

Article

Gamification as a Supportive Tool for School Children with Dyslexia

Paweł Dymora  and Karol Niemiec

Faculty of Electrical and Computer Engineering, Rzeszów University of Technology, al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland; irocket94@gmail.com

* Correspondence: Pawel.Dymora@prz.edu.pl

Received: 19 August 2019; Accepted: 30 October 2019; Published: 1 November 2019



Abstract: Gamification, in its nature, combines not only games but also the whole psychological environment. Thanks to this, a properly prepared implementation of gameplaying can encourage people to compete with others and achieve the set tasks and goals. A person feels fulfilled that through his actions has performed a mission or reached a new level. It stimulates them to continue their activity and self-improvement to be better and beat their records. Its advantage is also that it does not have to be limited to one technology or method—it can be realized both through a simple scenario and a corkboard with results, it can also be embedded, e.g., in a virtual or augmented reality. This article focuses on the gamification of dyslexia, a common disorder of developmental disorders among pupils. It affects about 10%–15% of school-age children. The research narrowed the field of the study to one of the aspects of developmental dyslexia—dysorthography and making spelling mistakes by people affected by this disorder. This work aims to present an original application which is using gamification as a supportive tool for the learning of school children with diagnosed dyslexia. The conducted study was based on the implementation of original algorithms and scenarios of gamification on mobile devices, especially smartphones. School children are following a gamification approach for a specified period. As a conclusion, it can be stated that the proposed framework and gamification can help in the learning of people with dyslexia.

Keywords: gamification in learning; playful experiences; smartphone applications; immersive learning; dyslexia

1. Introduction

Gamification refers to the use of mechanisms and schemes known from role-playing (RPG) or computer games to increase the involvement and mobilization of participants. According to the Lexico dictionary (which is an extension of the Oxford University Press dictionaries), gamification is “*the application of typical game elements (e.g., scoring, rivalry with others, rules of the game) to other areas of activity, typically as an online marketing technique to encourage involvement in a product or service*” [1].

Authors in [2] define the concept of gamification as “*the use of game design elements in non-game contexts*.” This definition is widely used by many researchers, i.e., [3–5]. In [3], authors stipulate that gamification may be defined as “*the use of design (rather than game-based technology or other game-related practices) elements (rather than fully developed games) characteristic for games (rather than play or playfulness) in non-game contexts (regardless of specific usage intentions, contexts, or implementation media)*.” Gamification, when used in the educational process, can be defined as gamified learning [4]. Gamified learning is similar to the game-based learning concept. Authors in [2,4] define game-based learning as an approach that implies the design of real, autonomous games, whereas the gamified learning is a learning process that concentrates on extending or changing an established learning process in order to create an improved version of the learning process that users

experience as being game-like. In our work, we focus on gamification that is not a commercial product but in the context of learning where we design educational processes and add some game elements in order to enhance existing learning processes.

Although, at first glance, it may seem that games are a rather new phenomenon; they have been accompanying people since the beginning of time. The first board games found by archaeologists came from ancient Mesopotamia; in ancient times, people also played with dice—Claudius, the Roman Emperor (10 BC–54 AD) wrote a guide for players. Gamification was widely disseminated in 2010, although it is not a new phenomenon. It transfers motivational techniques used by game designers and producers for many years to other fields, such as business, marketing, or education. As mentioned earlier, gamification focuses on the use of a game mechanic—such as a badge system, points, scoreboard, or levels. Thanks to this, gamification participants start to perceive the tasks they face in the real world as challenges faced by the game hero. Participants tend to change their behavior to achieve their goals by competing with others or themselves. Such a change of behavior makes the user consider certain activities to be a play, a game, or a mission leading to victory, and not a typical job or an unpleasant duty [2,3,6].

Developmental dyslexia is a specific difficulty in writing and reading for people with healthy mental development. According to the International Statistical Classification of Diseases and Health Problems ICD-10, dyslexia is classified as F81 (specific skills developmental disorder). It is estimated that about 10%–15% of school children report dyslexia, 4% of whom are cases of very severe difficulties that can be described as deep dyslexia (often referred to as reading disorders delayed by 3–4 years or very severe typing difficulties, not only spelling but also speech development). People with dyslexia have difficulty in distinguishing sounds from words, as well as in memorizing and merging single sounds into words. They find it difficult to remember letters and their equivalents in the form of sounds and to read aloud, especially at the beginning of learning when decoding the written word by letter. It also leads to problems with understanding the text or remembering names and surnames, and even more so with remembering them. Children with dyslexia, despite years of learning, sometimes write the word correctly, sometimes not—the pattern is not fixed. If the disorder is not significant, fun and exercise will help to remove the problem. However, a neglected problem will grow with the child. The results one gets depends on the extent of the disorder and how many spheres one has deficits in. To be effective, pedagogical therapy must last 3–4 years and should start from an early age [7–9].

Although gamification in some areas is a highly advanced practice, in others, it is only beginning to pave the way. Such a field is medicine, in which the implementation of gamification is more complicated—for example, because of the broad spectrum of applications and the need to focus on a specific group of recipients. This does not mean, however, that it is not used—an example can be the Kids Pain Squad application designed by the Canadian agency Cundari. It addresses children and adolescents with cancer. It aims to improve treatment by motivating patients to inform doctors about the place and intensity of pain. The application realizes this by joining a fictitious police pain management unit. Its main task is to inform about any discomfort through styled police reports. Regularity and accuracy are rewarded with a promotion to higher grades and motivational films recorded by actors of crime series [10,11].

Another example of the use of gameplay in medicine is the Foldit project, in which players have the task of modifying the presented proteins in such a way as to obtain as many points as possible (the most energy-efficient protein models). Thanks to this, within ten days from the start of the application, the users were able to create a virtual model of protease responsible for the multiplication of the HIV retrovirus, the cause of AIDS, which was not achieved by the scientists for many years [10–12].

In [13], the authors showed that rhythmic reading training (RRT)—introducing rhythm into dyslexia therapy—is able to help develop reading skills among children with dyslexia. RRT improved the reading speed of children participating in the study, but also their understanding of the text they read. The authors in [14] proposed the use of the classDojo platform to motivate dyslexic students. They showed different approaches to the use of motivational badges, where one of them assumed a top-down

badge for achieved goals, the other one focused on the customization of individual badges, for a particular student. The results showed that in the second approach, the pupils were more motivated and offered their own badges for achieving the goal.

However, the authors in [15,16], explored seven games elements (story, goals, levels, points, awards, achievements, and opinions) as the basis for the application to play dyslexia. They developed the LexiPal application, which they introduced to a group of children aged 5–8, and then asked them two questions—whether they liked it and whether they would like to use it again. This showed that all the respondents liked the application, and the vast majority agreed to use it again.

Children quickly lose interest, get bored, and repetitive activities discourage them. However, an attractive presentation of repetitive actions in the form of competition, raises their achievements in classification, i.e., may break through resistance in performing these tasks. Besides, in some medical conditions, exercise can cause physical pain. The distraction of thoughts and concentration, immersion in the scenario of a game or competition allows them to achieve successful results in the medical treatment of children and youth. In particular, modern information technologies, including advanced graphics engines, virtual/augmented reality, availability of efficient small computer devices such as smartphones and tablets, can significantly help to achieve this goal [17–19]. Assembling all these components and offering an exciting gameplay and competition scenario can improve the effects of many medical and other treatments. In [20–22], researchers have shown increased effects on people with dyslexia through the use of video games. Video games can improve letter-to-speech sound mapping (phonological decoding) and, consequently, reading abilities. Some researchers have focused on identifying relationships between game characteristics and psychological outcomes (e.g., motivation) [23,24].

The literature mentions gamification in the context of benefits, but as is often in life, also some negative impacts can be expected. Authors in [10,14,19] present some adverse effects associated with gamification in educational contexts and aimed at understanding the outcomes and reports in the studies. Generally, they identified a few negative effects, e.g., *indifference*, *loss of performance*, *undesired behavior*, and *declining effects*. Also, the study found that gamification had a negative effect on users' enjoyment and flow experience. Furthermore, the use of gamification was found to have a negative indirect effect on users' loyalty. It is important that these insights are examined and taken into account when designing applications supporting the learning process, especially in the field of special education.

Most of the research conducted so far has narrowly presented specific programs that can be used in the teaching process and has focused more on confirming the impact of gamification on the student. In [2,5,6,10,11,14,16–19], the authors discussed corresponding presented results of their research on the use of gamification in motivating dyslexic students. They presented different motivation strategies for use in existing applications with the design of the implementation for gamification in the education of dyslexia. Unlike this, from the beginning our solution is targeted at dyslexic students and so implies gamification mechanisms to maximize effectiveness in this target group.

In the paper, we proposed an application supporting the learning process of school children with diagnosed spelling difficulties, affected by developmental dyslexia. As our contribution, it should be considered the development of the original mobile application and implementation of original algorithms and gamification mechanisms to support the learning process. The application using gamification mechanisms was designed to increase the attractiveness of the tools in the context of immersive learning. In the proposed application, users can practice not only the correct spelling or orthography of words but also check their knowledge in tests or use it to achieve the highest possible result in different play modes. Also, they can perform tasks for which they receive items that make the game easier to play and help them gain a high position in the ranking. Preliminary studies were also carried out to assess the impact of the development of the application on student performance. These studies were carried out on the original system based on the www platform. Preliminary research allowed to obtain feedback on the application in order to evaluate it and prepare corrections

before researching a broader group. Pupils followed the path of gamification for a specified period. In conclusion, it can be stated that the applied approach to gamification helps in the learning of school children with dyslexia.

2. The Proposition of the Mobile Application for Dyslexia Gamification

In this paper, we present an application for mobile devices to implement gamification mechanisms to improve the effects of therapy for dyslexia school children. The application supports learning the rules of spelling and orthography, divided into different departments through the use of gamification mechanisms, i.e., the use of mechanisms of competition and gameplay known, e.g., from video games or RPG. The application implements such elements as different word bases, exciting game scenarios, character building, and unlocking equipment prizes, testing, and the current ranking of achievements motivating players. An important part is a separate web platform for testing the level of participants, which tracks achievements and compares them with the initial level achieved after a specified period of use of the application. According to [15,16], and presented in the paper concept, in our application, we adopted some most motivating elements, i.e., goals, levels, points, achievements, feedback from authors. The main characteristics of the modules are as follows:

Goal—choose the correct one of the two available buttons that complement the word displayed above and gain all achievements. It is assumed that providing a goal in gameplay will result in greater involvement among users.

Levels—application has different modes, e.g., one is time-limited, the next one with an infinite number of attempts, etc. The main goal of the levels is to encourage users to continue using (playing) the application and not get bored with it.

Points—when the user chooses all of the correct answers, the application gives points and achievements. Points are the main element of motivating users to get the best results.

Achievements—are optional avatar additions that are intended to encourage the user to improve the results.

In the adopted application logical model, only the implemented gameplay elements, such as levels, achievements, and points, can affect the user.

The application operates under the control of the operating system, Google Android. The second component of the platform, used for carrying out checking dictations, is implemented in the form of an interactive website. In order to ensure a balance between application functionality and compatibility with as many devices as possible, it was decided that the minimum supported version of the system would be 5.0 Lollipop. The application was written in Java, using IDE (integrated programming environment) from the Android Studio. The testing platform is based on a website with an integrated JavaScript and a code written in PHP, which is a popular scripting programming language suitable for web application development. Additionally, a MySQL database was implemented for data storage.

Figure 1 shows the application schema. The client (the researched person) had access to two devices—a computer that is connected to the website on the server via the Internet. The second client device was a smartphone with the Android system, which through the application, connected with both the database and the server to download additional files.

Application screens can be represented as the following tree structure, shown in Figure 2. Below the name of each screen, in parentheses, the programmed class name is shown, which corresponds to a particular screen. Each of them is supported by a dedicated class, except for the screens: WHOLE SPELLING, ONE SPELLING and BASIC, EXPANDED, FULL, because the classes assigned to them are parameterized—the content on them is automatically adjusted based on the screen selection. Thanks to this, it was possible to optimize the application by reducing the amount of code needed for proper operation.

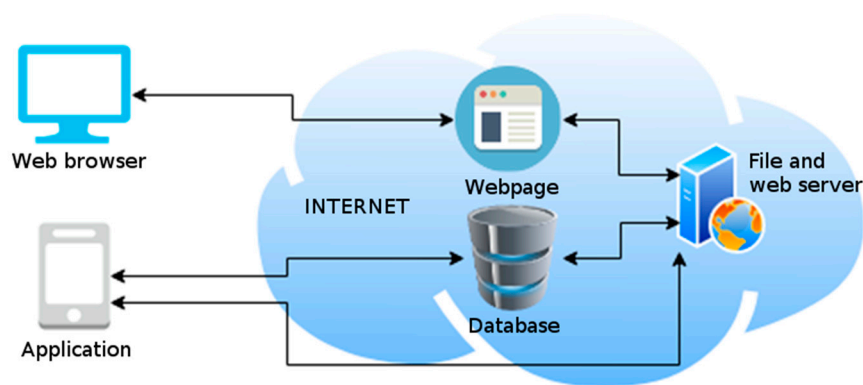


Figure 1. Application schema diagram.

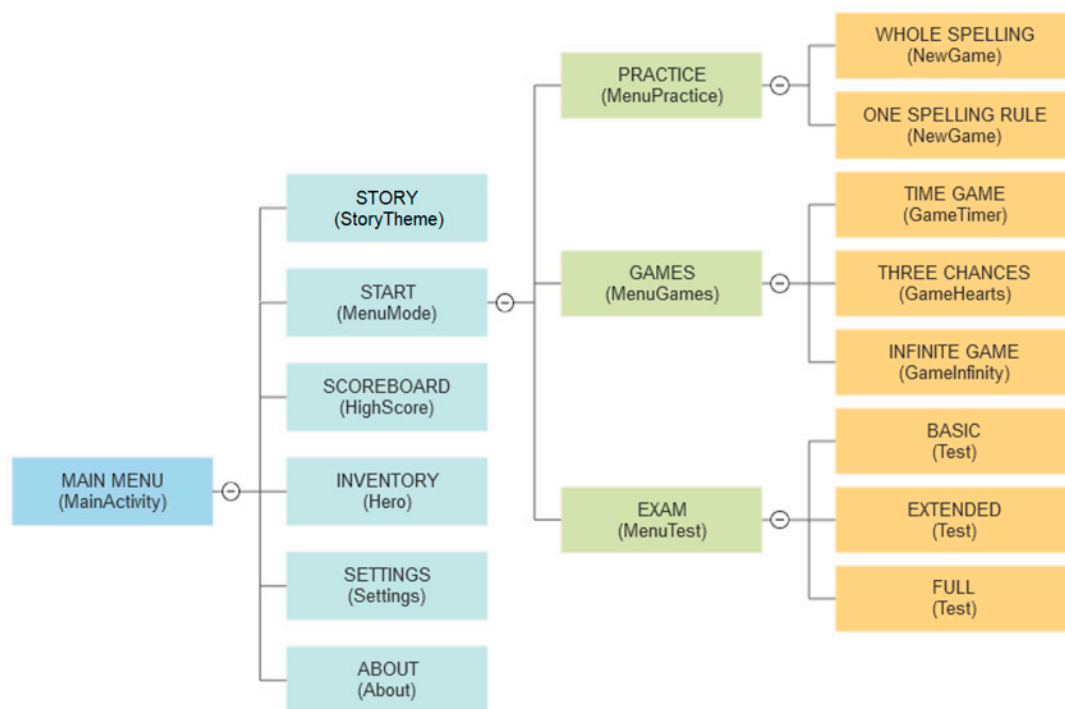


Figure 2. Application screen structure.

At the application startup, an instance of the MainActivity class is created, and the onCreate method is called automatically. In addition to integration with the screen layout, its task is to check whether there is an Internet connection (using the isNetworkAvailable method), verification of permissions granted to the application, downloading the necessary files (download method and internal class DownloadData), carrying out the user login process (login method), and finally, displaying and operating the main menu. This is done by displaying appropriate windows and messages. After logging in correctly, the user is asked to choose a character. The Hero class is started, and a modified inventory screen is displayed. The onCreate method includes a condition that checks if the character has been selected. If not, the CharacterSelection method is called. When the user selects a character, the internal SetCharacter class is started and inserts the data into the database. During subsequent application launches, the onCreate method of MainActivity class checks again the Internet connection, permissions, downloaded files, and logged the user. If any of the above items have not been correctly configured, the configuration is attempted again. If everything is successful, a message will be displayed, and when one closes the message, the main menu is displayed. The following sections describe the basic functionalities of each mobile application screen and related software components as well as classes and main methods.

2.1. STORY Screen

The STORY screen presents a story that is the main theme of the gameplay in which the user is asked to participate. As the user progresses in the game, further stages of the story are revealed. Progress in the story timeline depends on the achievements. The window contains a progress counter, text screen, and back button. The story of the game is based on the adventure of the knight Bolesław, who is supposed to defeat the dragon and save his castle. Each new equipment (see Section 2.7) increases story progress by 1/12.

2.2. MAIN MENU Screen

The MainActivity class is responsible for displaying and operating the MAIN MENU screen of the application. After operations related to application configuration, it is responsible for displaying the application's main menu, which consists of the following buttons:

- START—allows one to choose the game mode,
- SCOREBOARD—shows the total scores of all players,
- INVENTORY—will enable one to preview and select the player's equipment,
- SETTINGS—displays the menu of application parameter settings,
- ABOUT—shows essential information about the program,
- EXIT—closes the application.

2.3. START Screen

The START menu operation is performed by the MenuMode class. After starting the onCreate method, the checkFiles method is called, which verifies the correctness of the data files downloaded. In case of a problem, it informs the user about it and forces the user to return to the MAIN MENU screen. The START menu is a gateway to the main functionality of the application—supporting spelling learning. It allows to select one of the three modes of gameplay:

PRACTICE—aimed at improving skills,
GAMES—whose task is to compete with others,
EXAM—aimed at checking the acquired skills.

Selecting one of the buttons calls up the appropriate method: MenuPractice, MenuGames, MenuTest—responsible for moving to the next screen. Pressing the RETURN button calls the onBackPressed method inherited from the main class.

2.4. PRACTICE Screen

MenuPractice class is responsible for displaying and correct operation of the PRACTICE screen. Thanks to this functionality, it is possible to improve spelling skills in one of the two available modes, which differs from each other in terms of issues. In the WHOLE SPELLING mode, the user can practice all available material, in the ONE SPELLING ROULE mode, only a selected part of it, focusing on a specific group of words (e.g., in Polish the following sections are available: U/Ó, RZ/Ż, and CH/H). Selecting one of the modes will set the appropriate parameter and pass it on when calling the NewGame class.

The operation of the PRACTICE mode is shown in the block diagram in Figure 3. After starting the selected exercise, the parameter on the basis of which the data file is selected is downloaded. Then, the number of lines (words) in the file is calculated. Then, in the loop, a line from the file is randomly drawn, a word is extracted, and a possible answer can be inserted, the distribution of answers on the buttons is randomly drawn, and then, the text is displayed in appropriate places. Then, the condition responsible for unlocking the equipment is checked. If it is fulfilled (the number of incorrect answers is greater than or equal to 5), a class for checking the equipment is called. Then, the application waits for

the button to be pressed. When it is a RETURN button, the exercises are completed and returned to the previous screen. If it is a MUSIC or EFFECTS button, the settings are changed, and the user waits for the button to be pressed again. Otherwise, the answer button is pressed and the correct answer is checked. If the answer is correct, the word color is set to green, and the incorrect answer counter is reset. If the answer is wrong, the text color is set to red, and the inaccurate answer counter is increased by 1. Then, the text on the screen is completed with the selected answer and after three seconds, the loop iterates again.

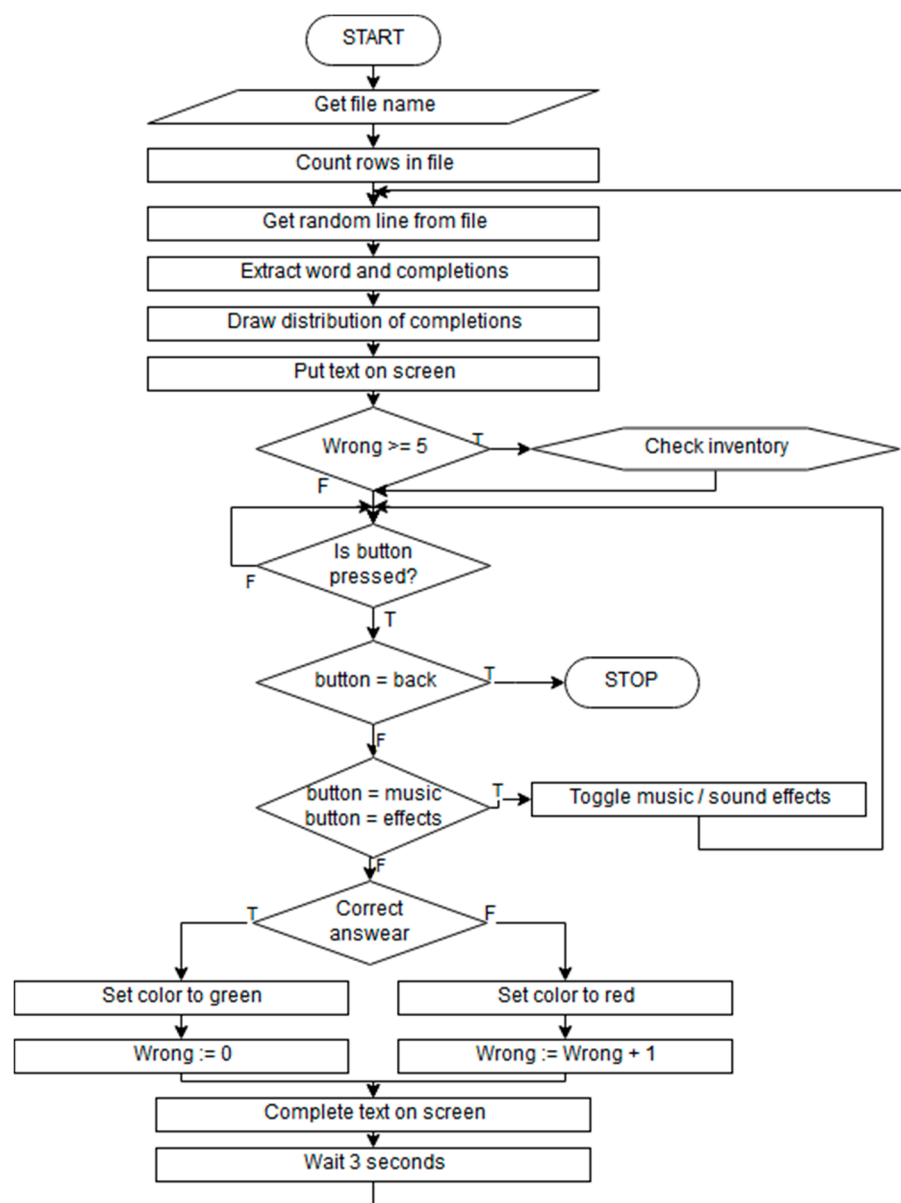


Figure 3. The operation scheme of the PRACTICE mode.

The PRACTICE mode is provided by the NewGame class. In the beginning, the transfers from the previous class (MenuPractice) are retrieved by parameter, from which a file with the word database is loaded. Then, using the LinesNumber method, the number of words in the file is taken, and the game method is called. It is responsible for the whole game. It draws one word from the file using the draw method, which is then prepared and displayed on the screen. The task of the user is to select one of the two available buttons that complement the word displayed above. After clicking the answer button, the word is completed appropriately, and if the answer is correct, the displayed word turns

green. If the answer is incorrect, the word is displayed in red (see Figure 4). Also, if sound effects are enabled in the application, an appropriate message is played stating whether the answer was correct or not. If 3 s have elapsed since the answer was given, the next question is displayed by calling the Game method.

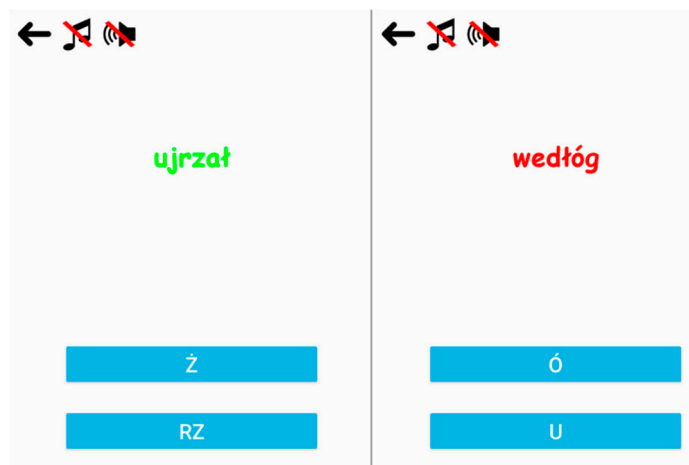


Figure 4. Example of correct (left) and wrong answer (right).

In the upper left corner of the exercise game, there are three buttons: (a) to exit the PRACTICE mode to the previous menu; (b) switch the music setting; and (c) switch the sound effects setting (see Figure 4). A crossed-out icon indicates that the option is disabled.

2.5. GAMES Screen

The GAME screen allows choosing between three game modes: TIME GAME Mode, THREE CHANCES Mode, and INFINITY Mode. These modes enable us to compete with other players (users). Each of the game modes is characterized by different rules of the game and the method of points calculation. It is executed by MenuGames class, in which after clicking on the selected button, the appropriate method is called and passed to the next screen. The following subsections describe the basic properties of these modes.

2.5.1. TIME GAME Mode—Operating Framework

In TIME GAME mode at the beginning, the number of lines (words) is calculated, the equipment currently set by the player is taken, and timers t_1 and t_2 (counting down to 0) are set. If a helmet level 3 (diamond) is worn, the timer t_2 is set to 75 s, if not then to 60 s. Next, waiting for the START button to be pressed takes place. After pressing it, the countdown of t_1 starts from 5 s down and the remaining time on the screen is printed out until t_1 reaches 0, and the countdown of t_2 starts. At this point, until t_2 reaches 0, two things happen in parallel: (a) printing out the remaining time on the screen (together with appropriate coloring) and (b) the proper gameplay. In the loop, a line is drawn, from which a word is separated, and two possible values are displayed on the screen. The application then waits for the answer to be selected, while monitoring the value of the t_2 counter. If one of the buttons is clicked, the correctness of the answer is checked and the number of correct or incorrect answers is incremented. Then, if the value of $t_2 > 0$, the loop is iterated again. If, during waiting for the answer, t_2 reaches the value of 0, the loop is interrupted, and further actions are performed to calculate the scoring. First, the condition that the hero has a second (iron) helmet on is checked, if so, the incorrect answer counter is reduced by three. The next step is to calculate the points and check if the hero is wearing a helmet 1 (straw hat). If this is true, the score is increased by 5%. Then the points are entered into the database and printed on the screen, after which the application waits for pressing the return button that moves the player to the previous screen.

In this mode, the player has 60 s to give as many correct answers as possible to the questions that appear on the screen. TIME GAME mode is operated by the GameTimer class. After starting the onCreate method, it calls the LinesNumber method counting the words in the file, and then the START button is displayed, pressing it starts a 5-s countdown (CountDownTimer class object is responsible for this). Thanks to this, it is possible to start the game fairly because the player has few moments to prepare. CountDownTimer class has two methods, onTick and onFinish, responsible for actions during and after the countdown. In the case of this counter, the first class is responsible for displaying the remaining time on the screen, and the second one is responsible for starting the timer and the game itself. When the countdown is over, it automatically starts the game (the method game of this class). It works similarly as in the PRACTICE mode, the word is displayed, which must be completed by clicking one of the two buttons. However, in this mode, the player does not know if the question has been answered correctly. In addition, at the top of the screen, there is a clock (modified during calling the onTick method of the second counter), which counts down the remaining time until the end of the game. When the time runs out (less than 5 s left), the color of the clock changes to red. After the game time expires, the onFinish method is called, in which the number of points scored by the user is calculated according to the number of correct and incorrect answers.

The final score is calculated by the method calcPoints, which algorithm is presented in Figure 5. Points are awarded to the users according to the rules in Table 1 (pp—points for correct answers and pb—points including wrong answers), then the result is calculated by the equation: $(pp - pb)$, and if it is negative, the player receives 0 points. After the calculation, the screen displays information about the result—how many correct and wrong answers have been given, and points scored. On the screen, there is also a button that allows returning to the GAMES screen (game mode selection menu). The points earned by the user are summed up together with the previously collected points, and then the application creates and calls up a new object of the class EnterPoints, which saves the points in the database. At the same time, a new object of the CheckAccounts class is created, and the called check method is called.

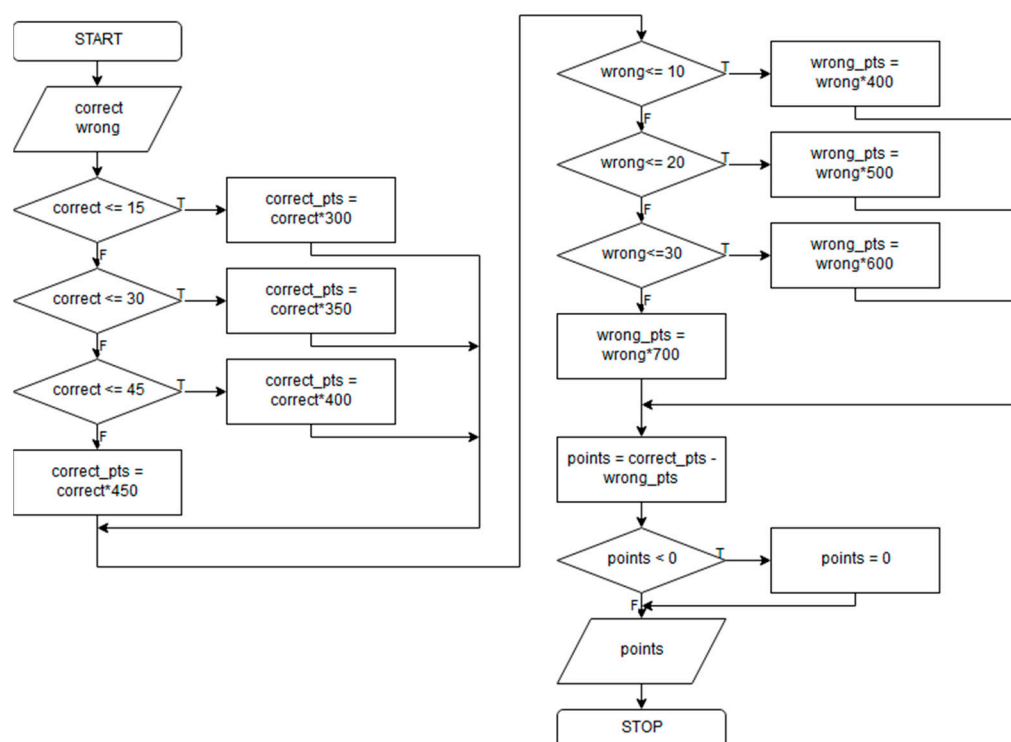


Figure 5. The algorithm of the calcPoints method in TIME GAME mode.

Table 1. Points calculations in the TIME GAME mode.

Correct Answers		Wrong Answers	
Number of Answers	Points	Number of Answers	Points
$x \leq 15$	$pp = x \times 300$	$y \leq 10$	$pb = y \times 400$
$15 < x \leq 30$	$pp = x \times 350$	$10 < y \leq 20$	$pb = y \times 500$
$30 < x \leq 45$	$pp = x \times 400$	$20 < y \leq 30$	$pb = y \times 600$
$x > 45$	$pp = x \times 450$	$y > 30$	$pb = y \times 700$

2.5.2. THREE CHANCES Mode—Operating Framework

The operation of the THREE CHANCES mode is shown in Figure 6. Initially, as in the previous mode, the number of lines in the file is calculated, and the assumed equipment is downloaded. Then, if armor level 3 (diamond) is worn, the number of lives (chances) is set to 4, otherwise, to 3, the control variable k is also set. The loop begins, the remaining lives on the screen are drawn, the system time is taken and if the number of lives is different from 0, similarly to the previously random word, the order of the buttons is printed on the screen. Then, the application waits for the button to be pressed with the selected answer. If this happens, the system time is retrieved again. If the answer is wrong, one life is subtracted, and the execution loop is continued. If the answer is correct, the result is calculated. After that, it is checked whether the control variable is equal to the number of lives (which means that no wrong answer was given) and whether the result ≥ 5000 . If so, the achievement check is performed (whether the equipment item should be unlocked), then the result is displayed, and the next loop iteration begins.

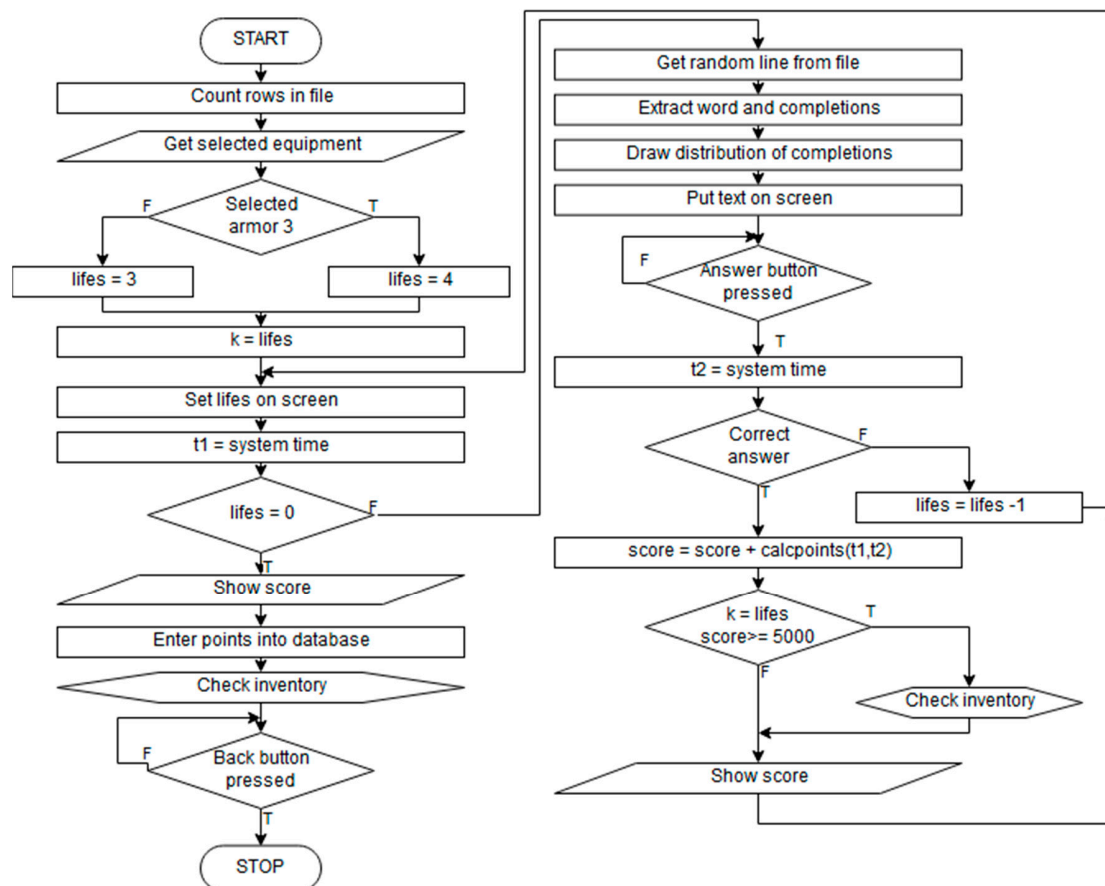


Figure 6. The algorithm of the THREE CHANCES mode.

The THREE CHANCES mode allows users to play until they lose all their lives (symbolized as a heart in the upper left corner of the screen). This is done by the GameHearts class. In the onCreate method, just like before, the LinesNumber method counts the number of words in the file. Then, the setLife method is called, whose task is to display a sufficient number of hearts on the screen (3 or 4 depending on the equipment—see Figure 7). Then the game method responsible for the gameplay is called. In order to maintain consistency, the game is played according to the same rules as in the other modes—it is necessary to complete the word by selecting the correct answer.

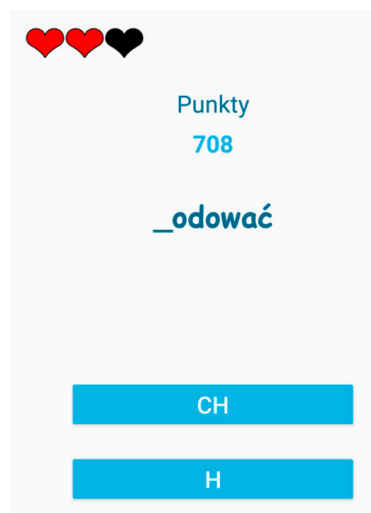


Figure 7. The THREE CHANCES mode. The remaining lives, points, words, and two possible answers are visible.

Points in this mode are calculated according to the time of answering the subsequent questions, using the calcPoints method called after each correct answer. Formulas for calculating points for a correct answer are given in Table 2. This mode, by its construction, could cause some abuse by users, and therefore, if the response time was less than 200 ms (average human response time to visual stimulus is about 250 ms), the player did not get points because ones probably clicked “blindly” on the answers [12].

Table 2. Points calculations in the THREE CHANCES mode.

Response Time t [ms]	Points Awarded
$t < 200$	0
$200 \leq t < 500$	450
$500 \leq t < 2000$	$(2000/t) \times 50 + 200$
$2000 \leq t < 10,000$	$200 - (t - 2000)/75$
$t \geq 10,000$	50

Additional points depend on the equipment chosen by the player and are added to the basic points presented in Table 2. As in TIME GAME mode, after losing all hearts (lives)—which is equivalent to losing—the player receives a message about the number of points scored. As before, the points are added to the points earned by the player and saved in the database by creating and invoking a new EnterPoints class object. It is also checked if the player has unlocked the equipment (one’s achievements)—the check method of the the CheckAccounts class is performed.

2.5.3. INFINITY Mode—Operating Framework

In the INFINITY mode, initially, the number of words in the database file is calculated and the equipment created by the player is downloaded. If weapon level 1 (pitchfork) is created, the player

starts from level 3 (instead of level 0), if weapon level 2 (iron sword) is created, the player gets an extra chance. Then a loop begins in which a question is drawn, processed, and displayed, and the application waits for the button to be pressed. If the answer chosen is correct, the level is increased by 1, and if the hero has a weapon level equal 3 (diamond sword) and the level is higher than 30, the hero will be raised again by 1. Then it is checked if the player has reached level 50—when this happens, the achievement check will be started. This is followed by another loop through. If the player answers the question incorrectly, then if the variable add-on was equal to one, it is reduced, the level gained is also lowered (to a minimum of 0), but the player can continue the gameplay. If the player has already used the chance or has not used the weapon level 2 (an iron sword), the game ends, the result is calculated, which is entered into the database and displayed on the screen. After this, the application waits for clicking on the button to return to the previous screen.

In this mode, the player answers questions in the same way as in the previous modes (completing the missing letters). The mode works with GameInfinity class. Similarly, as before, the onCreate method calls the LinesNumber method, which returns the number of words in the file, and then the game method responsible for the gameplay is called. This mode is more complicated than the previous ones because there is no possibility of making a mistake—one wrong answer is the end of the game. With each correct answer, the higher level is gained, based on the final points calculated. The player can see the current level at the top of the screen. The game method is responsible for all this. It draws the word (draw method), divides it, and displays it on the screen. When the correct answer is selected, the level is increased, and the game method is called up again.

After giving an incorrect answer, the assumed equipment is verified and, depending on it, the gameplay is continued or interrupted. In the second case, the end method is called, calculating the points according to the rules in Table 3. The final result is the sum of points earned for each level. As in other modes, the player receives information about the level and how many points ones earned, and this information is saved in a database (class EnterPoints). It is also verified whether any item of equipment has been unlocked (the check method of the the CheckAccounts class is performed).

Table 3. Points calculations in the INFINITY mode.

Level Reached p	Points Per Level
$p < 100$	$100 + (p/2 - 1) \times 5$
$100 \leq p < 250$	$100 + p/5 \times 3$
$250 \leq p < 750$	$100 + p/10$
$p \geq 750$	$100 + p/20$

2.6. EXAM Screen

The EXAM screen allows users of the application to test their skills acquired during exercises and games. The MenuTest class is responsible for this operation, which, based on the mode selected by the user, sets the appropriate parameter and launches the Test class. There are three tests to choose from, differing in the number of questions:

Basic—consists of 35 questions,

Extended—consists of 70 questions,

Full—consists of 100 questions.

The operating diagram of the EXAM module is shown in Figure 8. After starting, the test mode (basic, extended, full) is obtained, based on which the number of questions is determined. The number of lines (words) in the data file is calculated and the value of correct answers is set to 0. Next, the number variable stores the number of questions in the test, then the loop begins, where it is checked whether the questions have ended. If not, the word is drawn randomly, and two variants are prepared—correct and incorrect—and they are assigned randomly to buttons. In the next step, the application waits for the button to be pressed. If the right answer button is selected, the correct answer counter is increased

by one. Then, the number of remaining questions is reduced by one, the number of the actual question and the number of all questions are displayed, and then, the loop iterates again. When the questions are over, the percentage score obtained by the user is calculated, displayed on the screen, and entered into the database. The end is reached by clicking on an END button.

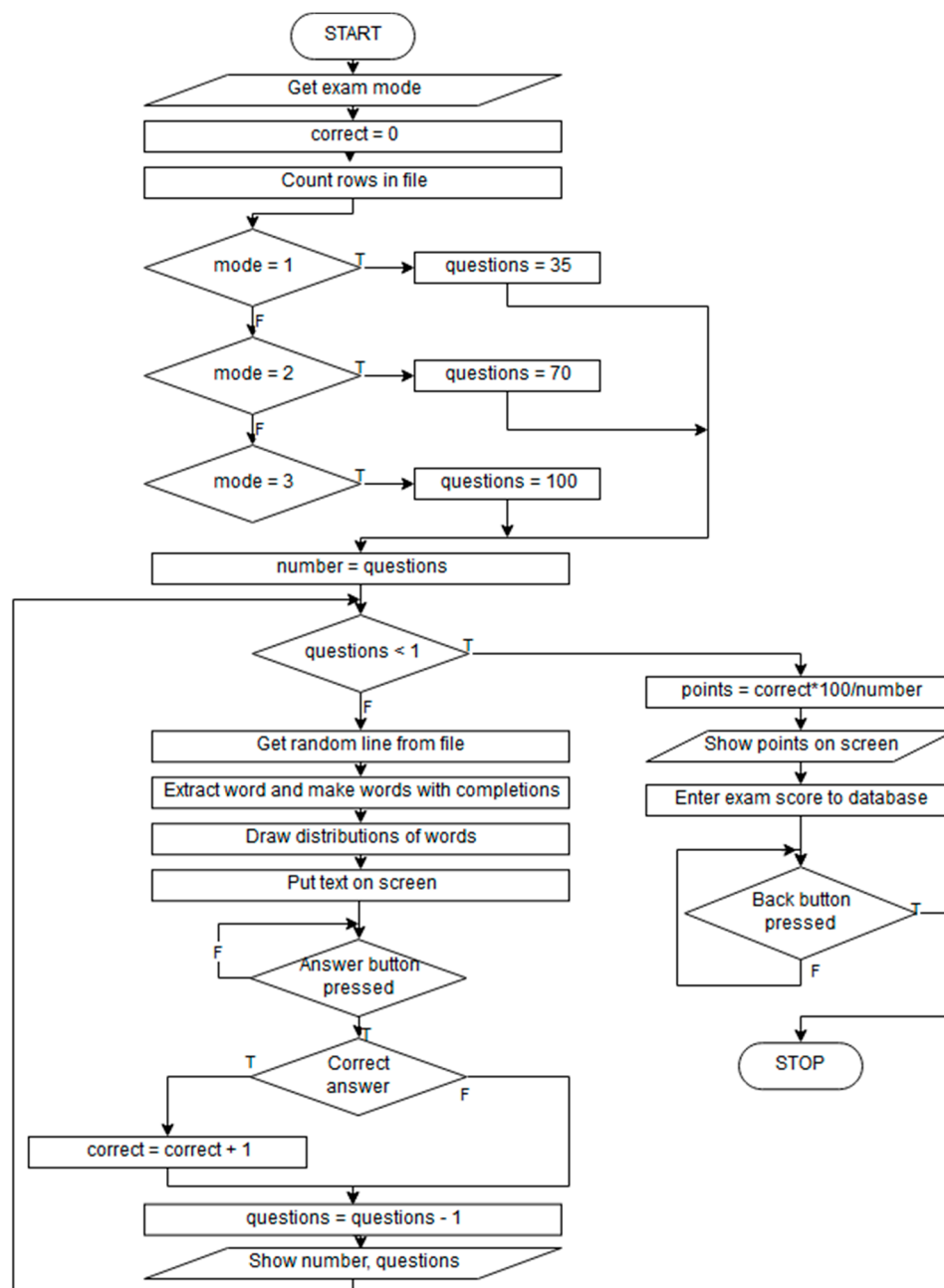


Figure 8. Diagram of the EXAM module operation.

Each gameplay in the EXAM mode is executed by the Test class. After starting, it takes the passed parameter on the basis of which the number of questions is determined. Then, the LinesNumber method is used to count the number of words. The gameplay in this mode has been simplified in comparison to other modes. Instead of a word and two options to complete it, two words are displayed, one of which is correct and the other is not. The selection is made by clicking one of the buttons. In the upper part of the screen, there is a counter indicating the number of the current question and the number of all questions in the given test. The whole is realized by a method game of this class, which

is called until the player answers all the questions. After the end of the test, the result is calculated inside the game method and is saved in the database through an object in the EnterTest class. Next, a message is displayed on the screen, informing the user about the points earned.

2.7. SCOREBOARD and INVENTORY Screens

The SCOREBOARD screen is one of the options available on the MAIN MENU screen of the application. It is supported by the HighScore class. In the main method onCreate, an internal Score class object is created and called, whose task is to retrieve data from the database. Then, it creates a CustomAdapter class object that displays a list of players on the screen. The scoreboard contains a ranking of players in descending order of their points. The data is updated each time the user enters this section. The scoreboard is divided into two sections: on the left, there are the names of the players, and on the right, there are the points earned by them. The result of the currently logged in user is highlighted in green.

Another option available directly from the MAIN MENU screen of the application is the INVENTORY option. The functionality available in this module allows the user to build their avatar and, by achieving better and better results, unlock available items of equipment. These capabilities have the most significant impact on the attractiveness of the gameplay and enhance the spirit of competition resulting in an improvement in the results achieved by users of the application. The INVENTORY screen is operated by the Hero class. After starting the onCreate method, visOff class is called to hide all elements on the screen. Then, depending on whether the player has chosen his hero or not, the character selection method called setCharacter or the showEquipment method is invoked. Its task is to display the items of equipment (unlocked, locked, selected) and to call the setEquipment method responsible for displaying the selected character and items at the top of the screen (“clothing” of the character).

In the INVENTORY module, one can view its avatar and select equipment that has bonuses in selected game modes. There are three sections of the screen—in the upper left corner, there is an arrow to return to the main menu; in the central part, there is an avatar that the user has chosen; and in the lower part, there is a selection of the available equipment. Figure 9 shows a sample screen showing the user’s constructed avatar and each item of equipment.

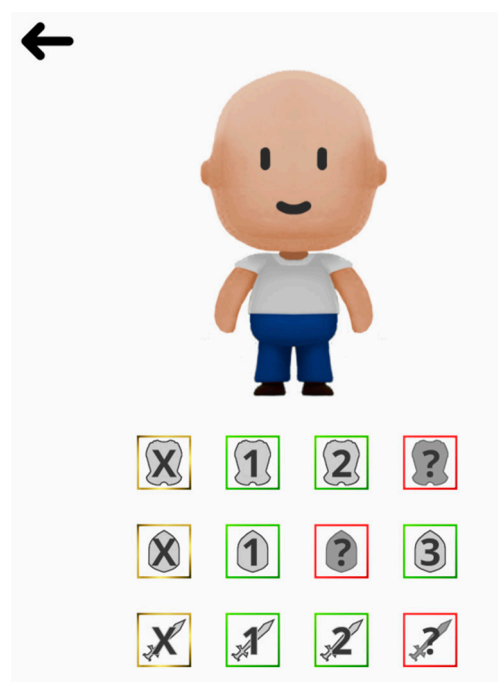


Figure 9. INVENTORY screen and avatar equipment.

The user can choose between three helmet types, armor, and weapons. In addition, the user can deactivate the selected item of equipment by selecting the icon containing X. The selection of the item is realized by the `changeEquipment` method, which checks if the item is unlocked. If yes, the `showEquipment` method described above is called, and if the item is locked, an appropriate message is displayed. Each item of equipment has a unique effect that is available in the game modes and modifies the gameplay. Bonuses have been selected so that the easiest to unlock items have the smallest bonus. In addition, each type of equipment (armor, helmet, weapon) has effects that work in one of the three game modes (Three Chances, Time Play, Infinity) in such a way that once set, the equipment with interesting bonuses does not need to be changed while playing in another mode. Each achievement increases the history progress indicator and unlocks the next stages of the game's history.

2.8. SETTINGS and ABOUT Screens

The SETTINGS screen of the applications (supported by the Settings class) allows users for basic configuration of the parameters of the application such as music, sound effects, and vibrations. Each setting can be switched independently of the other settings, and the change has an immediate impact (no confirmation required). The settings are cached and loaded each time ones start the application—so ones do not have to configure the application according to preferences every time the application begins.

The ABOUT screen contains a summary of the essential information about the application and rules of its operation. It also contains a summary description of the application and its functions, as well as a clause describing the purpose of the application, the author, and other related information.

3. Methodology

Ethical approval: All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. The right to patient privacy and confidentiality of health information is respected.

Informed consent: Informed consent was obtained from all individual participants included in the study.

The scenario of the research assumed that a survey was conducted among all the respondents from primary and secondary school students in order to collect relevant data about the participants, including those with spelling difficulties or dyslexia. At the beginning of the survey, participants registered on a web platform. Before granting appropriate permissions in the system and access to applications, they had to read the study regulations, and then the participants had to sign the informed consent. After this step, the participants were given an identifier and access to the system and applications. The first step after receiving access to the system, which the users had to go through, was to fill in a preliminary questionnaire, the analysis of which is in the next subchapters. In the next step, through a web platform created for this purpose, a preliminary verification study was conducted, that is, a verification dictation. Then, the participants were divided into three groups—one not using the application, one using the application with the vocabulary appearing in the verification dictation, the third using the application, but without the vocabulary appearing in the verification dictation. After a specified period of time (about one week), all three groups (each with four participants) were subjected to the second verification test in the form of a dictation number two. Based on the results obtained, it was possible to determine the impact of the application on the learning process and the results achieved in comparison to groups without the application as well as those learning with traditional methods. Approximately 40 school children applied to the research program; however, 1/3 of the whole applied for the entire research procedure (12 school children finished the whole survey procedure). Persons who did not complete the survey were mainly motivated by lack of time, but expressed positive opinions about the application itself. The knight's story was positively received by users. In the opinion of some users, the history of Bolesław (based on Polish folklore) motivated

them to get all achievements and finish the application. The feedback option was not included in the application, but an email contact was provided to which users could submit comments. Figure 10 shows the interconnections between individual components of the system in the context of subsequent stages of the research methodology.

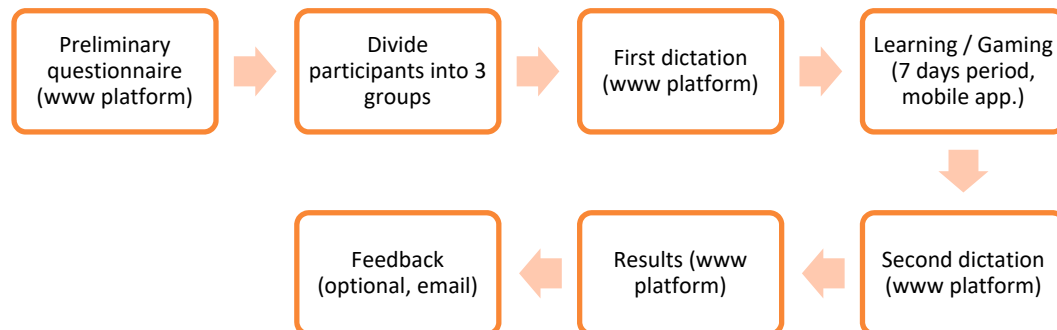


Figure 10. Adopted methodology.

3.1. Preliminary Questionnaire

Each research participant filled in a short questionnaire in the first step. This stage was intended to provide the preliminary data for each user. Thanks to this, the created base was a reference point to check whether the game in this form improved spelling and orthography competence. The results of the interview are as follows. Figure 11 shows the percentage age distribution of the respondents. All participants were between 13 and 17 years old. The largest group was a group of 16-year-olds, which constituted 30% of the whole group.

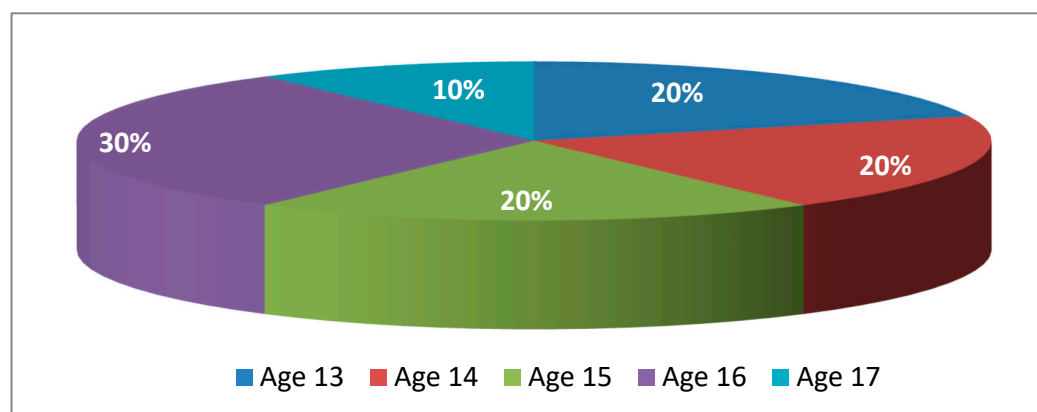


Figure 11. Age distribution of the examined respondents.

The next question in the questionnaire concerned the time at which dyslexia or dysorthography was diagnosed among the respondents. The results are presented in Figure 12. A total of 60% of the respondents had dyslexia/dysorthography diagnosed at least 3 years ago. Only 10% of the respondents had dyslexia/dysorthography at most one year before the survey.

The third question concerned the subjective feeling of the respondents about the frequency of their spelling mistakes. Figure 13 shows the percentage distribution of answers. More than half of the respondents believed that they did not make any mistakes at all, very rarely, or rarely. Only one respondent in the whole research group answered that they often make spelling mistakes.

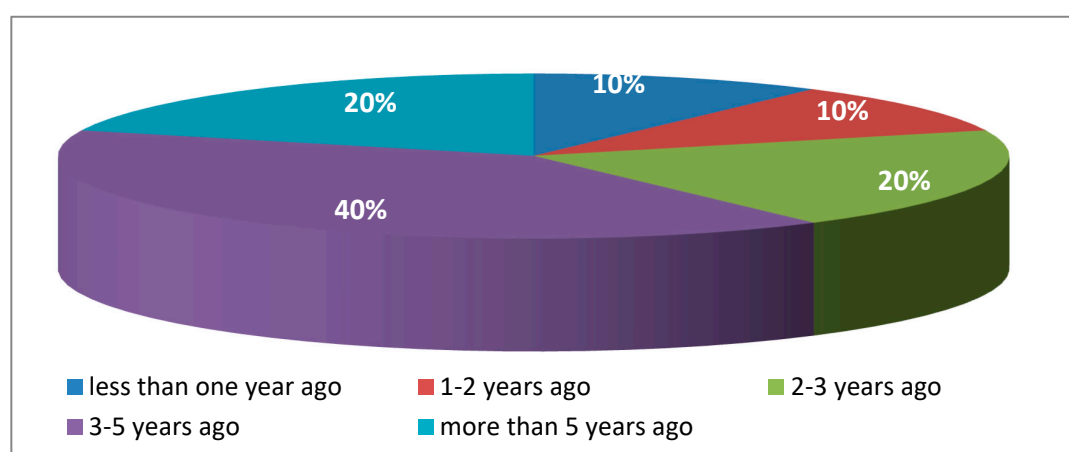


Figure 12. Time at which dyslexia or dysorthography was diagnosed among the respondents.

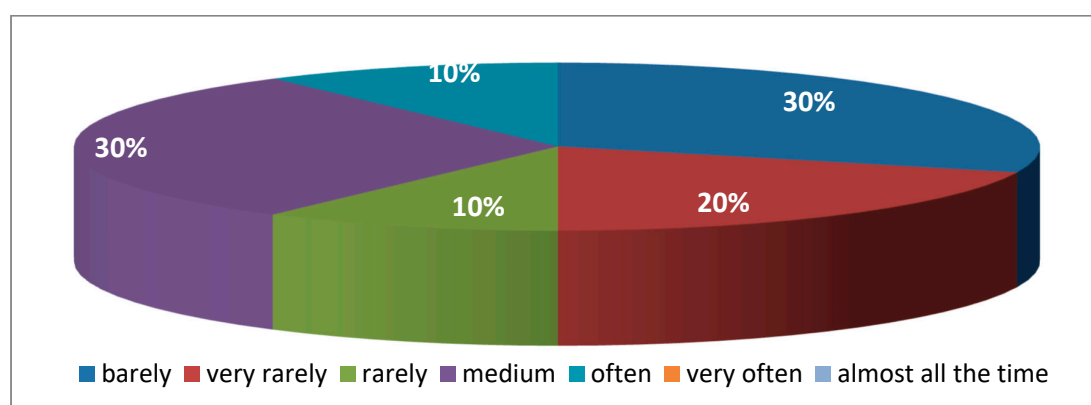


Figure 13. Subjective assessment of spelling mistakes frequency by respondents.

3.2. The First Test Dictation Results

After completing the questionnaire, each participant received an individual link to the dictation on the provided website. The respondents approached the dictation in several ways—some of them tried to write the dictation text correctly from the very beginning, keeping the spelling and punctuation correctness, while others initially wrote a text devoid of Polish characters or containing so-called “typos”, and then, after the recording of the dictation was finished, corrected errors, completed commas and dots. Table 4 shows the types of errors made by the examined users, along with the description.

Table 4. Errors made by the examined users during the first dictation.

Error Type	Description
Spelling error	Respondents made spelling mistakes occasionally from different spelling categories;
A hyphen notation error (hyphenation)	The respondents quite often made mistakes in the spelling of multipart words, omitting the sign of a hyphen [6];
Misinterpretation of proper names	Respondents often considered popular expressions as their names;
Lowercase of proper names	The respondents very often made mistakes by writing the proper names in lowercase letters;
Omitted text fragments	Some of the respondents omitted similar fragments of the text—probably due to too fast dictation of these fragments;
Spelling mistakes “typos”	Seldom typing errors;

The most common mistakes made by the respondents involved writing the proper names in lowercase letters (mainly the names of countries). The names of cities, i.e., Warsaw and Cracow, were

written correctly in most cases (or written initially in lower case and then changed to upper case). Moreover, writing the names of the popular expressions as their names, in capital letters was quite frequent. According to the dictionary of the Polish language PWN, these expressions should be written in lower case [10,11].

There were few spelling mistakes, they were made mainly by 5 people, the rest of the respondents spelled correctly (this applies to words: rz/ż, u/ó, ch/h, etc.), but several respondents did not insert Polish diacritical marks everywhere, replacing them with similar letters (ż/z, ó/o, ł/l, ś/s, etc.). Different types of typos also appeared in the dictation—whether it was moving two letters next to each other, omitting a letter, double adding a letter, or inserting a letter on the keyboard next to the intended letter (e.g., inserting t instead of y or o instead of p).

4. Results

As mentioned earlier, in order to verify whether an application developed within the research is working as intended, the respondents were divided into three groups. The first group was a control group—it did not gain access to the application. The aim of the study was to obtain comparative data. The other two groups had access to a research application containing learning vocabulary, but one of the applications was modified in such a way that learning vocabulary was not included in the second checking dictation (the so-called placebo). In the following subsections, the results obtained and conclusions reached after the second verification dictation are presented.

4.1. The Second Test Dictation Results

After a few days of using the application by the examined users, the second checking dictation for all three groups was performed. Table 5 presents the most common mistakes made by the respondents in dictation number 2.

Table 5. Errors made by the examined users during the second dictation.

Error Type	Description
Spelling error	Respondents quite often made spelling mistakes from different spelling categories;
A hyphen notation error (hyphenation)	A small number of errors;
Misinterpretation of proper names	The respondents made a few mistakes incorrectly using a capital letter in the middle of a sentence. Practically all the respondents misspelled the word “Sahara” with a lower case letter;
Omitted text fragments	Some respondents omitted the same part of the text - some, however, omitted most of the text;
Word twisting, letter addition/loss, letter confusion, inflection errors	Respondents quite often made errors due probably to dyslexia;
Spelling mistakes “typos”	Seldom typing errors;

As can be seen, the second dictation was more complicated, the respondents made more spelling mistakes, in addition, these errors appeared several times, not like in the previous dictation, i.e., a maximum of two times. Moreover, this time very often there were errors associated with dyslexia—omitting or adding letters in words, twisting words or typing words that sounded similar to those presented in a dictation.

4.2. Analysis of Data Collected by the Application

The application collected various data (such as game time, points earned by users or unlocked equipment) during the operation. However, the application did not collect time that the user spent on exercises or tests in the application. This data is summarized in Table 6, and the average of each value is given in brackets. These data show that users of the corresponding application spent an average of at least 4.5 min using the application. However, the time was counted only during the gameplay, not during exercises. Among users with modified applications (the so-called placebo), the average time of

using the application was 8.4 min. Users of both versions of the application conducted a total of 7 tests, including 3 basic tests (35 questions), 1 extended test (70 questions), and 3 full tests (100 questions). The average results achieved by the respondents were 88.6%, 91%, and 63%, respectively.

Table 6. Statistical data collected by the application.

Application Type	Game Time [s]	Points Scored	Unlocked Equipment	Number of Tests Completed
The proper one	826 (275.3)	82,700 (27,566.7)	7 items	2
Modified (placebo)	1516 (505.3)	126,509 (42,169.7)	12 items	5

The individual results of each of the tested user were collected according to the groups in which they were examined, in Table 7. Each of the parameters was summarized within the groups, and the average number per user was given in brackets.

Table 7. Data summarized within the groups and the average number per user (in brackets).

Type of Test Group	Omitted Words		Other Errors	
	Dictation No 1	Dictation No 2	Dictation No 1	Dictation No 2
Control group	82 (16.4)	88 (17.6)	166 (33.2)	175 (35)
Group with a proper application	12 (4)	14 (4.67)	72 (24)	53 (17.67)
Group with a modified application (placebo)	16 (8)	3 (1.5)	68 (34)	61 (30.5)

Several conclusions can be drawn from the data contained in the table. The first is that the control group, during both the first and second dictations, omitted a large part of the text. The first dictation consisted of 68 words, the second of 61, and it results that the control group omitted 24.12% of the text of the first dictation and as much as 28.85% of the second. This is much higher than 5.88% and 7.66% for the proper application group or 11.76% and 2.46% for the modified application group, respectively. Figure 14 presents the distribution of errors committed by specific groups of respondents during both testing dictations.

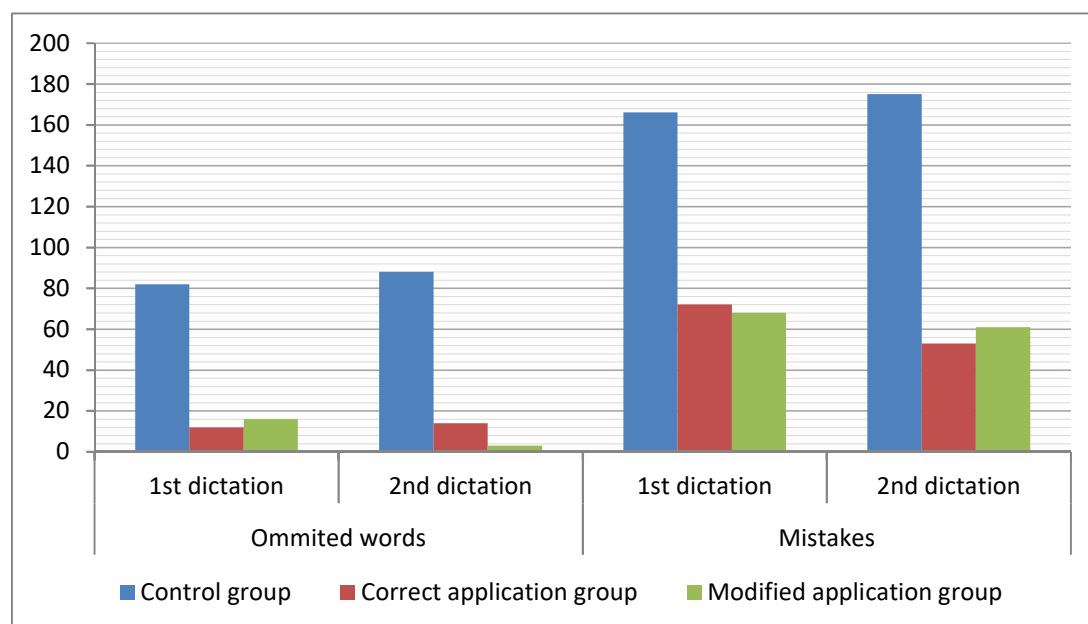


Figure 14. Distribution of errors committed by groups of respondents.

Therefore, the statistics of spelling mistakes made by control group subjects may differ significantly from those of the other two groups. In order to prevent this, the values of omitted words and spelling mistakes were summed up for further analysis. These data are collected in Table 8. As before, the average number of errors per one respondent is given in brackets.

Table 8. Summary of errors.

Type of Test Group	Errors	
	Dictation No 1	Dictation No 2
Control group	248 (49.6)	263 (52.6)
Group with a proper application	84 (28)	67 (22.33)
Group with a modified application (placebo)	84 (42)	64 (32)

This procedure (combining errors into a whole) makes it possible to analyze the results more accurately and effectively. The first group, the control group, was very similar in both dictations, making an average of about 50 errors per person in the first dictation and 53 errors in the second. The group that had access to the proper research version of the application in the second dictation made 20% fewer errors than in the first, more straightforward dictation. Interestingly, the group having access to the research application, but in a modified version, wrote the second dictation better, improving the result by about 23.8%. However, it is not clear whether it was a matter of chance (i.e., the participants of the research knew better the spelling of the vocabulary contained in the second dictation) or placebo (people believed that thanks to the application, they would write the dictation better and thus try harder). The raised doubts may be solved by analyzing the data from Table 7 again and focusing only on incorrectly written words. Then, we can see that the average number of errors in the group using the proper research application dropped from 24 to 17.67 (a decrease of 26.38%), while in the group using a modified application, this ratio dropped only by 10.29% (from 34 to 30.5 errors per person).

Figure 15 illustrates a summary of the learning outcomes for dyslexics/dysgraphic participants after the completion of the second assessment. It can be seen that the modified application group improved their results by about 5.15%. The group with the correct application had the highest percentage of orthography improvement, which is as much as 9.31% compared to the first dictation. The results concerning the omission of words are distributed differently. In the modified application group, the group changed its word skip rate by −9.56%. This means that this group significantly improved the response level and was able to face more words rather than skipping them without giving the answer during testing. The remaining groups slightly increased the number of omitted words by (1.76% for the control group and 0.98% for the group with the proper application), which may be associated with the intention to provide the best possible answers, in particular, concerning those words on which they are fully confident. In terms of answers and orthographic correctness of words, each group visibly improved its achievements by at least 5%, which is a good result and a basis for increased research on gamification.

Analyzing the data presented, there are a few conclusions to make. The first of them is that the respondents, despite their quite high age (13–17 years old), made a lot of spelling mistakes, punctuation, or inflection errors in dictations, but also omitted some words. In the first dictation, the respondents very often wrote the names of countries and other names themselves in lower case letters, but in the case of the names of Polish cities, if they wrote in lower case, they mostly corrected it to upper case. During dictations, participants in the study wrote the text by ear in three patterns—the first one was writing without paying attention to correctness or punctuation; the second one was correcting errors they had noticed; the third one consisted of writing the text as correctly as possible from the very beginning (taking into account the rules of punctuation and spelling). The second conclusion that can be made is related to the results of dictations and access to applications among the respondents. The respondents who had to deal with the application, regardless of whether they

used the proper (with words contained in the second checking dictation) or modified (without words from the second dictation) version, performed much better in the second dictation than those who did not use the application. The difference was 20%–23%, which may indicate the need to modify or supplement the classical methods of dyslexia and dysorthography therapy with new technologies and mechanisms, which may include gamification. Due to a small group of people participating in the study, the effectiveness of gamification of a medical disorder selected at work cannot be 100% confirmed. Therefore, it is necessary to conduct further research on the use of the application in a larger group of people. A preliminary written dictation on paper would have to be carried out to see if gamification can really help and motivate people with dyslexia/dysgraphia. The partial results obtained by the individual respondents differed from each other, resulting from the individual predispositions of the subjects participating in the study. Some of the respondents obtained much worse results in the second dictation than in the first one (making even more than twice as many errors as in the first dictation), while others obtained much better results, of a similar scale as in the first case. It should be noticed that the second dictation had a higher level of difficulty than the first one—it contained words that were not often used in everyday speech. However, analyzing the results of the number of mistakes made, the respondents in the second dictation made fewer mistakes than in the first one, mainly due to a smaller number of errors among the groups using the application implementing gameplay. Based on the collected data, we can conclude that the proposed framework and gamification can help in the learning of people with dyslexia. Visible improvement in the results positively affects the learning process of spelling and orthography.

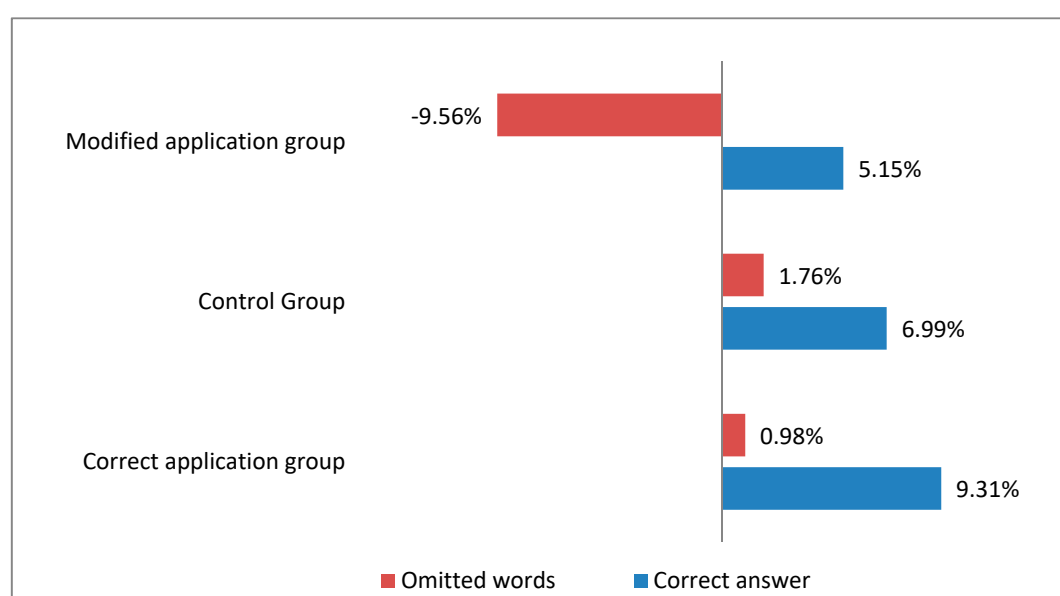


Figure 15. Summary results of learning effectiveness.

4.3. Feedback from the Users

Muhammed [25] said that “Stories are essential elements of child development as it helps in both native and second language acquisitions.” The last element of the whole process was the information directed to the participants of the research about sharing their opinions about the system and the research. The participants were eager to express their opinions (an email option), which were mostly positive. We learned from the obtained feedback that the knight’s story was positively received by users, which motivated them. According to users, the story of Bolesław was interesting, based on the legend of the Wawel Dragon. Thanks to such an exciting story, their involvement in the project and the improvement of skills increased. In their requests, users asked to add a more extended and more productive history, some knight animation, and increase the database of questions.

5. Conclusions

In the paper, we proposed an application supporting the teaching of school children with diagnosed developmental dyslexia—dysorthography. The application using gamification mechanisms was developed and designed to increase the attractiveness of the tools of the teacher and the use of immersive learning. Preliminary studies were also carried out to assess the impact of the development of the application on student performance.

Analyzing the data collected by the application, it can be noticed that the respondents were not eager to run the tests contained in the application. It could have been caused by the fact that the application was associated with a game or entertainment, and the module for conducting tests (which can be called exams) did not offer users the opportunity to compete with others and could be perceived by them as an unpleasant duty (as well as tests or tests at school). In the future, this module may be modified so that it becomes more pleasant to receive by the users of the application so that they can use it more often. The second option could be to build a separate application based on this module only for conducting tests or exams and to remove it from the application implementing the gamification of selected disorders. While using the application, the respondents probably focused their attention on the exercises preparing for the second dictation. This resulted in a relatively short time spent in the implemented game modes. In the future, it is possible to modify the application in such a way that the time spent in the application is counted, as well as the time spent on exercises, games, and tests. Thanks to this, it would be easier to analyze the results of the research, and it would be possible to link the results of the research with the time spent on specific activities. It would also be possible to implement mechanisms for “tracking” users’ actions, to be able to adjust later the operation of the application to the actions of users and mechanisms for notifying the user about a downgrade in the ranking or too long a period of not using a research application.

Summarizing, in the presented application we can distinguish the following advantages: Compatibility—works on most Android devices and adapts to any screen; Simplicity—activities are grouped in a straightforward menu, elements are legible; Attractiveness—the application is visually attractive and also provides entertainment (e.g., when playing in built-in modes); Possibility to use in any place—after the first start (selection of avatar and downloading additional files), one can use the application even without the Internet (the result is saved locally); Gameplay—an exciting competition scenario and a scoring system that allows for the expansion of an avatar in the game. It naturally enhances the desire to compete and achieve better results by passing the next stages of the game and tests, of which a side effect is the learning of selected skills.

In the current version, it is also possible to identify some shortcomings that will be eliminated in future versions, e.g., The necessity of connecting to the Internet—after starting up, data from the database are downloaded, and during the operation, the data are updated—without an Internet connection, scores are saved only locally; Monitoring options—only the game time was monitored, no other activities could be monitored; Polish language—the application has been prepared only in Polish, for Polish users; Range of material—the application focuses on only selected spelling principles.

In the future, the application will be slightly modified, more accurate tracking and monitoring of the researched activities will be added—not only the time spent in each activity but also the number of visits in order to identify the most attractive activities. A feedback option will also be added as a separate screen in an application (not only email option) so that respondents can report problems or tips on the fly. The next research should be conducted on a much wider group of people but narrowed down to two groups (instead of three)—using and not using the application. If possible, people would be subdivided into subgroups according to age or defect level. The time for which people would use the application should also be extended. The analysis should also be carried out in terms of time spent on specific activities in the application and the final results. It is also possible to use another form of spelling validation or to extend the application with other spelling rules.

Author Contributions: Conceptualization, P.D.; methodology, P.D. and K.N.; software—formal analysis and investigation, P.D. and K.N.; resources, P.D. and K.N.; writing—original draft preparation, P.D. and K.N.; writing—review and editing, P.D. and K.N.; visualization, P.D. and K.N.; supervision, P.D.; project administration, P.D. and K.N.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Data Availability: The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. Gamification—Definition of Gamification in English by Lexico Dictionaries. Available online: <https://www.lexico.com/en/definition/gamification> (accessed on 22 July 2019).
2. Deterding, S.; Dixon, D.; Khaled, R.; Nacke, L. From game design elements to gamefulness: Defining “gamification”. In Proceedings of the 15th International Academic Mindtrek Conference: Envisioning Future Media Environments, Tampere, Finland, 28–30 September 2011; Lugmayr, A., Ed.; ACM: New York, NY, USA, 2011; pp. 9–15.
3. Sailer, M.; Hense, J.U.; Mayr, S.K.; Mandl, H. How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. *Comput. Hum. Behav.* **2017**, *69*, 371–380. [CrossRef]
4. Armstrong, M.B.; Landers, R.N. An Evaluation of Gamified Training: Using Narrative to Improve Reactions and Learning. *Simul. Gaming* **2017**, *48*, 513–538. [CrossRef]
5. Hamari, J.; Koivisto, J.; Sarsa, H. Does Gamification Work?—A Literature Review of Empirical Studies on Gamification. In Proceedings of the 47th Hawaii International Conference on System Sciences, Waikoloa, HI, USA, 6–9 January 2014; pp. 3025–3034.
6. Burguillo, J.C. Using game theory and Competition-based Learning to stimulate student motivation and performance. *Comput. Educ.* **2010**, *55*, 566–575. [CrossRef]
7. Malhi, S.K.; McAuley, T.L.; Lansue, B.; Buchanan, L. Concrete and abstract word processing in deep dyslexia. *J. Neurolinguist.* **2019**, *51*, 309–323. [CrossRef]
8. Toffalini, E.; Pezzuti, L.; Cornoldia, C. Einstein and dyslexia: Is giftedness more frequent in children with a specific learning disorder than in typically developing children? *Intelligence* **2017**, *62*, 175–179. [CrossRef]
9. Maehler, C.; Joerns, C.; Schuchardt, K. Training Working Memory of Children with and without Dyslexia. *Children* **2019**, *6*, 47. [CrossRef] [PubMed]
10. Hamari, J.; Shernoff, D.J.; Rowe, E.; Coller, B.; Asbell-Clarke, J.; Edwards, T. Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. *Comput. Hum. Behav.* **2016**, *54*, 170–179. [CrossRef]
11. Hainey, T.; Connolly, T.M.; Boyle, E.A.; Wilson, A.; Razak, A. A systematic literature review of games-based learning empirical evidence in primary education. *Comput. Educ.* **2016**, *102*, 202–223. [CrossRef]
12. Experiment: How Fast Your Brain React To Stimuli. Available online: <https://backyardbrains.com/experiments/reactiontime> (accessed on 22 July 2019).
13. Cancer, A.; Stievano, G.; Pace, G.; Colombo, A.; Antonietti, A. Cognitive Processes Underlying Reading Improvement during a Rhythm-Based Intervention. A Small-Scale Investigation of Italian Children with Dyslexia. *Children* **2019**, *6*, 91. [CrossRef] [PubMed]
14. Gooch, D.; Vasalou, A.; Benton, L.; Khaled, R. Using Gamification to Motivate Students with Dyslexia. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, 7–12 May 2016; ACM: New York, NY, USA, 2016; pp. 969–980.
15. Saputra, M.R.U.; Risqi, M. LexiPal: Design, Implementation and Evaluation of Gamification on Learning Application for Dyslexia. *Int. J. Comput. Appl.* **2015**, *131*, 37–43.
16. Lumsden, J.; Edwards, E.A.; Lawrence, N.S.; Coyle, D.; Munafò, M.R. Gamification of Cognitive Assessment and Cognitive Training: A Systematic Review of Applications and Efficacy. *JMIR Serious Games* **2016**, *4*, e11. [CrossRef] [PubMed]
17. Dymora, P.; Mazurek, M.; Kowal, B. Dydaktyczne aspekty projektowania aplikacji w środowisku Unity 3D. *Eduk. Tech. Inform.* **2019**, 185–193. [CrossRef]

18. Byeong, C.L. The Effect of Gamification on Psychological and Behavioral Outcomes: Implications for Cruise Tourism Destinations. *Sustainability* **2019**, *11*, 3002.
19. Toda, A.M.; Valle, P.H.D.; Isotani, S. The Dark Side of Gamification: An Overview of Negative Effects of Gamification in Education. In Proceedings of the Higher Education for All. From Challenges to Novel Technology-Enhanced Solutions (HEFA 2017), Maceió, Brazil, 20–24 March 2017; Cristea, A., Bittencourt, I., Lima, F., Eds.; Communications in Computer and Information Science. Springer: Cham, Switzerland, 2018; Volume 832.
20. Franceschini, S.; Gori, S.; Ruffino, M.; Viola, S.; Molteni, M.; Facchetti, A. Action Video Games Make Dyslexic Children Read Better. *Curr. Biol.* **2013**, *23*, 462–466. [[CrossRef](#)] [[PubMed](#)]
21. Green, C.S.; Bavelier, D. Action video game modifies visual selective attention. *Nature* **2003**, *423*, 534–537. [[CrossRef](#)] [[PubMed](#)]
22. Green, C.S.; Pouget, A.; Bavelier, D. Improved probabilistic inference as a general learn. *Curr. Biol.* **2010**, *20*, 1573–1579. [[CrossRef](#)] [[PubMed](#)]
23. Cagiltay, N.; Ozcelik, E.; Ozcelik, N.S. The effect of competition on learning in games. *Comput. Educ.* **2015**, *87*, 35–41. [[CrossRef](#)]
24. Iacovides, I.; Cox, A.L.; McAndrew, P.; Aczel, J.; Scanlon, E. Game-play breakdowns and breakthroughs: Exploring the relationship between action, understanding and involvement. *Hum. Comput. Interact.* **2015**, *30*, 202–231. [[CrossRef](#)]
25. Muhammed, A.I. The Impact of Storytelling on Young Ages. *Eur. J. Lang. Lit.* **2016**, *6*, 115–118.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).