

UNIVERSITY OF TARTU  
FACULTY OF SOCIAL SCIENCES

NARVA COLLEGE  
INFORMATION TECHNOLOGY SYSTEMS DEVELOPMENT

Makar Drozdik  
CREATING AN ESTONIAN VOCABULARY LEARNING GAME FOR ANDROID  
PLATFORM

Diploma thesis

Supervisors: M.Sc. Andre Säask

NARVA 2023

## **Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks**

Mina, Makar Drozdik

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose Creating an Estonian vocabulary learning game for android platform, mille juhendaja on Andre Säask reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Makar Drozdik

10.05.2023

## Contents

TERMS AND ABBREVIATIONS.....	6
INTRODUCTION.....	7
The Problem.....	8
The Solution .....	8
The Aim.....	8
Tasks.....	8
Contents.....	9
1    MARKET RESEARCH .....	10
1.1    Drops.....	10
1.2    Speakly.....	10
1.3    Learn Estonian.....	11
1.4    Ling .....	11
1.5    50Languages Learn Estonian.....	11
1.6    Simply learn Estonian.....	12
1.7    Conclusion for market research.....	12
2    OVERVIEW OF TOOLS AND TECHNOLOGIES CHOSEN.....	14
2.1    Android.....	14
2.2    Android Studio .....	15
2.3    Google Play Store .....	15
2.4    Java .....	15
2.5    LibGDX.....	16
2.6    JSON .....	17
2.7    Hiero.....	17
3    RESULT OF THE DEVELOPMENT .....	18
3.1    Functional Requirements for the application.....	18
3.2    Non-functional requirements for the application.....	19

3.3	Design elements.....	19
3.4	Application interface description.....	20
3.4.1	Loading screen.....	20
3.4.2	Main menu screen .....	20
3.4.3	Settings screen.....	21
3.4.4	Choose word set screen.....	22
3.4.5	Dictionary adding screen.....	22
3.4.6	Game screen.....	24
3.4.7	Game over screen.....	26
3.5	LearningGame.....	27
3.6	Screens.....	28
3.7	Camera.....	29
3.8	Viewport .....	29
3.9	HUD.....	30
3.10	Stage .....	30
3.11	SpriteBatch .....	30
3.12	Asset Manager .....	31
3.13	Font.....	31
3.14	AppPreferences.....	31
3.15	Box2d World .....	31
3.16	B2BodyFactory.....	32
3.17	Contact listener.....	32
3.18	Entities custom classes .....	33
	CONCLUSION .....	34
	RESUMEE .....	35
	REFERENCES.....	36
	Appendix 1.....	38



## TERMS AND ABBREVIATIONS

**Application** – an Android application is a software program designed to run on the Android operating system.

**OS** – is system software that manages computer hardware, software resources and provides common services for computer programs.

**Gameplay loop** – fundamental set of actions that a player repeatedly performs within a game.

**Word set** – in this thesis word set is referred to a user defined dictionary of words that consists of at least three pairs of word and translation for that word.

**Reach** – in the context of software sales, the term "reach" refers to the extent to which a software solution can attract and cater to potential customers across different demographics.

**Skin** – The Skin class in LibGDX stores resources for UI widgets to use. It is used to conveniently store texture regions, fonts, colors, etc. Skin also provides convenient conversions, such as retrieving a texture region as sprite, or drawable. (LibGDX, 2023)

**UI** – User Interface

**Gamification** – the practice of making activities more like games to make them more interesting or enjoyable. (Definition of “gamification” ... 2018)

**HUD** – heads-up display

## INTRODUCTION

Motivation is considered as the most crucial factor for achieving success in the learning process. It serves as a driving force that determines whether a learner will undertake a task, the amount of energy they will dedicate to mastering a specific language, and the extent to which they will persevere in their efforts. The absence of sufficient motivation can impede the progress of even the most proficient students, rendering them unable to achieve their long-term objectives (Wang, 2000).

According to a study conducted by Alsawaier (Alsawaier, 2018), gamification can significantly enhance motivation and engagement in various contexts, including education. The research suggests that gamification is an effective approach due to its ability to provide a sense of accomplishment, stimulate competition and collaboration, and enrich the overall learning experience.

As such, there has been a growing interest in incorporating gamification techniques into language learning to increase student motivation and engagement. Gamification is the process of applying game design elements and principles to non-game contexts, such as education, to make them more engaging and enjoyable. By introducing features like gameplay, and according to Michael Sailer (Sailer, 2017) specific game elements, like progress tracking, language learning can be transformed into a more interactive and immersive experience.

Based on Papastergiou's study (Papastergiou, 2009), it was found, that gamification can be an effective tool for improving student motivation and enhancing the learning experience. The research shows that students who utilized a gamified educational application demonstrated an improvement in their learning outcomes when compared to those who used a non-gamified application.

This diploma project aims to explore the development of a gamified mobile application designed to enhance the learning experience for people who study Estonian language vocabulary. The game will provide a customizable tool for users to learn Estonian vocabulary through an interactive game, with the aim of increasing their motivation and engagement. By incorporating gamification elements into the learning process, this project hopes to contribute to the field of language learning and provide a useful resource for Estonian language learners.

## The Problem

The author found out that the language learning market misses an application which would allow users to create their own word sets while at the same time being engaging and entertaining by gamifying the process of learning.

## The Solution

To address this issue the development of a game with customizable vocabulary that caters to all levels of Estonian language proficiency could be a potential solution. The ability for users to create their own sets of words eliminates the limitation of predetermined sets of words, making the application useful for a wider range of language learners. Additionally, the incorporation of gamification elements enhances the learning experience and promotes engagement and motivation for users. Overall, such a game would provide an effective and enjoyable method for language learners to expand their vocabulary and improve their fluency in Estonian in particular.

## The Aim

The objective of this university diploma project is to develop an educational mobile game designed to aid in learning of Estonian vocabulary. The aim is to create an application that would provide a different experience compared to other similar language learning applications currently available on the market. Also, instead of looking like an interactive dictionary, the application should have the feel of a game.

The main feature of the application is for user to have the ability to create and store their own custom word sets. Users should be able to write word pairs and their corresponding translations, and these could be combined to create personalized sets of words.

This application should be designed in a way to be a companion for reinforcement and assistance in learning new words. Users should be able to decide which words to learn either by themselves or with the help of the Estonian language teacher.

In addition, the application is required to provide feedback to players regarding their performance. This feature would allow users to monitor their progress.

## Tasks

To achieve the goal following tasks must be completed:

1. Analyze the current market for available solutions.
2. Choose the most suitable technologies for the task.
3. Study the necessary technologies.
4. Develop the core gameplay.
5. Develop the game interface.
6. Develop the main menu interface.
7. Develop the ability to include custom word sets in the gameplay.
8. Develop the ability to save data between the game sessions.

## Contents

Section 1 describes market research for similar solutions that was done their examples.

Section 2 describes which technologies were used during the development and their usage rationale.

Section 3 provides a description of the development process, including creating the application interface.

# 1 MARKET RESEARCH

To explore the existing language learning solutions available for the Estonian language, the most relevant applications available on Android found with the keyword search "learn Estonian" were examined, as well as the top results that appeared when making a Google search on the same topic. Applications listed below were downloaded at least 10000 times.

## 1.1 Drops

Drops is a language learning app that offers a game-like interface to make vocabulary acquisition enjoyable and easy. The app utilizes illustrations instead of translations to enhance retention, and users are required to practice for only five minutes a day. Drops focuses on practical vocabulary and excludes grammar, making it accessible to all users. The app aims to develop a habit of daily language learning for its users. It provides pronunciation by experienced speakers and is available for free, with the option of upgrading to the premium version for more extensive learning. The app aims to empower people worldwide to develop language skills using pictures, which serve as a universal language. (Drops Languages, 2023)

The Drops app has the best interface, and it is the most well-built application from the list. The app utilizes illustrations and pronunciations that correspond to words, which aids in memorizing them. One of the app's strengths is the varied gameplay, which keeps users engaged and challenged. It gives user one random word, and then asks user if he knows this word or not. If the user does not know this word, it teaches user this word and then adds this word to the test.

Main differentiation is that it does not give user choice which word to learn. Second difference is that learning process is different.

## 1.2 Speakly

Speakly is a language learning application that prioritizes teaching Estonian vocabulary through real-life situations. The app uses dialogues and multiple-choice questions to facilitate learning through speaking, writing, and listening. It also does not have the ability to choose what to learn and instead teaches user pre-selected vocabulary.

Its main learning process is presented in a form of multiple-choice questions. Additionally, it offers services of the personal tutor, live situations, listening exercises, song listening and repetition of learned words.

Main difference is that it does not give user choice which words to learn. Learning process is also different.

### 1.3 Learn Estonian

The Learn Estonian app is designed for Android users who are interested in learning the Estonian language. The app includes a wide variety of words and phrases in Estonian, along with their Russian translations.

One of the app's main features is the ability to play the pronunciation of each word, which is useful for learners who want to improve their pronunciation skills.

The app also allows users to add words to their favorites, but only pre-existing words in the app can be added to that list, not user-defined words, or phrases.

Overall, while the Learn Estonian app has useful features such as word pronunciation and translation.

Main difference is that this application does not allow user to add their own words into the application.

### 1.4 Ling

Ling is interactive application to learn more than 60 languages including Estonian. It provides a range of interactive lessons that are combined into topics. It has listening and speaking exercises. It is used mostly used for learning vocabulary. The lessons each introduce user to a word and then give them the same word in a context. At the end of each lesson, there's a dialogue section. The speaking lesson use voice recognition.

It differentiates in not having an ability for user to sort words into their own word set and ability to choose which word to learn is limited to choosing topic.

### 1.5 50Languages Learn Estonian

"50 languages learn Estonian" is an Android and iPhone app that offers lessons from the 50LANGUAGES Estonian curriculum. The lessons are available as audio files, vocabulary cards, and games. The app features a built-in dictionary.

Main difference is that this application does not allow user to add their own words into the application.

## 1.6 Simply learn Estonian

"Simply Learn Estonian" is an Android application designed to help users learn the Estonian language. The application is aimed at beginners and offers various learning tools, such as flashcards, quizzes, and audio recordings. The app also includes a pronunciation guide to help users improve their speaking skills.

Users can choose from various lessons to start learning Estonian. The app has a user-friendly interface and is easy to navigate.

Main difference is that this application does not allow user to add their own words into the application.

## 1.7 Conclusion for market research

In summary, these applications provide a useful starting point for learning Estonian by offering a range of vocabulary and phrases in a user-friendly format with in some cases helpful features like pictures and audio pronunciations.

Game developed in this diploma project aims to distinguish it from other already existing language learning applications by providing two main features which should make it different enough to acquire its own niche audience.

Custom word sets are a feature that allows the user to create their own sets of vocabulary to learn. This feature gives the user the flexibility to personalize their learning experience and focus on the specific words or phrases that they want to learn. The user can select the words they want to learn, add them to their custom word set, and then practice them using the application's gamified learning environment.

A gamified learning environment is a feature in the language learning application that makes the learning process more engaging and entertaining.

Application developed for this work will be intended as a support tool for learning Estonian words. User will be able to decide themselves or with the help of a teacher, which words they will be learning. This app will not be suitable for those who want to start learning Estonian from beginning, its main purpose will be designed as a companion application for assistance and reinforcement in learning new words.

These unique features make the application well-suited to appeal to a specific niche of users in the market who value the ability to create their own word sets and desire a more engaging and challenging language learning experience.

## 2 OVERVIEW OF TOOLS AND TECHNOLOGIES CHOSEN

In this section the author provides a description of the tools and technologies utilized during the development process of the application. This includes an explanation of the rationale behind the selection of each tool and technology.

### 2.1 Android

The Android operating system is a comprehensive software platform that offers a range of hardware and software resources for mobile devices. The Android Framework is a collection of Application Programming Interfaces (APIs) that enable software developers to develop Android applications with ease and speed. In the context of this project, the Android operating system was selected as the preferred platform for the game application due to its user-friendly application distribution system, relatively low entry cost, broad application reach and easy distribution through the Google Play Store. Moreover, the Android operating system's touch screen functionality is suited for the game's interactive features, making it a suitable platform for the project's development. (Android Studio, 2023)

The primary objective of this project was to develop a software application on the Android operating system. This choice was made due to several key advantages, including the accessibility of development software, ease of use for end-users, and a wider reach for users through the Google Play Store.

The Android operating system is one of the most popular mobile platforms globally, with a significant market share and a vast community of developers worldwide. This popularity has resulted in a wide range of development tools and resources that are easily accessible, making it an attractive option for software development projects. Example of these tools are Android Studio and libGDX.

Furthermore, the Google Play Store provides a convenient and accessible platform for users to download and install applications developed on the Android operating system. This feature has enabled software developers to reach a larger audience, including international markets, with their software applications.

Overall, the Android operating system was chosen as the main platform for this project due to its accessibility, ease of use, and wider reach for end-users through the Google Play Store.

## 2.2 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, designed specifically for Android development. It is based on IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools. Android Studio offers a flexible Gradle-based build system, which is used to compile the code and create an APK file for submission to the Google Play Store (Wikipedia, 2023)

Android Studio is a software development environment that provides a dedicated platform for Android application development. This integrated development environment (IDE) offers advanced features and tools that cater specifically to the development needs of Android applications. One of the key features of Android Studio is its ability to emulate the Android operating system, allowing developers to test their applications on a virtual device.

By providing developers with a virtual platform for testing and debugging applications, the emulation capability of Android Studio gives them ability to identify and resolve issues before deploying applications on actual devices, improving the reliability and functionality of the application. Moreover, the emulation feature allows for the testing of application interface on various Android screen sizes, ensuring usability and functionality across a wide range of devices.

## 2.3 Google Play Store

The Google Play Store is a digital distribution service operated and developed by Google. It serves as the official app store for certified devices running on the Android operating system and its derivatives, as well as Chrome OS, allowing users to browse and download applications developed with the Android software development kit (SDK) and published through Google. (Wikipedia, 2023)

The Google Play Store facilitates the effortless installation of applications on mobile devices with a single click, without requiring any technical expertise. Furthermore, the Play Store provides a user-friendly interface that makes it easier for individuals to search for and discover new applications as compared to desktop alternatives.

## 2.4 Java

Java is a widely used programming language known for its versatility, portability, and scalability. Java is object-oriented, meaning it is based on the concept of objects, which have

properties and methods that can be used to manipulate and interact with data. Java is well-suited for developing mobile applications, as it can be used across multiple platforms and devices. (Wikipedia, 2023)

For this project, Java was chosen as the programming language due to its suitability for developing mobile applications. Additionally, the game engine chosen for the project, libGDX, is written in Java, making it the most suitable choice for the game. Additionally, the author of this thesis wanted to gain further experience working with Java, as it is a widely used language in the development industry.

The use of Java and libGDX allowed for efficient development of the game and ensured that the game would be compatible with multiple platforms, namely desktop personal computers, Android, iOS and HTML, and consistent interface and game size on all possible device screens. Java also offers a vast array of tools and resources for game development, making it a popular choice for developers in the gaming industry.

## 2.5 LibGDX

LibGDX is a robust and versatile cross-platform game development framework that allows developers to write, test, and debug Java-based games on a desktop PC running Windows, Linux, or Mac OS X and deploy that same code to Android, iOS, and WebGL-enabled browsers. This feature-packed framework abstracts away the low-level details, enabling developers to write 2D or 3D games with ease.

LibGDX is an open-source game development framework that is free to use. Its cross-platform support is a key advantage as it supports a wide range of platforms, including Windows, Linux, Mac OS X, Android, iOS, BlackBerry, and web browsers with WebGL support. This cross-platform support enables developers to create games for multiple platforms using a single codebase, thereby reducing development time and costs.

Another advantage of LibGDX is its performance. The framework is designed to be lightweight and efficient, resulting in better performance for games developed using this framework. This enables the games to run smoothly even on budget systems, which is a crucial consideration for mobile gaming.

Furthermore, LibGDX provides a wide range of features, including support for 2D and 3D graphics, audio, input handling, networking, and physics. Additionally, it offers a range of libraries and tools that make it easier to develop games using this framework. In this thesis, the third-party library Box2D was used to make physics environments.

The use of the Java programming language is another advantage of LibGDX. The author of the thesis is more experienced in working with Java, making it a more natural choice than alternatives such as C# in Unity.

In summary, LibGDX was chosen for this thesis for its cross-platform support, efficient performance, wide range of features, and the authors familiarity with the Java programming language.

## 2.6 JSON

JSON stands for JavaScript Object Notation, and it is a lightweight data interchange format that is used to store and transmit data objects consisting of attribute-value pairs and arrays. JSON is often used when data is sent from a server to a web page, and it is a common data format with diverse uses in electronic data interchange, including that of web applications with servers. (Wikipedia, 2023)

In this project, the decision was made to use JSON as the data format for storage because limited capabilities of LibGDX preferences, which cannot be used to store dictionaries or data arrays. To overcome this limitation, the data was stored in a JSON format and then transformed into a string for storage purposes. Later in the application, the string is then reconverted into JSON to access the data. Using JSON for data storage and retrieval is an efficient way to store data in a meaningful way. JSON allows data to be stored in a way that makes sense, making it easy to retrieve and use as needed.

## 2.7 Hiero

Hiero is a bitmap font packing tool from the libGDX framework. It saves fonts in the Angel Code font format, which can be used by BitmapFont in libGDX applications. (libGDX, 2023) This tool is used to create bitmap fonts which are used in libGDX games. Bitmap fonts are created by using images of individual characters or glyphs and packing them into a larger image called a texture atlas.

In this project, Hiero will be used to create custom bitmap font files that can display characters from a wide range of languages.

## 3 RESULT OF THE DEVELOPMENT

### 3.1 Functional Requirements for the application

1. User-defined Dictionary: The game must allow the user to create their own dictionary containing at least three Estonian words and its translation in their native language.
2. Word Activation: The game must randomly select a word from the user-defined dictionary and activate it, causing it to appear on the screen and start moving towards the player.
3. User Interaction: The game must allow the user to click on the active word to bring up three buttons which must contain choices with translation options.
4. Translation Options: The game must provide three translation options for the user to choose from, one of which is the correct translation for the active word, and the other two are randomly selected from other words in the user-defined dictionary.
5. Correct Answer Validation: The game must validate whether the user has chosen the correct translation for the active word and provide feedback accordingly.
6. Game Status System: The game must keep track of the number of lives the user has, with one life lost for each incorrect translation chosen or for allowing the active word to reach the player without being clicked on. When the player finishes words set or loses all lives game should change its status.
7. Win/Lose Conditions: The game must have a win condition when the user has correctly translated all words in their user-defined dictionary, and a lose condition when the user loses all their lives.
8. User Interface: The game must display a player sprite in the center of the screen, three buttons in the bottom part of the screen, seven words in the upper part of the screen, and an interface indicating the number of lives the user currently has. The active word and buttons should only be visible and active when needed.
9. Feedback to User: User should see after each completed game feedback about this session. The feedback should consist of two key elements: the number of mistranslations the user had during the quiz and the amount of time the user spent completing the quiz.

### 3.2 Non-functional requirements for the application

1. Performance: The game should be responsive and have fast load times to ensure a smooth user experience.
2. Usability: The game should have an intuitive and user-friendly interface, with clear instructions and easy navigation.
3. Compatibility: The game should be compatible with a wide range of mobile devices and operating systems to ensure broad accessibility.
4. Reliability: The game should be reliable and free from bugs or glitches that could interrupt the user's learning experience.
5. Scalability: The game should be scalable to able to meet a demand for new features.

### 3.3 Design elements

Design elements of games that facilitate learning can be categorized into cognitive, behavioral, affective, and sociocultural engagement with the subject matter. Cognitive engagement refers to activities that stimulate learners' thinking and problem-solving skills, such as challenges, puzzles, and problem-solving activities. Behavioral engagement refers to activities that motivate learners to engage in learning, such as rewards, feedback, and progress tracking. Affective engagement refers to activities that evoke emotions in learners, such as storylines, characters, and scenarios. Lastly, sociocultural engagement refers to activities that promote social interaction and collaboration, such as multiplayer modes and group activities. (JL Plass, 2015)

The present game utilizes a design element to encourage player motivation, which is closely linked to behavioral engagement. Specifically, feedback has been identified as a critical design element that can motivate learners to engage in learning activities and enhance their performance. The game created for this thesis incorporates feedback mechanisms such as the game timer, personal best timer, and accuracy count to promote player motivation.

The game timer and personal best timers, although not directly related to gameplay, serve as sources of motivation for players to enhance their performance and achieve their personal best records. The inclusion of such design elements can effectively stimulate players' motivation to improve their knowledge and understanding of the set of words presented in the game.

In summary, incorporating feedback design elements into games can be a powerful motivator for players to engage in learning activities and improve their performance. In this

game, the game timer, personal best timer, and accuracy count are utilized to encourage players to strive towards achieving their personal best and enhance their learning outcomes.

### 3.4 Application interface description

The purpose of this section is to provide an analysis of the application interface, including layout, and functionality. This will be achieved through a text description of the interface, accompanied by annotated screenshots to illustrate key features and usability. By providing a comprehensive overview of the application interface, readers will be able to gain a better understanding of the user experience.

#### 3.4.1 Loading screen

The loading screen is an important feature in modern-day applications and games as it provides users with a visual cue that the application is initializing. It appears when the application is first launched and serves the purpose of loading game assets, such as UI skins, fonts, and textures, into the AssetManager for later retrieval.

It is used to preload all necessary game components and to show user that the game is initializing.

#### 3.4.2 Main menu screen

The main menu screen of the application features three primary buttons: 'Start Game', 'Settings', and 'Exit'. Selecting the 'Start Game' button leads the user to the 'Choose Set' screen, which offers the user a options for selecting set of words to start learning, as well as the ability to add set or delete existing sets. The 'Settings' button provides the user with access to the application settings, where they can customize difficulty level. Finally, selecting 'Exit' will close the application.

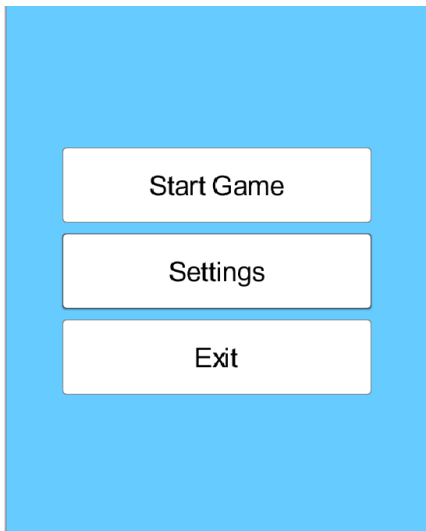


Figure 1 Appearance of the main menu screen (Source: author)

### 3.4.3 Settings screen

The "Settings" screen includes accessibility options that are intended to enhance the gameplay experience. The first setting, "Game Speed," enables the user to control the speed at which the words move towards the center of the screen during gameplay. This feature is particularly beneficial for users with less developed reflexes, as it ensures that they can still engage in the game at a pace that suits their abilities.

The second setting, "Number of Lives," allows the user to set the number of lives they have during the game. This feature caters to two different user preferences: those who prefer a more challenging experience can decrease the number of lives, while those who prefer a more relaxed experience can increase the number of lives and not worry about making mistakes too much.

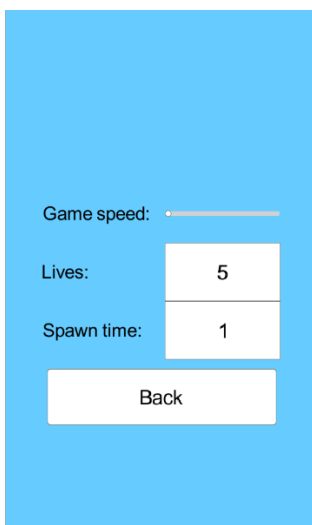


Figure 2 Settings screen appearance (Source: author).

#### 3.4.4 Choose word set screen.

The "Choose Word" screen serves the purpose of allowing users to select the word set that will be utilized in the gameplay. Additionally, the screen provides a tooltip that explains its intended function, thus enhancing the user experience by reducing any potential confusion or uncertainty.

The primary interface of the screen features a set of six dynamic buttons: three of which showcase example word sets, while the other three allow the user to delete word sets with the corresponding delete button.

Additionally, the screen includes a button that takes the user to the 'Add Word Set' screen, where they can create a new word set. A 'Back' button is also provided, allowing the user to navigate back to the main menu screen.




Figure 3 Appearance of the screen for choosing word sets (Source: author).

#### 3.4.5 Dictionary adding screen.

The Dictionary edit screen allows users to add a new set of words to their saved sets of words. The screen contains a text label that informs the user that a minimum of three words is required to create a set, helping to avoid confusion. The "Set name" text field allows users to name their set, which is displayed on the screen where word sets are chosen. Text fields appear after the user presses the "Add Word" button. If at least three words have been added, as shown in Figure 6, the "Save dictionary" button appears, allowing the user to save the current dictionary into preferences.

If there are not enough words added, the "Save dictionary" button does not appear, until there are enough words added, as shown in figure 5.



Minimum is three words

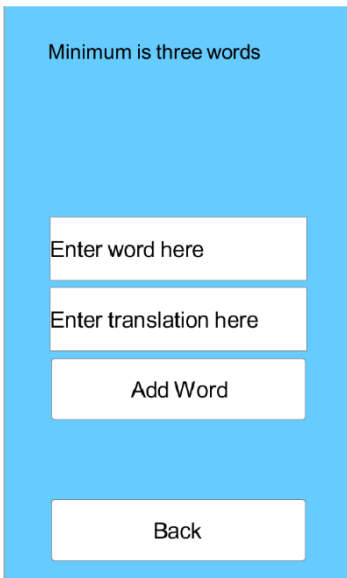
Enter set name here

Save set name

Back

This screenshot shows a mobile application screen for creating a set. The background is light blue. At the top, the text "Minimum is three words" is displayed. Below this is a white text input field containing the placeholder text "Enter set name here". Underneath the input field is a white button with the text "Save set name". At the bottom of the screen is another white button with the text "Back".

Figure 4 Set creation screen.



Minimum is three words

Enter word here

Enter translation here

Add Word

Back

This screenshot shows the same mobile application screen as Figure 4, but with different input fields. The text "Minimum is three words" remains at the top. Below it are two stacked white text input fields. The first contains the placeholder text "Enter word here" and the second contains "Enter translation here". Below these two fields is a white button with the text "Add Word". At the bottom of the screen is a white button with the text "Back".

Figure 5 Set creation screen when there is less than three words added (Source: author).

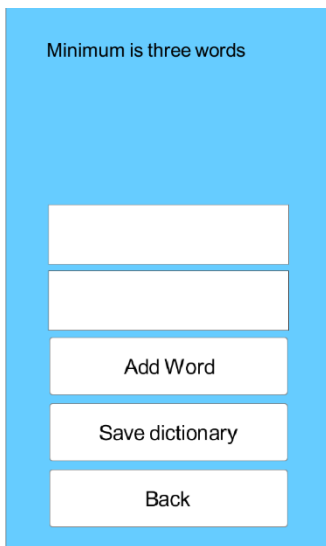


Figure 6 Set creation screen when there are enough words added (Source: author).

### 3.4.6 Game screen

The gameplay of the game starts with the player selecting a word set on the "Choose Word Set" screen. Once the player has made their selection, the core gameplay loop begins. Words will randomly appear from one of five positions on the screen, as shown in Figure 7. The player's objective is to press on each word and choose the right translation for word, before it collides with the blue circle representing the player. If a word collides with the player, the player's lives counter will decrease. If it reaches zero, the game is over and the game over screen will appear.

Once the player selects a word, three buttons will appear on the screen, with two of the buttons showing random word translations from the current word set and the other button showing the correct translation for the chosen word. The player must select the button with the correct translation to destroy the word. If the player chooses the wrong translation, the words count will decrease and the word will be destroyed. The player can't choose a new word until the current word is destroyed either by selecting the correct answer or by reaching the player and getting destroyed on collision.

The goal of the game is to choose the correct translations for all the words in the chosen word set without losing all the lives and to try attempting so as quickly as possible. To help achieve this goal, players can customize the gameplay speed and number of lives based on their preferences.

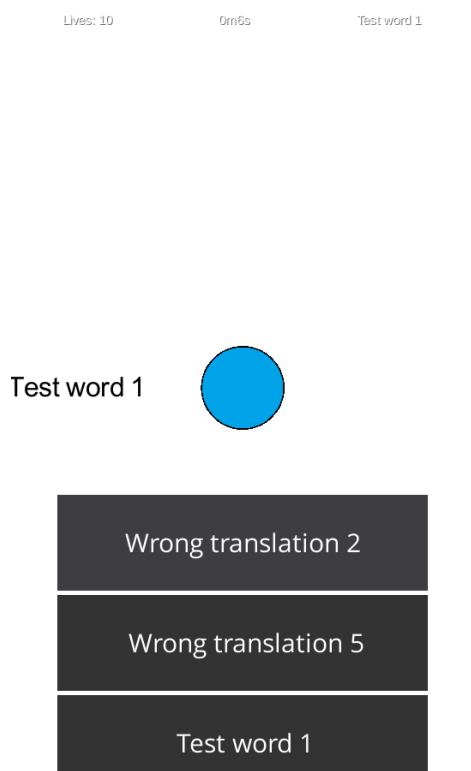


Figure 7 Gameplay example when word is chosen (Source: author).

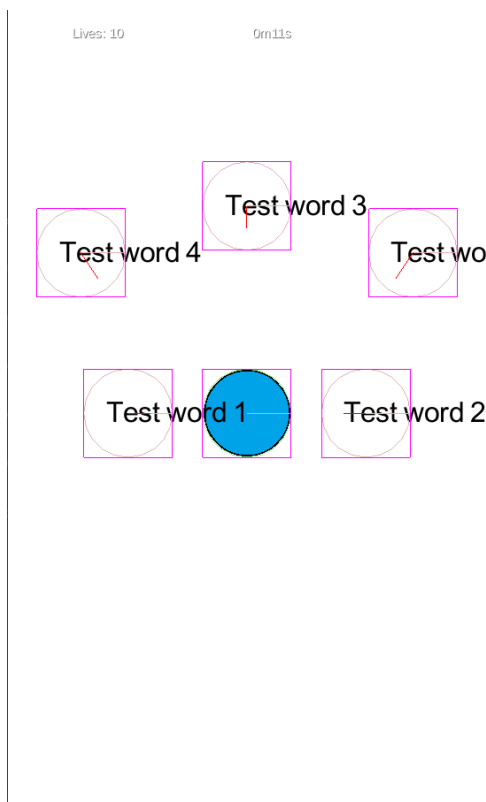


Figure 8 All possible word directions, debug lines to better indicate where words are in physics (Source: author).

### 3.4.7 Game over screen

The Game over screen appears when a player finishes the current set or loses all their lives. Its purpose is to indicate that the game is over and provide feedback to the user, which includes the game time, set name, and number of lives left as shown in figure 9.

The screen begins with a label showing the name of the completed set. The second label displays how many lives the user has left at the end of the set. If the number of lives is zero, it indicates that the word set was failed.

The third label displays the amount of time spent completing the set. If the time is lower than the best recorded time for the set, it changes to inform the user that they have beaten their personal best record, as shown in figure 10.

The "Play again" button starts the current set again, while the "Menu" button sends the user back to the main menu.

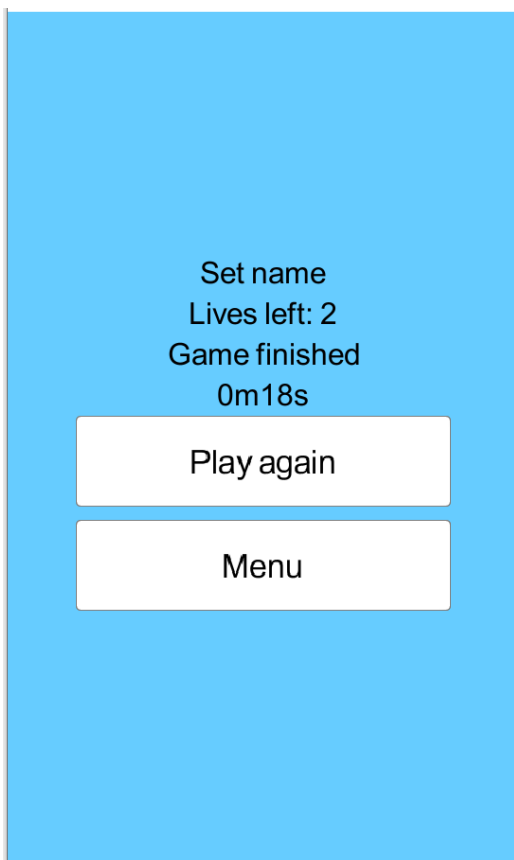
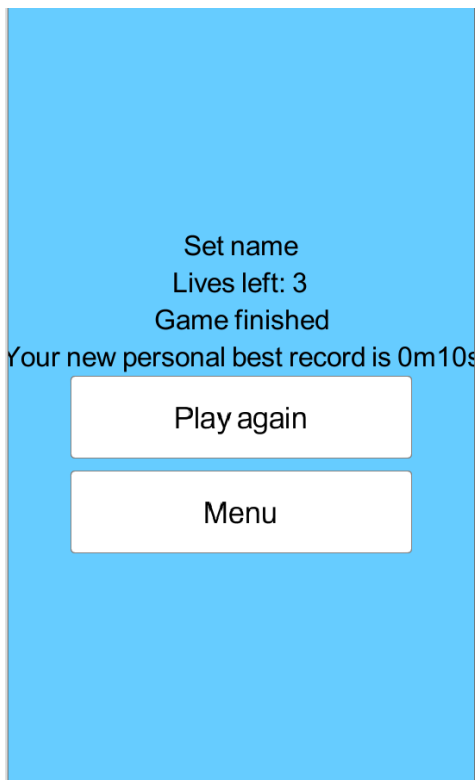


Figure 9 Game over screen (Source: author).



*Figure 10 Game over screen with new personal best record (Source: author).*

### 3.5 LearningGame

In the context of programming, an orchestrator can be thought of as a design pattern that helps to coordinate the interaction between different software components or services. The role of an orchestrator is to ensure that all the different parts of a system work together in a seamless and efficient manner.

In the context of game development, an orchestrator can be used to manage the different game objects and their interactions with each other. This can include managing the game logic, handling user input, and coordinating the rendering of the game world.

In this project, LearningGame serves as an example of the orchestrator design pattern. The class extends the Game class, which enables it to use the setScreen method for switching between screens. Other screens, in turn, use the changeScreen method provided by the LearningGame class for this purpose.

In addition to its role in screen management, LearningGame is also responsible for loading the AssetManager class and providing a single instance of it to other classes. This ensures consistent management of resources throughout the application.

## 3.6 Screens

Screens are used in game development to separate different sections or states of a game. Each screen represents a different state of the game, with its own logic, graphics, and user interface. This game is separated into LoadingScreen, MenuScreen, PreferencesScreen, ChooseSetScreen, EditSetScreen, GameScreen and GameOverScreen.

Screens provide a modular approach to game development, allowing developers to create and manage different sections of a game independently, which makes it easier to add new features or modify existing ones without affecting other parts of the game.

In addition, screens can improve the performance and efficiency of a game by allowing it to render only what is necessary for each screen.

The Screen interface in LibGDX provides a set of methods that need to be implemented by any class that implements this interface. These methods are responsible for initializing and updating the screen, handling user input, and rendering graphics on the screen. Here is an explanation of all the methods in the Screen interface:

`show()`: This method is called when the screen becomes the current screen. It is typically used to initialize the screen and its components, such as loading resources, setting up the game camera, and starting music or sound effects.

`render(float delta)`: This method is called every frame to update and render the screen. The parameter delta is the time elapsed since the last frame in seconds, which can be used to make the screen update at a consistent rate, regardless of the frame rate.

`resize(int width, int height)`: This method is called when the screen size changes. It is typically used to adjust the viewport to match the new screen size, which is necessary for correct rendering on different devices.

`pause()`: This method is called when the game is paused, such as when the player presses the home button on a mobile device. It is typically used to pause any ongoing activities, such as animations or sound effects.

`resume()`: This method is called when the game is resumed after being paused. It is typically used to resume any ongoing activities, such as animations or sound effects.

`hide()`: This method is called when the screen is no longer the current screen. It is typically used to clean up any resources used by the screen, such as stopping music or sound effects and disposing of any loaded resources.

`dispose()`: This method is called when the screen is no longer needed and is being disposed of. It is typically used to dispose of any loaded resources that were not cleaned up in the `hide()` method, such as textures or sound effects.

By implementing these methods, a class that implements the `Screen` interface can handle all the necessary tasks for a game screen, such as updating game logic, rendering graphics, and handling user input. This provides a flexible and efficient way to create and manage multiple screens in a game.

### 3.7 Camera

In LibGDX, a `Camera` is an object that determines what part of the game world is visible on the screen. It defines the position, orientation, and size of the viewport, which is the area of the game world that is displayed on the screen. The `Camera` is responsible for transforming the world coordinates into screen coordinates, which are used by the graphics rendering system to draw the game objects.

There are several types of `Cameras` in LibGDX, including `OrthographicCamera`, `PerspectiveCamera`, and `OrthographicCameraWithViewport`. The `OrthographicCamera` is the most used camera in 2D games, while the `PerspectiveCamera` is used for 3D games. In this project used `OrthographicCamera`, as the most used one for 2D applications.

The `Camera` has several properties that can be modified to control its behavior, such as its position, rotation, and zoom level. The `Camera` can also be moved and rotated in real-time to follow the player or other game objects. In this game, the camera remains static as there is no requirement to move it.

### 3.8 Viewport

`Viewport` is an object that defines the area of the game world that is visible on the screen. It specifies the width and height of the screen in pixels and the scale of the game world relative to the screen size. The `Viewport` is responsible for scaling and positioning the game world to fit within the screen while maintaining its aspect ratio. It is used in conjunction with the `Camera` to determine what part of the game world is visible on the screen. The `Camera` defines the position and orientation of the viewport, while the `Viewport` scales and positions the game world to fit within the viewport. There are several types of viewports in LibGDX and the most common are `FitViewport`, `FillViewport`, `ExtendViewport`, `ScreenViewport` and `StretchViewport`.

In this application are used two types of viewports: `ScreenViewport` for menu and `ExtendViewport` for gameplay.

`ScreenViewport` - This viewport maps the game world directly to the screen, with no scaling or positioning. This can be useful and commonly used for creating fixed-size UI elements or other static visual elements.

`ExtendViewport` - This viewport extends the game world to fill the entire screen but maintains the aspect ratio of the game world. This means that the entire screen is filled with the game world, but some parts of the game world may be off-screen for some users, depending on the screen size. When resizing the game window on desktop versions, the `ExtendViewport` becomes particularly noticeable as it extends the visible portion of the game world to fit the new screen size.

### 3.9 HUD

The top side of the screen contains several elements of the HUD, including the lives meter, time meter, and currently selected word. Buttons that appear after a player chooses a word are not considered part of the HUD. These buttons are rendered in the `GameScreen` class and are updated in the render method.

Hud statistics include the player's performance tracking, like number of remaining lives and amount of time passed from the start. The stats are displayed in real-time on the screen, allowing the player to keep track of their progress while in the game.

### 3.10 Stage

In LibGDX, a stage is a container that holds and manages the graphical elements of a game's user interface. It is used for easy and efficient handling of user input events like touch events and key inputs. The stage is used for arranging the UI elements, such as buttons, labels, and images, on the screen without assigning them positions in coordinates. It is useful because it provides more consistent experience across the different screen sizes, and it is also easier to place screen elements in relative to each other positions.

### 3.11 SpriteBatch

The `SpriteBatch` class is a commonly used tool in LibGDX for rendering 2D graphics onto the screen. It is an essential component in game development as it provides a simple and efficient way to render sprites and textures. This class works by grouping multiple sprites

or textures together and rendering them in a single draw call. This significantly improves performance compared to rendering each sprite individually.

To use SpriteBatch, defined properties for each sprite such as its position and scale. The SpriteBatch class is often used in combination with a camera to render graphics onto the screen.

### 3.12 Asset Manager

The Asset Manager is used for loading and retrieving various in-game assets, such as sounds, skins, and textures. During the Loading Screen, the Asset Manager queues up the loading process for these assets, and later, retrieves them for use in the game. This system serves two essential purposes: convenience and optimization.

By using the Asset Manager, it is easier to retrieve the assets as necessary. Additionally, the Asset Manager helps optimize the game's performance by preloading necessary assets, reducing the game's load time. It is also more convenient to manage all file locations from one place.

### 3.13 Font

To ensure that Estonian and Russian characters are displayed properly, was made the decision to use a custom font rather than the built-in LibGDX standard Bitmap Font. The Bitmap Font only supports English characters, and attempting to display Estonian or Russian characters would result in an empty rectangle showing that this character does not exist in font. To avoid this issue, Font was created by using tool Hiero provided by LibGDX and then it was set up as the project font.

### 3.14 AppPreferences

App preferences serve to save user settings between gaming sessions. In this game, the user can personalize the number of lives and game speed from the preferences screen. Additionally, users can store their custom word sets.

### 3.15 Box2d World

Box2D world is a 2D physics engine that simulates interactions between objects, enabling the creation of gameplay mechanics, such as collision detection, gravity, and friction. In a game, the world manages physical objects, such as bodies, fixtures, and joints, which

respond to user input and other game events. The Box2D world's behavior is determined by a set of properties, such as gravity, sleep settings, and time step, that allow for realistic physics simulations.

### 3.16 B2BodyFactory

B2Body factory is an example of factory pattern class. Factory pattern is a creational design pattern that provides a way of creating objects without exposing the instantiation logic. The main advantage of using this design pattern is that it allows to distance object creation logic from creating objects. That means that the main benefit of that it provides a way to separate the process of creating objects from the logic that governs how those objects are created. This means that the code that needs to create objects can do so without worrying about the specific details of how the objects are created, making the code more modular, flexible, and easier to maintain. The class is implemented using the singleton pattern to ensure stable access across all classes and optimize resource usage, as it is not required to be instantiated multiple times.

To create a body in the Libgdx factory class, you first need to define the body settings using a BodyDef. This includes specifying the body type (static, dynamic, or kinematic), as well as the position coordinates (x and y) and whether the body can be rotated. Once the BodyDef is defined, you can create an actual body in the world, which serves as an abstract representation of the body.

Next, you create a fixture, which represents the physical properties of the body and can be collided with. Then body factory returns the body, which is then assigned to the entity.

### 3.17 Contact listener

In LibGDX, a ContactListener is an interface that is used to detect collisions between two objects in a Box2D world. When two objects collide, Box2D generates a contact between them. The ContactListener listens for these contacts and allows the programmer to respond to them in a customized way.

It has several override methods, but in current project used only beginContact, which triggers when two fixtures collided.

In Box2D, a fixture is a shape of body, and it also has material properties such as density, friction, and restitution. It is used for collision detection.

### 3.18 Entities custom classes

In Java, an abstract class is a class that cannot be instantiated directly and serves as a template for other classes. An abstract class can be used to define a set of methods and properties that must be implemented by any concrete class that extends it. When a class extends an abstract class, it must implement all the abstract methods defined in the abstract class.

It is used to define common behavior for classes Player and Word. It is necessary for calling an update and draw methods for all entities while looping through single array with B2dBodyEntity. It is also used to define common properties body, velocity, position, world and if entity is active.

Player class is used to define player, draw player, and track number of lives player has.

Word class is used to draw, define, and control each word. Each word can be activated and deactivated. When word is active it has assigned word, and it is moving towards the player. When word is inactive it is outside the visible screen space and does not move. Word activation is used by GameScreen class, which after a set time period activates random inactive word object from the array of all word objects and assigns a random word from the currently selected word set. After the word is destroyed, it becomes inactive again, until GameScreen activates it again with another word.

ChooseButton is different enough from Player and Word classes, that it does not extend abstract class B2dBodyEntity. Instead, it inherits from the TextButton standard libGDX class, which offers a range of functionality, including the ability to set itself as invisible until the player clicks the word.

When the player selects a word, the GameScreen class activates three buttons and assigns each one of them a value. Two of these values are randomly assigned values from the selected word set, while the third corresponds to the translation of the chosen word. The class also includes a method that checks whether pressed button assigned value corresponds to the currently selected word, returning a Boolean value of true if there is a match, and false if there is not.

## CONCLUSION

This diploma project aimed to explore the development of a gamified mobile application designed to improve the learning experience for people studying Estonian vocabulary by contributing to the field of language learning tools.

Based on previous research, it was hypothesized that the gamified mobile application would lead to an improvement in learning outcomes when compared to a non-gamified application and by incorporating gamification elements into the learning process, developed game aimed to give a tool which would help to improve learning outcomes for its users.

The project began with an exploration of the market for similar solutions and the selection of appropriate technologies for the game's development. Then market analysis of existing solutions was finished and described in section 1. After this research it was decided, that according to results described in section 1, application will have enough difference to have its own niche userbase, and that it is justified to develop this application.

After that selection of technologies used in this diploma and their usage rationale were described in section 2.

The main task of developing a video game was completed and described in section 3. In this section was also described finished product.

In the future there are existing plans to improve application interface for the purpose of making it more comprehensible for the user, enhance the game's visual appeal for application to look more marketable, and provide tutorial prompts to help players avoid confusion at the start of the game.

## RESÜMEE

Käesoleva diplomiprojekti eesmärk oli uurida mängulise mobiilirakenduse arendamist, mille eesmärk on parandada eesti keele sõnavara õppimise kogemust, andes oma panuse keeleõppe vahendite valdkonda.

Tuginedes varasematele uuringutele, hüpoteesiti, et mänguline mobiilirakendus viib õpitulemuste paranemiseni võrreldes mitte-mängulise rakendusega, ning mängulisuse elementide lisamisega õppeprotsessi, välja töötatud mängu eesmärk oli anda vahend, mis aitaks parandada õpitulemusi selle kasutajate jaoks.

Projekt algas sarnaste lahenduste turu uurimisega ja mängu arendamiseks sobivate tehnoloogiate valimisega. Seejärel viidi lõpule olemasolevate lahenduste turuanalüüs, mida kirjeldati 1. osas. Pärast seda uuringut otsustati, et vastavalt punktis 1 kirjeldatud tulemustele on rakendus piisavalt erinev, et omada oma nišši kasutajaskonda, ning et selle rakenduse arendamine on põhjendatud.

Pärast seda kirjeldati käesolevas diplomitöös kasutatavate tehnoloogiate valikut ja nende kasutamise põhjendusi punktis 2.

Peamine ülesanne, videomängu arendamine, viidi lõpule ja seda kirjeldati punktis 3. Selles osas kirjeldati ka valmis toodet.

Tulevikus on olemasolevad plaanid täiustada rakenduse kasutajaliidest, et muuta see kasutajale arusaadavamaks, parandada mängu visuaalset külge, et rakendus näeks välja turukõlblikum ja pakkuda juhendmaterjali, mis aitaks mängijatel vältida segadust mängu alguses.

## REFERENCES

- Alsawaier, Raed. (2017). The Effect of Gamification on Motivation and Engagement. *International Journal of Information and Learning Technology*. 35. 00-00. 10.1108/IJILT-02-2017-0009.
- Papastergiou, Marina. (2009). Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation. *Computers & Education*. 52. 1-12. 10.1016/j.compedu.2008.06.004.
- Sailer, Michael & Homner, Lisa. (2020). The Gamification of Learning: a Meta-analysis. *Educational Psychology Review*. 32. 77-112. 10.1007/s10648-019-09498-w.
- Shufang Wang, William Littlewood, Exploring students' demotivation and remotivation in learning English, *System*, Volume 103, 2021, 102617, ISSN 0346-251X,
- Plass, Jan & Homer, Bruce & Kinzer, Charles. (2015). Foundations of Game-Based Learning. *Educational Psychologist*. 50. 258-283. 10.1080/00461520.2015.1122533.
- Wikipedia 2023. Android (operating system). Available at [https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)), accessed April 23, 2023.
- Wikipedia 2023. Google Play. Available at [https://en.wikipedia.org/wiki/Google\\_Play](https://en.wikipedia.org/wiki/Google_Play), accessed April 23, 2023.
- Wikipedia 2023. Java (programming language). Available at [https://en.wikipedia.org/wiki/Java\\_%28programming\\_language%29](https://en.wikipedia.org/wiki/Java_%28programming_language%29), accessed April 23, 2023.
- Wikipedia 2023. JSON. Available at <https://en.wikipedia.org/wiki/JSON>, accessed April 23, 2023.
- Android Studio 2023. Meet Android Studio. Available at <https://developer.android.com/studio/intro/>, accessed April 23, 2023.
- libGDX 2023. libGDX is a cross-platform Java game development framework based on OpenGL (ES) that works on Windows, Linux, macOS, Android, your browser and iOS. Available at <https://libgdx.com/>, accessed April 23, 2023.

libGDX 2023. Hiero. Available at <https://libgdx.com/wiki/tools/hiero> , accessed April 23, 2023.

libGDX 2023. Skin. Available at <https://libgdx.com/wiki/graphics/2d/scene2d/skin> , accessed April 23, 2023.

Drops Languages 2023. Drops: Изучайте эстонский язык. Available at <https://play.google.com/store/apps/details?id=com.languagedrops.drops.learn.learning.speak.language.estonian.words&hl=ru&gl=US> , accessed April 23, 2023.

ASI Edu 2021. Учить эстонский. Available at <https://play.google.com/store/apps/details?id=asi.education.language.learnestonian> , accessed April 23, 2023.

## Appendix 1

GitHub link to the source code:

<https://github.com/makardr/LearningGame>