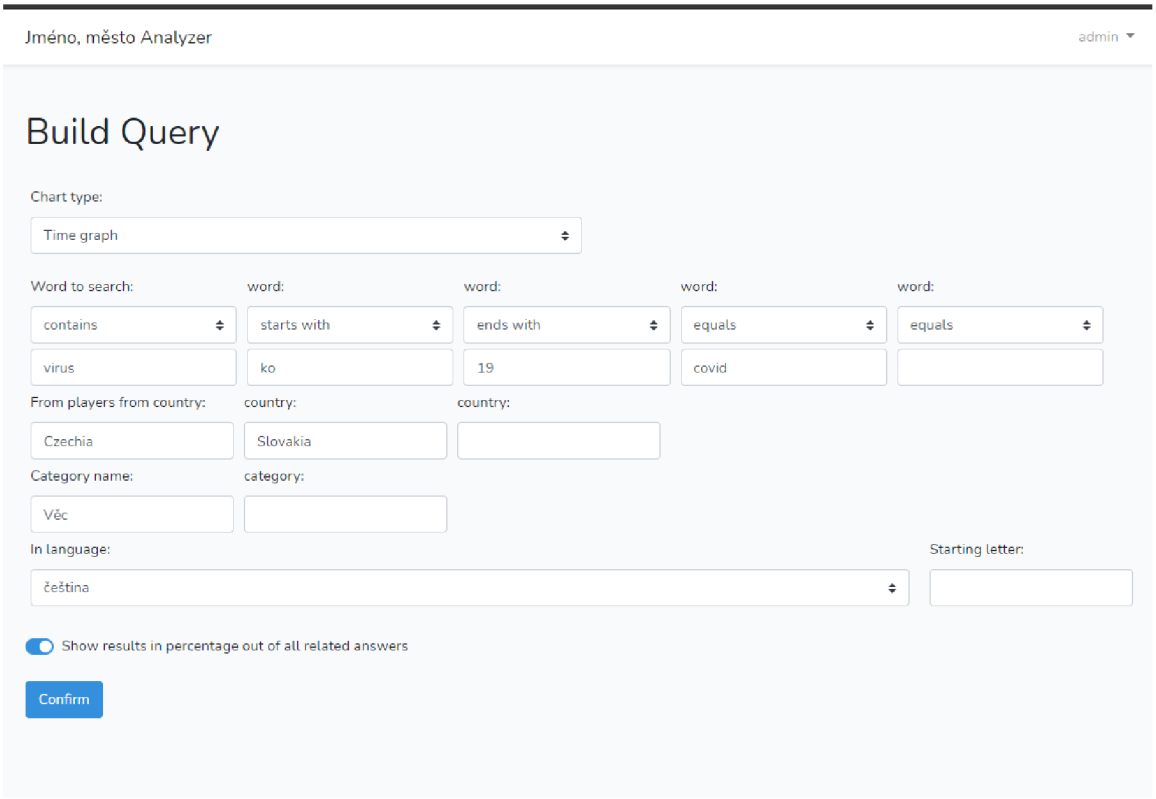

Figure 6.6: Filled in form example of the "Answer popularity over time" selection.

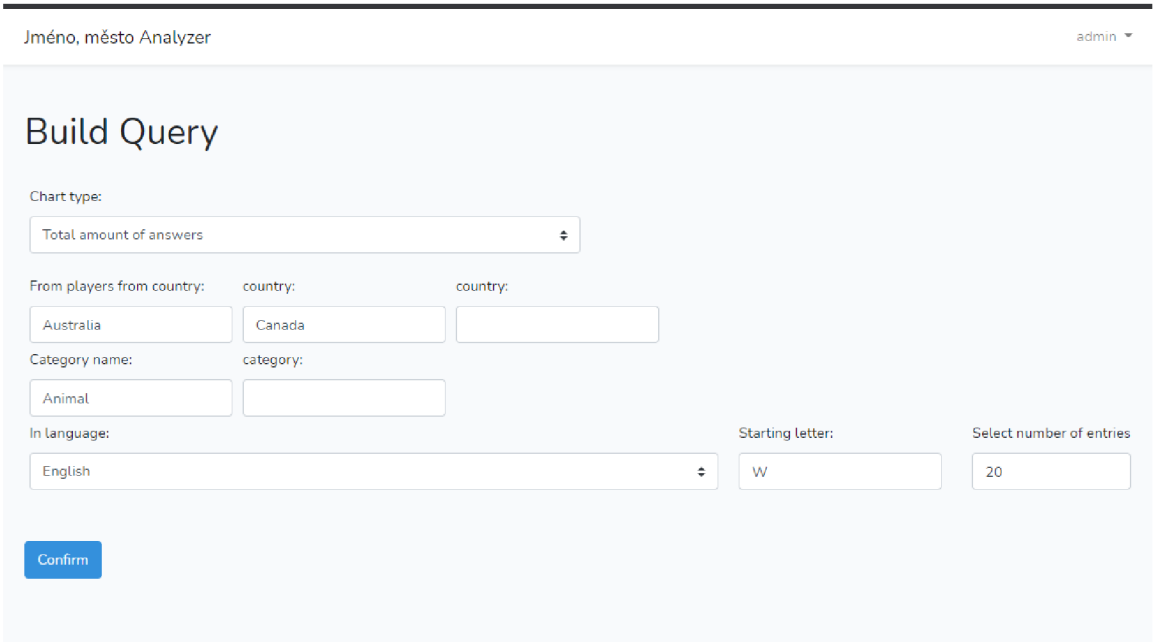

Figure 6.7: Filled in form example of the "Total amount of answers" selection.

The `graph.blade.php` view displays the results of the query returned from the API. All of the non-generated JavaScript code related to this view is located in the file `graph.js`. All

results are displayed in the form of a graph. The graphs are created using the JavaScript library chart.js. The "Most popular answers", "Most popular categories", and "Total amount of answers" selections results are presented by a horizontal bar graph, as demonstrated in Figure 6.8. The graph page allows the user to customise the graph by creating the implicit title or changing the labels of the axes. Graph values axis can also be modified to begin at zero. Font sizes of the title and the axes' labels are scalable using the range sliders under each input field.

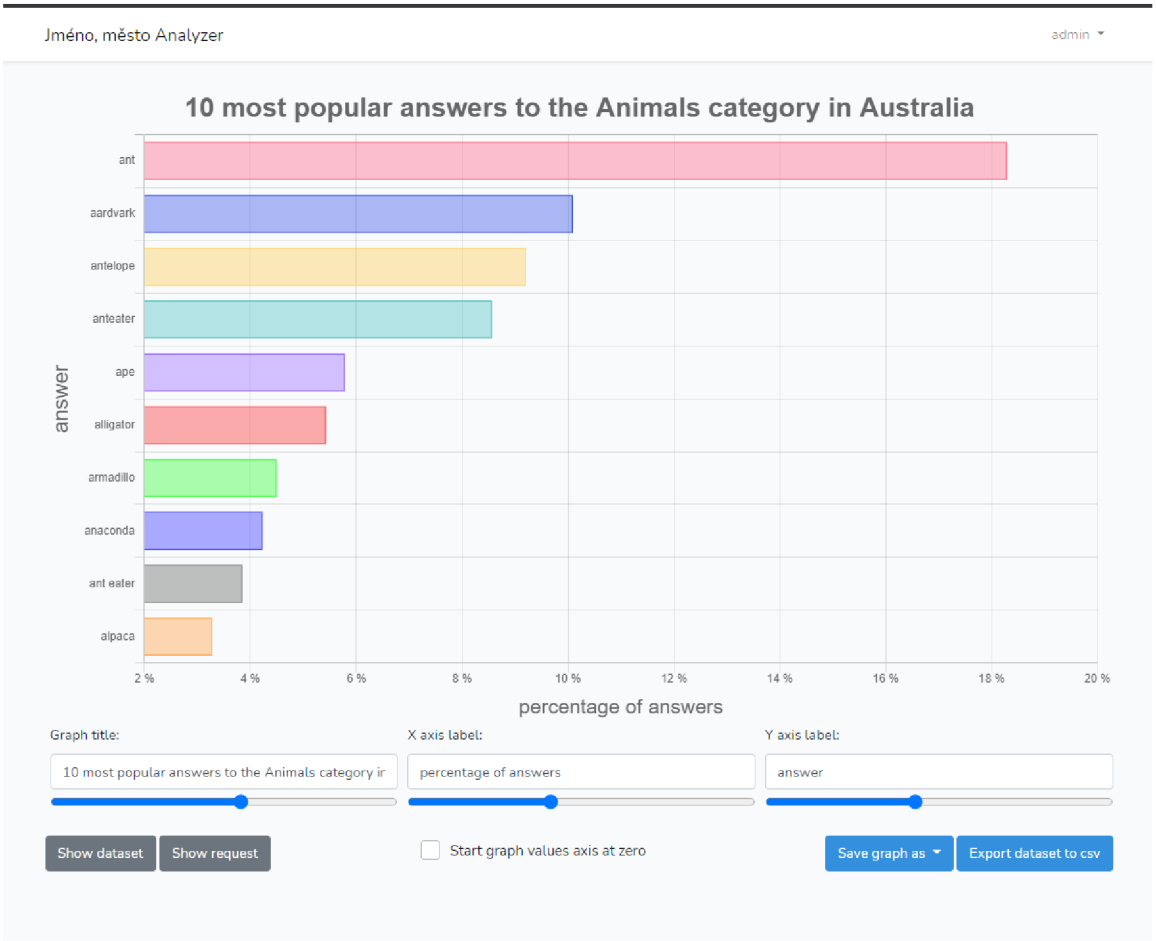

Figure 6.8: An example of a "Most popular answers" selection results page.

Figure 6.9 shows a dataset table containing data used in the graph and the options of graph export. This table can be displayed by clicking the "Show dataset" button. The dataset table data can be exported as a file in the CSV format using the button "Export dataset to csv". This export provides easy access to the data for further work and is implemented using the table2csv[7] library. The graph, including the title and the axes' labels, can be exported as a PNG image or a PDF file with the "Save graph as" button. The PDF export uses the jsPDF[8] library. In case the user forgets what the data represent, they can also display the request and the SQL query by clicking the button "Show request".

[7]More information at: https://github.com/OmbraDiFenice/table2csv
[8]More information at: https://github.com/MrRio/jsPDF

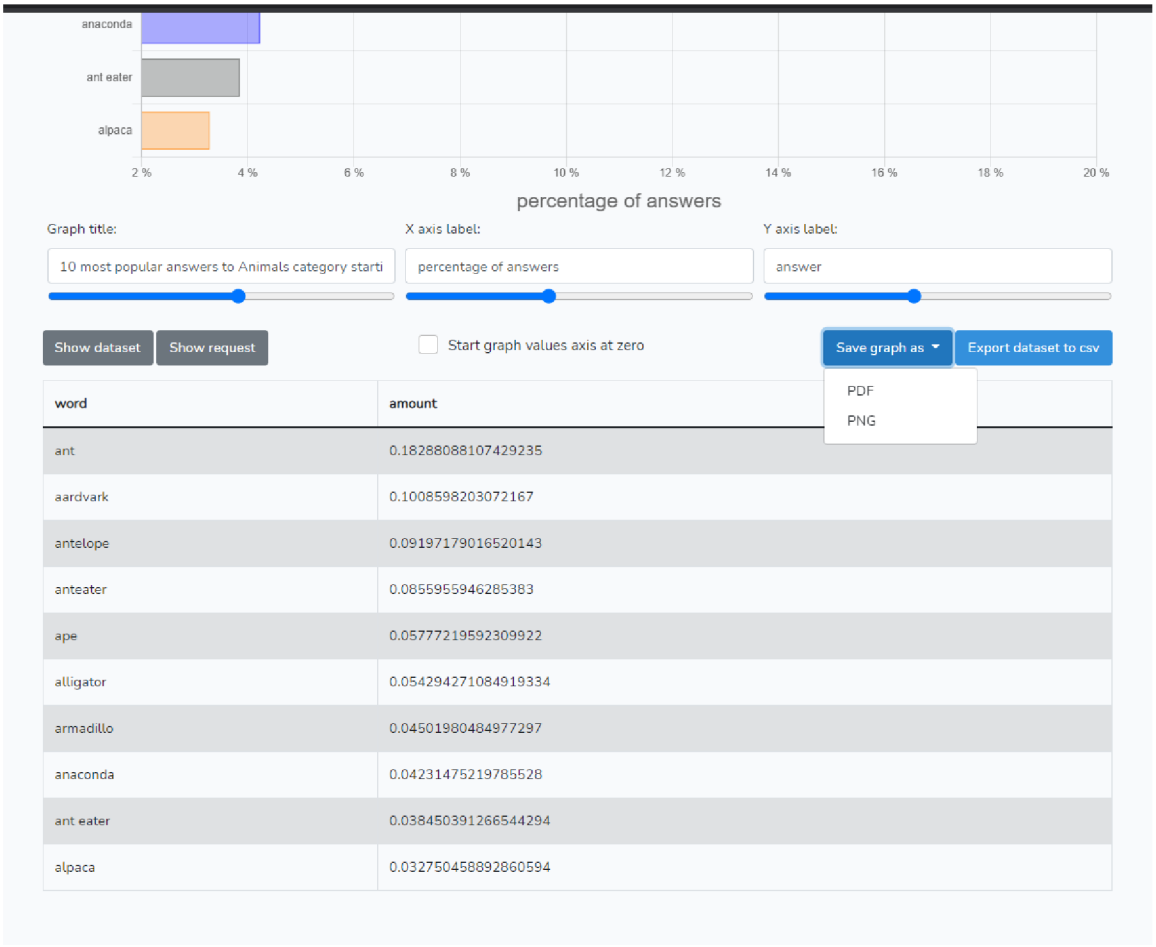

Figure 6.9: An example of the dataset table and graph export options on a selection results page.

Figure 6.10 illustrates the "Answer popularity over time" selection results page. The results of this selection are displayed in a line graph. The view provides additional customisation options for this type of graph. The user may adjust the starting and ending dates of the graph and apply a moving average[9] to the values.

[9]More information at: https://www.investopedia.com/terms/m/movingaverage.asp

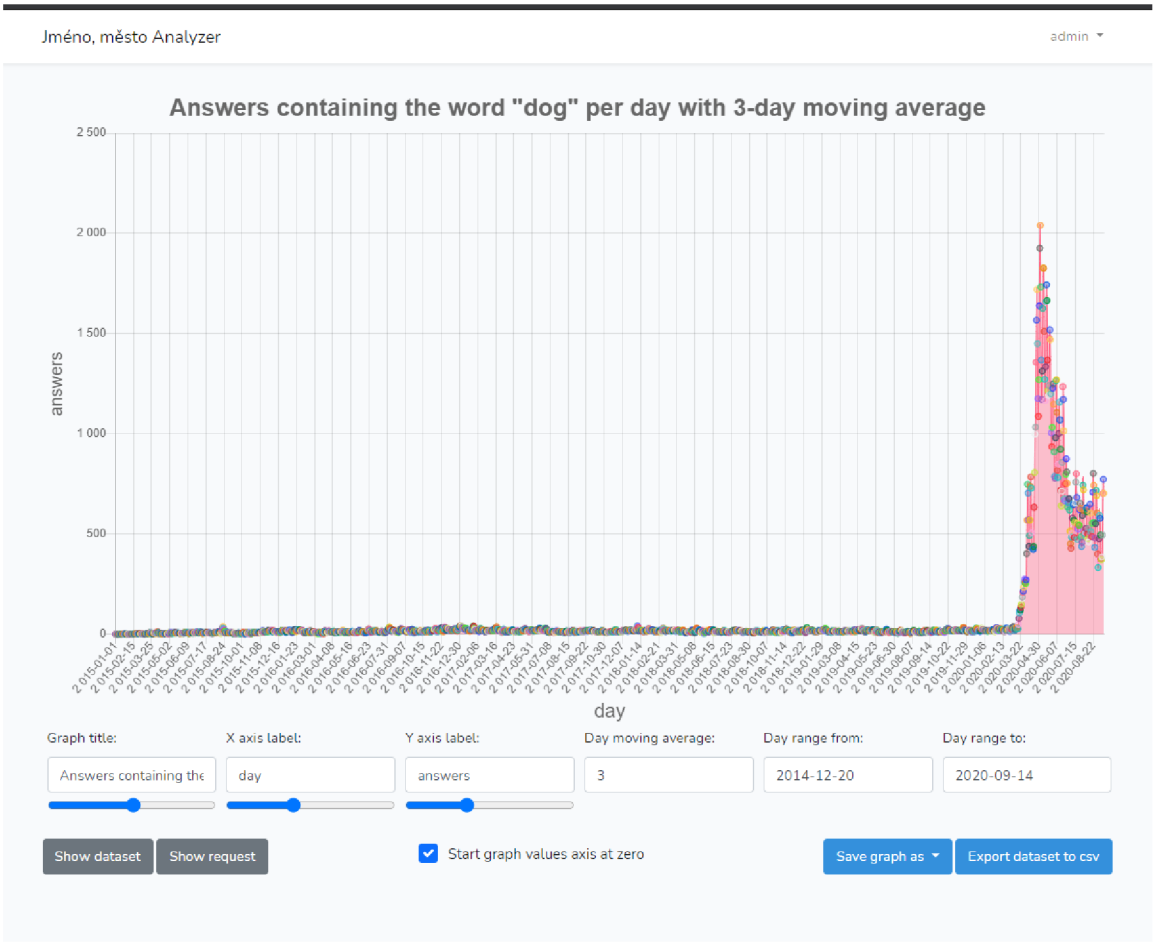

Figure 6.10: An example of a "Answer popularity over time" selection results page.

Application Programming Interface

The Application Programming Interface (API) represent the core of the application's back-end. The API compiles and provides all of the views for the front-end.

The main feature of the API is the ability to create executable database queries and return their results. After the user fills in the form fields in the query builder UI, a request is sent to the API. This request carries the specifications of the query selection chosen by the user. The fields are then validated, and if the validity is not violated, they are used to build an SQL query which, if the build is successful, is executed. The execution results are converted to percentages had the user selected this option, and the final results are returned to the front-end in a view.

A detailed description of the API and its source code can be found in Appendix B.

Testing

Testing was carried out by using the application to help perform the analyses in Chapter 5.

Chapter 7

Suggestions for Game Improvement

The game *Jméno, město* and its various language versions have gained worldwide popularity over the last few years. The game has undergone many changes over time to improve gameplay and marketability. As part of this work, the initial goal was to modify the application in order to improve the game. However, since there was no access granted to the game itself, only to a copy of the database, the created software is a stand-alone application that works completely separately from the game. Thus, this chapter proposes changes that could improve the game in terms of both user experience and revenue.

7.1 User Interface Design

The game's biggest competitor is *StopotS*, which is also a party online game that works on an identical basis. At first glance, *StopotS* offers a more modern-looking and uncluttered design, something that would benefit *Jméno, město* as well. The homepage of *Jméno, město*, shown in Figure 7.1, is usually the first page the user sees before playing a game. Since it serves as an introduction to the game, it must leave a good first impression because otherwise the user may reconsider playing the game and leave the site. The background image, appearing on all pages, is of low quality and gives the site a dated look. The stats panel on the centre-left is arguably unnecessary and distracting, as is the positioning of the advertisements, with one occurring on the centre-right and one at the bottom of the page. A more eye-pleasing solution would be to remove the background image and the stats panel, and place the advertisement panels vertically from top to bottom on both sides, farther from the centre of the page. The panels would thus not distract from the game description and the *Play game* button placed in the very centre. At the same time, the advertisements could take up at least as much, or even more, space to maximize revenue. The game would also merit having a logo at the top of the page instead of the game's name in plain text. The layout of the entire centre section of the page could then be designed using the Golden Ratio[19], more specifically the Rule of Thirds, which would make it look natural and aesthetically pleasing. The same design principles could then be applied to the rest of the pages to maintain a consistent appearance.

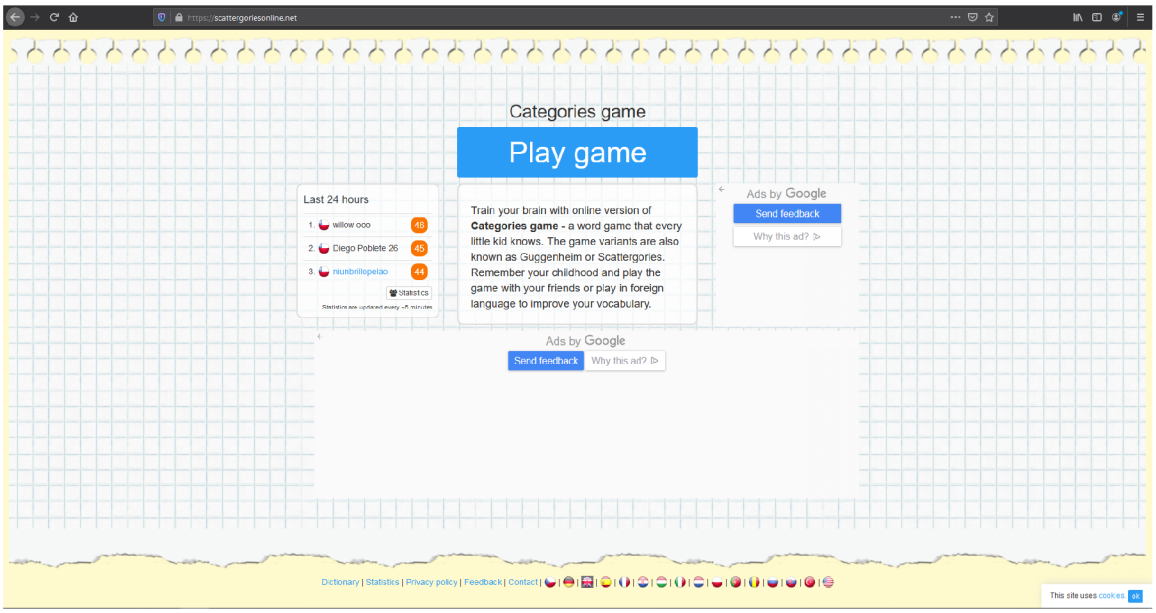

Figure 7.1: Homepage of the English version of the game. Homepages of all the language versions only differ in the language used, not in the UI.

7.2 Gameplay

Any new ideas regarding the gameplay of the game are limited by the rules of the original board game. Should the game want to remain faithful to the original, it would not make sense to change the gameplay significantly. Thus, the parts that could be changed are not directly related to the rules of the game but rather to the visual aspects and different ways a game could be played.

The first suggestion is to show players the initial letter draw. Before a round starts, the players are shown a five-second countdown to prepare them for the start. As part of this countdown, it would be more useful to simultaneously show the players the letter draw. This would force the players to pay attention to the round's start and allow them to prepare better for the round. Of course, this visualization would not have to be identical to how the draw is programmed in the game code; it would serve mainly as a visual representation of the draw.

After a game is finished, the players also have the option of playing the game again with the same categories. With this option, though, players cannot set the allowed starting letters. Thus, all starting letters can be drawn again, resulting in a round identical to a round already played before. One solution would be to allow players to select letters not only when creating the first game but also when re-playing. However, it is not possible to expect players to always remember all the initial letters of the previous game, so offering the option to automatically filter out these letters or allowing the players to skip a round would be ideal.

The game enjoys considerable popularity on Twitch, where people live stream their game sessions with their friends. A modern trend in online party games is to involve the viewers of these broadcasts directly in the game. Such features make the game more attractive to both the viewers, who appreciate the opportunity to influence the events of the broadcast, and the creators, who thereby attract and retain more viewers to their broadcasts. In the

game *Jméno, město*, the audience could be added as an additional player. The game would collect responses in a specific format from the chat of the broadcast. After the other players have finished their round, it would tally the most frequent responses per category, which it would mark as the answers of the audience. In the same way, viewers could also be involved in voting on the correctness of the answers and collectively represent one vote. To avoid potential griefing, the creators could turn off one or the other part of this feature.

7.3 Revenue

As mentioned in the previous paragraph, Twitch is a popular platform and should the game want to expand its player base, it should also emphasize its presence on Twitch. Twitch allows creators to mark what they are doing or what game they are playing during a broadcast. This tagging breaks down broadcasts into categories by which one can search for live broadcasts, past broadcasts, or so-called clips – short, one-minute maximum, video snippets from broadcasts. In order for a game to be tagged, it must be included in the list of games. Unlike its competitors such as Stopots, *Jméno, město* is not on such a list and cannot have its own category. It costs nothing to be put on the games list, and by not being on it, the game loses the possibility of more exposure.

An interesting way to make the game more profitable would be to offer sponsored categories. Product companies might be interested in how well known their products are compared to their competitors. These companies typically pay for surveys to obtain this information. However, respondents tend not to be paid very highly for completing surveys and often click through the answers without much thought, sometimes even without filling in the correct information. This problem would not arise in *Jméno, město*, as the players try to win the game by filling in correct answers, either those that come to mind first or, if they find the answer too obvious, meaning getting a lower score, they even fill in less frequent answers. This approach generates fairly accurate information about which brands are known among people. When creating a new game, some basic categories are already pre-selected so that players can quickly start a game. Sponsored categories could be pre-select in the same way. It can be assumed that most players would still choose to pick categories manually and decide to remove the sponsored category. However, if the category name is appropriately designed and sounds engaging, it could attain a sufficient number of plays.

Should the owners become willing to invest in marketing, they could further increase the number of players. Considering how much influence videos and live broadcasts have on the number of players, it would not be out of place to approach some creators to collaborate. The creators would receive a certain amount of money upfront, and in return, they would play the game and show it to their audience. As identified in section 5.1, although the Spanish and German versions of the game are prevalent, the English version does not have many players in comparison. This difference could also be attributed to the fact that the game's board game original is not well known in English-speaking countries. Competing games do not hold a strong position in this market either. Hence, for many people, this would be an entirely new concept of a game that they have not encountered before. Targeting a marketing campaign on English-speaking countries, and the US in particular, could potentially increase the number of players and thus the revenue drastically. The big advantage of this market and the English-speaking creators is that their audience is international. That means that the marketing campaign would be reflected in the popularity of all the language versions of the game. This market could also be reached with a targeted

pay-per-click marketing campaign to players of similar online party games who are not familiar with *Jméno, město* already.

Chapter 8

Conclusion

This thesis aimed to propose analytical methods to analyse the responses and behaviour of players of the game "Jméno, město", use them to explore the provided database and then evaluate the results and validity of the methods. Furthermore, the aim was to develop an application that would simplify data collection and analysis. The final goal was to extend this application to directly improve the game itself in terms of player experience and profitability.

Four types of analyses were performed: on the number of players, on geocultural differences, on cross-lingual differences and on the trends in responses. Analysis of the change in player numbers over time found that spreading awareness of the game was key to expanding the player base across languages. YouTube videos featuring gameplay footage with millions of views and live streams on the platform Twitch were most often playing a pivotal role. The global pandemic of the covid-19 disease in 2020 also significantly impacted the number of players, with an increase of hundreds to thousands of per cent in the number of games across all languages. The geocultural analysis addressed differences in the cultures of different English-speaking countries. For this analysis, the United States, Canada, the United Kingdom and Australia were selected. Based on players' responses from these countries, the analysis concluded that the cultures in these countries are nowadays very similar, yet differences can still be commonly found. The more geographically distant the countries are, the more noticeable these differences become. The cross-lingual analysis addressed the differences between popular responses in Czech and Slovak. These are very often similar, but as with the previous analysis, differences can be found. The data shows that Slovak-speaking players are aware of the popular answers in Czech, but not necessarily the other way around. Finally, the analysis of trends in answer popularity over time examines terms related to the 2020 pandemic and the popularity of the application TikTok. In most cases, the results of the analyses correlated with data from Google Trends. Thus, all methods proved capable of producing meaningful results. These methods could thereby be used for further analyses.

The developed application allows the user to choose what data they want to extract from the database using a simple and clear UI. It then displays this data in a graph that can be further customised and exported. Similarly, a table with the found data can be optionally displayed and exported. The application allows collecting the data necessary to perform analyses, and the generating graphs make it easier to evaluate the quality of the data for the selected analysis. The application was an important tool to improve and speed up performing the analyses mentioned above. A possible further extension of the application would be more a detailed customisation of the graphs and more export options.

The goal of extending the application with features to improve the game itself could not be met in the end, as no access was granted to the game and the developed application works fully autonomously. Instead of extending the application, the thesis presents suggestions for improving the player experience and increasing profitability. These suggestions were compiled based on personal experience of playing the game and the result of the analysis of the change in the number of players. The thesis proposes changes to the UI design and gameplay, adding support for integrating viewers of live broadcasts on the Twitch platform into gameplay, and recommends an overall greater focus on this platform. To possibly increase revenue, it proposes the creation of sponsored categories and a marketing campaign targeting English-speaking players, where awareness of the game is not yet widespread.

Bibliography

[1] *About JavaScript - JavaScript | MDN* [online]. Mozilla [cit. 28. 4. 2021]. Available at: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript.

[2] *Chart.js - Introduction* [online]. Chart.js [cit. 28. 4. 2021]. Available at: https://www.chartjs.org/docs/latest/.

[3] *CSS basics - Learn web development | MDN* [online]. Mozilla [cit. 28. 4. 2021]. Available at: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics.

[4] *Five Cents | Royal Australian Mint* [online]. Royal Australian Mint [cit. 28. 4. 2021]. Available at: https://www.ramint.gov.au/five-cents.

[5] *Installation - Laravel - The PHP Framework For Web Artisans* [online]. Laravel LLC [cit. 28. 4. 2021]. Available at: https://laravel.com/docs/8.x#meet-laravel.

[6] *Introduction* [online]. Bootstrap team [cit. 28. 4. 2021]. Available at: https://getbootstrap.com/docs/5.0/getting-started/introduction/.

[7] *Introduction to HTML* [online]. Refsnes Data [cit. 28. 4. 2021]. Available at: https://www.w3schools.com/html/html_intro.asp.

[8] *JQuery* [online]. OpenJS Foundation [cit. 28. 4. 2021]. Available at: https://jquery.com/.

[9] *MVC Framework - Introduction* [online]. Tutorialspoint [cit. 28. 4. 2021]. Available at: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm.

[10] *PHP: New features - Manual* [online]. The PHP Group [cit. 28. 4. 2021]. Available at: https://www.php.net/manual/en/migration70.new-features.php.

[11] *PostgreSQL: About* [online]. The PostgreSQL Global Development Group [cit. 28. 4. 2021]. Available at: https://www.postgresql.org/about/.

[12] *PostgreSQL: Documentation: 13: Part II. The SQL Language* [online]. The PostgreSQL Global Development Group [cit. 28. 4. 2021]. Available at: https://www.postgresql.org/docs/13/sql.html.

[13] *Usage Statistics and Market Share of JavaScript Libraries for Websites, April 2021* [online]. Q-Success [cit. 28. 4. 2021]. Available at: https://w3techs.com/technologies/overview/javascript_library.

[14] *What can PHP do? - Manual* [online]. The PHP Group [cit. 28. 4. 2021]. Available at: https://www.php.net/manual/en/intro-whatcando.php.

[15] *What is PHP? - Manual* [online]. The PHP Group [cit. 28. 4. 2021]. Available at: https://www.php.net/manual/en/intro-whatis.php.

[16] *Spectator Attendance at Sporting Events, 2009-10* [online]. Australian Bureau of Statistics, Dec. 21, 2010 [cit. 28. 4. 2021]. Available at: https://www.abs.gov.au/ausstats/abs@.nsf/Products/4174.0~2009-10~Main+Features~Most+popular+sports+attended.

[17] De Smith, M. J. *Statistical Analysis Handbook*. Edinburgh: The Winchelsea Press, Drumlin Security Ltd, 2018. ISBN 9781912556076.

[18] Denton, M. *Game Genre & Statistics: Not All Games Are Created Equal (part 1)* [online]. 2021 [cit. 28. 4. 2021]. Available at: https://www.gamify.com/gamification-blog/not-all-games-are-created-equal-pt1.

[19] Gross, R. *What is the golden ratio?* [online]. [cit. 28. 4. 2021]. Available at: https://www.canva.com/learn/what-is-the-golden-ratio/.

[20] Ighodaro, N. *How Laravel implements MVC and how to use it effectively* [online]. [cit. 28. 4. 2021]. Available at: https://blog.pusher.com/laravel-mvc-use/.

[21] LaPierre, J. *Gaming's Biggest Demographics are Changing – Here's Why That Matters* [online]. Jul. 20, 2020 [cit. 28. 4. 2021]. Available at: https://www.filamentgames.com/blog/gamings-biggest-demographics-are-changing-heres-why-that-matters/.

[22] Levene, M. and Loizou, G. *A Guided Tour of Relational Databases and Beyond*. Springer London, 1999. ISBN 9781852330088.

[23] Whitehorn, M. and Marklyn, B. *Inside Relational Databases with Examples in Access*. London: Springer London, 2007. ISBN 9781846283949.

Appendix A

Game Database

Game

The attributes that each record of the relation consists of are the following:

- `id` – [PK] unique game record identifier
- `max_players` – maximum number of players of the game
- `cnt_rounds` – number of rounds set to be played in the game
- `id_usr` – [FK] identifier of the user who created the game
- `date_cr` – timestamp of the creation of the game
- `ts` – timestamp
- `round_timer` – duration of a round of the game
- `src_domain` – top-level domain from which the game was created on
- `id_lang` – [FK] identifier of the language the game is played in
- `flag_finished` – flag that states whether the game was played until the end and finished
- `id_cat_game_settings_type` – [FK] identifier of the game categories setting type

The `id_cat_game_settings_type` attribute refers to the "cat_game_settings_type" relation, which is an enumerated type consisting of values `DEFAULT_CATEGORIES`, `USER_CHANGED_CATEGORIES`, `SAME_CATEGORIES`, `RANDOM_CATEGORIES`, and `CATEGORIES_SET`.

Round

The attributes that each record of the relation consists of are the following:

- `id` – [PK] unique round record identifier
- `starts_with` – initial letter that all answers have to start with in a round
- `round` – round position out of all rounds
- `id_game` – [FK] identifier of the game the round is part of

Category

The attributes that each record of the relation consists of are the following:

- `id` – [PK] unique category record identifier
- `name` – name of the category
- `id_usr` – [FK] identifier of the user who created the category
- `date_cr` – timestamp of the creation of the category
- `games_played` – amount of games the category had been selected in
- `verified_flag` – flag that states whether the category was verified by the owner of the application
- `id_cat_category_status` – [FK] identifier of the category status
- `id_duplicity` – identifier of a category which is a duplicate of this category
- `id_word_bullshit_regex` – [FK] identifier of a filter regex
- `id_lang` – [FK] identifier of the language the category was created in

The `id_cat_category_status` attribute refers to the "cat_category_status" relation which is an enumerated type consisting of values `KIDS_OK`, `ADULT_ONLY`, `ILLEGAL`, `NONSENSE` and `DUPLICITY`. Categories that have the `verified_flag` set as "true" are more likely to be featured at the top of the categories list for player to select from.

Word

This relation stores data about each answer to a category in the whole game. The "word" relation has a one-to-many relationship with the "word_report" relation. This means that an answer can be reported multiple times.
The most important attributes for the analysis are:

- `id` – [PK] unique answer record identifier
- `value` – string containing the submitted answer
- `id_round` – [FK] identifier of the round the word was answered in
- `date_cr` – timestamp of the submission
- `id_category` – [FK] identifier of the category the word was answered in
- `id_game_player` – [FK] identifier of the player who sent the answer

Usr

The attributes that each record of the relation consists of are the following:

- `id` – [PK] unique user record identifier
- `flag_guest` – flag that states whether the user is registered or not
- `date_cr` – timestamp of the creation of the "usr" record
- `robot_flag` – flag that states whether the user is a real person or a robot
- `id_cat_usr_status` – [FK] identifier of the user status
- `lang` – language the user is playing in
- `country_code` – country in which the user is located
- `id_cat_privacy_settings` – [FK] identifier of the privacy setting

The `id_cat_usr_status` attribute refers to the "cat_user_status" relation, which is an enumerated type consisting of values `ACTIVE`, `DELETED`, `BANNED`, and `SCHEDULED_DELETE`. The `id_cat_privacy_settings` attribute refers to the "cat_privacy_settings" relation, which is an enumerated type consisting of values `PUBLIC`, `PRIVATE`, and `NO DRAWINGS (LDI)`.

Game player

The attributes that each record of the relation consists of are the following:

- `id` – [PK] unique user record identifier
- `id_round` – [FK] identifier of the game this player is a part of
- `id_usr` – [FK] identifier of the "usr" table record this player belongs to
- `flag_robot` – flag that states whether the user is a real person or a robot
- `flag_guest` – flag that states whether the user is registered or not
- `round_left` – number of a round during which the player left the game
- `score` – amount of points earned in the game
- `date_cr` – timestamp of the creation of the "game_player" record which is the start of the game
- `victory_points` – amount of victory points earned for winning the game
- `country_code` – country in which the user is located
- `lang` – language the user is playing in

Lang

The attributes that each record of the relation consists of are the following:

- `alphabet` – alphabet of the language
- `date_cr` – timestamp of the creation of the "lang" record
- `default_alphabet` – altered alphabet of the language that does not contain letters too challenging to play with
- `domain_name` – name of the default site domain for the language
- `catalog_preposition` – equivalent of the English "with" preposition for the phrase "starting with"
- `flag_enable` – flag that states whether the language is enabled in the game
- `show_name` – name of the language in the respective language
- `id` – [PK] unique two-character language code

Word Report

This relation stores which answers were reported by the players at the end of a round as not correct.
The attributes that each record of the relation consists of are the following:

- `id` – [PK] unique word report record identifier
- `id_word` – [FK] identifier of the answer reported
- `date_cr` – timestamp of the report
- `id_player` – [FK] identifier of the player who submitted the answer

Word Bullshit Regex

This relation stores regexes used for filtering offensive words, characters and other inappropriate answers. Some of the players might be children, and these regexes serve to avoid showing offensive content to them.
The attributes that each record of the relation consists of are the following:

- `id` – [PK] unique regex identifier
- `id_cat_language` – [FK] identifier of the language the regex applies to
- `regex` – regex to filter offensive words, characters and other inappropriate answers
- `note` – note describing the regex
- `bullshit_type_flags` – flag that states the type of the filter

Appendix B

Application Programming Interface

The directory tree of the Application Programming Interface (API) files covered by this section is shown below.

```
/
├── app
│   ├── Actions
│   │   ├── ExecuteSqlCommandAction.php
│   │   ├── GetGameLanguagesAction.php
│   │   └── GetIsoCountriesAction.php
│   ├── Constants
│   │   └── QueryConstants.php
│   ├── Http
│   │   └── Controllers
│   │       └── MainController.php
│   ├── Rules
│   │   └── CountryExistsRule.php
│   └── Services
│       ├── QueryBuilderService.php
│       ├── RequestHandlerService.php
│       └── RequestInputService.php
└── routes
    └── web.php
```

The implicit API consists of user authentication routes and two routes to the base URI "/". One of them for the HTTP method GET, the other for the method POST. All of the application's routes are defined in the file `web.php`. Both of the mentioned routes to the base URI are redirected to the controller class `MainController`.

Method GET to URI "/" redirects to the `MainController` class method `index`. This method is responsible for providing the main view, which is described in more detail in section 6.3. The method firstly receives a list of ISO3166[1] countries by calling the `GetIsoCountriesAction` action, which uses the `league/iso3166`[2] library to get the list. Then, the `GetGameLanguagesAction` action is used to retrieve the languages available in

[1]More information at: https://www.iso.org/iso-3166-country-codes.html
[2]More information at: https://packagist.org/packages/league/iso3166

the game from the database. Finally, these two lists are sent to the view, which is built and returned to the browser.

Method POST to URI "/" redirects to the `MainController` class method `handleRequest` shown in Listing B.1. This POST request carries information about the type and other specifications of the query selection chosen by the user. The method is responsible for returning the results of the user's selection. The process is described in detail below.

```
/**
 * Pass request and get query results.
 *
 * @param Request $request
 * @return View
 */
public function handleRequest(Request $request): View
{
    $requestInputService = new RequestInputService($request);
    $requestHandlerService = new RequestHandlerService(new
        ↪ QueryBuilderService($requestInputService), $requestInputService);

    $queryResult = $requestHandlerService->handle();
    $query = $requestHandlerService->getQuery();
    $filteredRequest = $requestHandlerService->getFilteredRequest();

    return view('main.graph', [
        'data' => $queryResult,
        'request' => $filteredRequest,
        'query' => $query
    ]);
}
```

Listing B.1: Method `handleRequest` of class `MainController`.

Query Selection Request Handling

Method `handleRequest` of the `MainController` controller first creates objects of all three application services. These services are `RequestInputService`, `QueryBuilderService`, and `RequestHandlerService` and they represent the core of the request resolving. All services are located in the folder `/app/Services`. Then, the `handle` method of the `RequestHandlerService` service, shown in Listing B.2, is called to return the results of the user input in a form ready to be presented at the front-end. The service also stores the SQL query and filters the user input, both also returned to the controller. The results, the SQL query, and the filtered input are then sent to the view. Finally, the view is built and returned to the browser.

```php
/**
 * Handles request and returns query results.
 *
 * @return array<int, array<string, int>>
 */
public function handle(): array
{
    $request = $this->requestInputService->getRequest();
    $this->filterRequest($request);

    $this->query = $this->queryBuilderService->build();
    $queryResult = $this->executeQuery();

    return $this->prepareResults($queryResult);
}
```

Listing B.2: Method `handle` of class `RequestHandlerService`.

The `RequestHandlerService` class calls the `RequestInputService` to get the user's request. The request is then filtered by the method `filterRequest` so that it can be shown to the user on the front-end. The first item of the request, the user token, is removed as well as all empty inputs and, in the case of the "Answer popularity over time" selection, an excess comparison operator.

Afterwards, the service calls the `QueryBuilderService` service to build an SQL query from the user's request. This query is then executed using the `ExecuteSqlCommandAction` action. The returned query results are not quite ready to be returned to the controller. The method `prepareResults`, shown in Listing B.3, is called. If the user selected the option to have the results in the percentage out of all related answers, method `changeResultsToPercentage` is called, too. For the "Most popular answers" selection, each result's item amount is added up to a sum. Then, the user-selected limit of returned entries is applied, and the remaining entries are divided by the previously calculated sum. The results from the "Answer popularity over time" selection already contain the sum of the answers for each day, so it is only needed to make the division. If the percentage option is not selected, the method `prepareResults` only applies the user-selected entry limit for any selection except for the "Answer popularity over time" selection.

```php
/**
 * shortens the array by inputted limit and changes to percentage if
    ↪ necessary
 *
 * @param array<int, array<string, mixed>> $result
 *
 * @return array<int, array<string, mixed>>
 */
private function prepareResults(array $result): array
{
    if ($this->requestInputService->getInputValue(QueryConstants::
        ↪ PERCENTAGE_KEY)) {
        return $this->changeResultToPercentage($result);
    }

    if ($this->requestInputService->getInputValue(QueryConstants::
        ↪ GRAPH_TYPE_KEY) !== QueryConstants::TIME_GRAPH) {
        $limit = $this->requestInputService->getInputValue(QueryConstants::
            ↪ LIMIT_KEY);
        array_splice($result, $limit);
    }

    return $result;
}
```

Listing B.3: Method `prepareResults` of class `RequestHandlerService`.

The `QueryBuilderService` class is responsible for the creation of the query that will be executed. The user's request is parsed, and once the type of the selection is known, a method resolving that selection is called. Some of the parts of the query are needed to be built for several selection types and need their own method. These methods are `BuildFromSubQuery` and `BuildWhereSubquery`.

The former method creates the part of the query that belongs to the `FROM` statement. As described in section 6.3, the relation that contains the answers from players is separated into ten smaller relations. This method selects the correct relation by the selected language. If the user chooses to search in all languages, all of the ten relations have to be merged, as shown in Listing B.4.

The latter method creates the part of the query belonging to the `WHERE` statement. This method resolves the user selection of the language, the countries of origin, the categories and the letter. The comparison of the category specification was first implemented as a string comparison, comparing each answer's category name with the ones that the user provided. It became clear during further development that this approach is not optimal because a string-to-string comparison proved to be very time-consuming. As a solution to this problem, the method `findCategoryIds` was implemented. This method builds a query that finds the identifiers of the provided category names for the selected language (or any language if "all languages" is selected), executes the query, and returns the found identifiers. These identifiers are then used for comparison instead of the category names. This solution

reduced the execution time of a query with a specified category down to a half compared to the string comparison approach.

```
$languageTables = ["word_cs", "word_pt", "word_de", "word_en", "word_fr", "
    ↪ word_es", "word_it", "word_pl", "word_sk"];

$fromQuery = " FROM ";

if ($language === QueryConstants::ALL_LANGUAGES){
    $fromQuery .= "(";

    foreach ($languageTables as $table) {
        $fromQuery .= "SELECT value, date_cr, id_category, country_code " .
            "FROM " . $table . " " .
            "UNION ALL "
            ;
    }

    $fromQuery .= "SELECT value, date_cr, id_category, country_code " .
                  "FROM word_rest " .
        ") AS word_tables"
    ;
}
```

Listing B.4: Merging answer relations when the "all languages" option is selected.

The `RequestInputService` service's primary purpose is to validate and return user inputs from the request. Different user inputs are needed at different times of query handling, and different types of queries need different user inputs. These inconsistencies are why input is validated every time it is requested from the other classes. The validations are implemented using the Laravel `validate` method[3]. Some inputs need changes before they can be used. To make sure this it not forgotten, any universal changes to inputs are performed just after the validation. An abbreviated version of the `getInputValue` method that implements the described functionality is shown in Listing B.5.

As seen in this Listing, the validation of countries calls the `CountryExistsRule` class. This class is a custom rule that checks whether an inputted country exists in the ISO3166 standard list.

The service is also called near the beginning of the query building process to modify all of the request inputs at once to prevent an SQL injection[4] attack. This attack is prevented by escaping the "'" character by doubling it. An attacker would not be able to modify the SQL command because whatever the attacker sends in would be treated as a whole string.

[3]More information at: https://laravel.com/docs/8.x/validation
[4]More information at: https://www.w3schools.com/sql/sql_injection.asp

```php
/**
 * Validates input and returns its value.
 *
 * @param string $input
 *
 * @return mixed
 */
public function getInputValue(string $input): mixed
{
    match ($input) {
        QueryConstants::LIMIT_KEY => $this->request->validate([
            ↪ QueryConstants::LIMIT_KEY => 'required|integer|min:1|max:' .
            ↪ QueryConstants::LIMIT_NUMBER]),
        QueryConstants::CATEGORY_KEY => $this->request->validate([
            QueryConstants::CATEGORY_KEY => 'array',
            QueryConstants::CATEGORY_KEY.'.*' => 'nullable|string'
        ]),
        QueryConstants::COUNTRY_KEY => $this->request->validate([
            QueryConstants::COUNTRY_KEY => 'array',
            QueryConstants::COUNTRY_KEY.'.*' => ['nullable', 'string', new
                ↪ CountryExistsRule($this->isoCountries)]
        ]),
    };

    return match ($input) {
        QueryConstants::LIMIT_KEY => intval($this->request->input(
            ↪ QueryConstants::LIMIT_KEY)),
        QueryConstants::COUNTRY_KEY => array_map(
            array($this, 'getCountryCode'),
            array_filter($this->request->input(QueryConstants::COUNTRY_KEY))
        ),
        QueryConstants::CATEGORY_KEY => array_map(
            fn ($str): string => mb_strtolower($str, 'UTF-8'),
            array_filter($this->request->input(QueryConstants::CATEGORY_KEY)
                ↪ )
        ),
    };
}
```

Listing B.5: Abbreviated method for validating and adjusting inputs from the request.

The `QueryConstants` class contains all of the API's universal constants. Using constants instead of hard-coded strings makes for a more precise and infallible code and makes it easier to maintain and make changes to the code.