Bachelor DSAI

The Influence of Personalised Feedback on Coding Education for Children through the Use of an Educational Game

Lisa Pomp

Supervisors:
Anna van der Meulen & Giulio Barbero

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

20/01/2025

Abstract

This study aims to investigate the impact of personalized feedback using an educational game on the learning process of children in the field of computer science. Programming as a school subject is becoming increasingly widespread. However, there are common challenges that occur when children first learn to code, that have yet to be overcome. One consideration in overcoming these challenges is by using personalized corrective feedback. In this research, six participating children between the ages ten to twelve were given an educational game, in which they had to solve coding puzzles to progress in the game. A system in the game kept track of the mistakes made by the child and alternated through different corrective feedback types, to find the one that worked best for them. Each time a child made a mistake, the researcher used a generative AI model to generate personalized feedback for the child, using the feedback type displayed in the game. The results did not provide clear evidence for a positive impact on the learning process of the children. This is partly due to the small sample of participants and the other limitations in the research. There were, however, small implications found that further research could show improved coding skills caused by personalized feedback, specifically through the use of AI.

Contents

1 Introduction

Programming education is becoming increasingly important in primary education for children. Currently, digital literacy is already a mandatory subject in Dutch primary schools [37]. However, there are some common challenges in this field that have yet to be overcome. Du Boulay [13] differentiated between the following five main areas of difficulty in: orientation, the notional machine, notation, structures, and pragmatics. Additionally, according to Cognitive Load Theory, the complexity of programming can lead to a cognitive load that is higher than the capacity of the working memory. This can impact the process of learning to code negatively.

One thing that can be challenging when learning to code, is comprehending error messages displayed by the compiler. They often use jargon and do not always refer to the correct line of code that is causing problems. This is problematic as feedback is a key part of education in any field. A type of feedback that is especially relevant for this topic is corrective feedback, which has the specific intention of helping the receiver learn [32]. This category of feedback has been divided up in seperate types in [38], building upon the work of [35] and [4]. These types are the following: Explicit Correction, Recast, Clarification Request, Metalinguistic Clues, Elicitation, and Repetition, which are further researched in Section 2.2.2 of this thesis. Which one is most beneficial for the learning process could differ per child. Several studies showed potential benefits of using personalized feedback on learning processes [29] [21].

A potential way to make personalized feedback practical and realistic in terms of decreasing the need for human labor, is through the use of Artificial Intelligence. Several studies have already been done, researching the possibilities of AI in education, even specifically looking at feedback in combination with AI [40] [21]. However, little research has been done that combines the existing applications of AI for personalized feedback in connection with pedagogical theories.

1.1 Research questions and contributions

This research aims to study the effects of personalized feedback on the learning process of programming education and the experience of it on children, considering the six different feedback types. This is done through the use of an educational game, which is used as a platform to explore the possibilities in personalized feedback. Additionally, this thesis will explore how Artificial Intelligence can be used to optimize the efficiency and quality of personalized feedback.

The main research question is:

RQ: In what way can personalized corrective feedback using an educational game influence the coding learning process of children around the age of ten to twelve?

The first subquestion is:

SQ1: In what way can an educational game be used to provide personalized feedback to a child?

Specifically, the use of Artificial Intelligence is explored.

The second subquestion is:

SQ2: In what way does personalized feedback enhance the learning progress and experience of a child in the field of computer science?

The final contribution of this thesis is an educational game that contains a system that allows for personalized feedback, in addition to insights about the influences of personalized feedback on the programming education process for children. Furthermore, this research provides insights in possible ways to apply personalized feedback in educational games, specifically through the use of AI.

1.2 Overview

This thesis provides background information and related research in Section 2. After that, the methodology of the experiment is described in Section 3. Then, the results are presented in Section 4 of this thesis, containing some general results in Section 4.1, results regarding the use of Artificial Intelligence in Section 4.2 and the specific results per session in Section 4.3. At last, all of the results are interpreted and discussed in Section 5.

2 Related Work

2.1 Programming education

2.1.1 Coding for young children

Programming education is becoming an increasingly relevant subject. In the Netherlands, digital literacy is even a mandatory part of primary school education [37].

Abesadze and Nozadze [2] analyzed previous research to list reasons that programming education is important for children. First, they mention the acquisition of several skills. These are problem-solving, independent decision-making, creativity, planning skills, logic skills, insistence, collaboration and communication [23] [6] [30] [33] [23] [6]. Another reason they named, is that it can in some ways, be compared to learning a new language. This in addition, means that skills acquired by learning a new language can also be acquired or improved by learning to code [20] [25]. Furthermore, they state that it is believed by many researchers that learning to code can improve the math skills of children, and make math a more engaging and enjoyable subject [17]. Lastly, they mentioned that it provides students with the possibility to build their own websites [33]. Regardless of the fact that it might be a useful and fun skill to obtain, this does not seem to be as crucial and important as the other benefits mentioned. Although some of the used resources are not rooted in academic literature, one can imagine these effects to be plausible.

Turan and Aydoğdu [36] studied the effect of robotic coding education on the scientific processing skills of pre-school children. They first gave two groups of five-year-old children a test to assess their skill of scientific process. Then, they provided one of the two groups with sixteen sessions of a "Coding and Robotic Education Program". After that, they tested both groups with the same test again. They found that the group that had been given the lessons significantly improved on their test, while the other group came up with similar scores as before [36].

All of these reasons show the potential benefits of teaching young children how to code. This can, however, be challenging in many ways.

2.1.2 Difficulties of learning to code

There are numerous difficulties that occur for children in their initial encounter with programming. Du Boulay [13] differentiated between five areas that provide challenges for novice programmers. These five areas are: orientation, the notional machine, notation, structures, and pragmatics. With the first area, orientation, du Boulay intends to address the difficulty for new students to understand the usefulness of learning to code and what can be achieved by programming. Second, the notional machine refers to comprehending how the computer processes and executes code. The third one, notation, is the area that includes both the syntax and semantics of programming languages. Fourth, there is the structure area of difficulty. This refers to understanding how to use certain concepts to reach a goal, such as loops or data structures. The last one, pragmatics, is about problem-solving skills and the ability to translate a problem into code. As mentioned by du Boulay as well, these areas cannot be completely separated from each other [13]. For example, difficulties with understanding variables could be linked to the notional machine, notation and structures.

Bakker [5] has a slighlty different outlook on, where the challenges lay in programming education. According to Bakker, learning to code is similar to learning a new language. Conversely, the computer will strictly adhere to executing the provided code. This is in contrast to learning a new language, as a person will actively attempt to interpret the intent of the language learner [5]. When the computer executes the code and it leads to a different outcome then the child had anticipated, it can be challenging to identify and resolve the problem. Additionally, a language can be actively used, even if mistakes are still made, while a code with bugs will not work. This shows the importance of feedback in the process of learning to code. This will be discussed further, later on in Section 2.2, as a central component in supporting learning how to code.

2.1.3 Cognitive Load Theory

Underlying the previous mentioned challenges in computer science education, is the Cognitive Load Theory. Human memory can be divided up into working memory and long-term memory. Working memory is also considered to be the conscious memory as that contains the information that a person is actively aware of, while long-term memory is more in the background [34] [27].

As theorized by Miller [24], the working memory can hold seven plus or minus two bits of information at the same time. This number counts however, only for purely holding the information. When it comes to actually using and processing it, the number is believed to be much lower; around two or three bits [34]. Long-term memory can store large amounts of information and it does this by using schemas. These are mental structures or frameworks that categorize and organize information. They can be brought from long-term memory to working memory [27].

Information can vary in element interactivity. This means that it can vary in how well it can be understood individually, thus without understanding other elements. Concepts with low-element interactivity have little to no relation to other elements when it comes to needing them for full comprehension [27].

Computer science has a high-element interactivity, as can be concluded when looking at the areas of difficulties discussed in the previous section on difficulties of learning to code (2.1.2). The areas of difficulties show not only the many aspects of learning to code, but also how they interact with each other. There are many concepts that need to be understood to be able to code well, and many of those concepts connect with each other, adding to the complexity. For example, one cannot understand how to use concepts such as variables and loops, if one does not comprehend how the computer executes the code.

In the Cognitive Load Theory, cognitive load is divided up in three categories: intrinsic load, extraneous load, and germane load. High-element interactivity falls under intrinsic load, as it mainly entails complexity of a subject. The second load is caused by the way information is presented to a learner. For example, if it is presented in an unclear or confusing manner, it increases the extraneous load. The last one is different from the first two, because it should be optimized instead of minimized to enhance learning. Just like extraneous load it depends on the instructional designer. Germane cognitive load leads to automation and development of schemas [27]. According to the Cognitive Load Theory, these three loads add up and cannot be more than the capacities of the working memory allow. In the case that it does, it negatively impacts the learning process [34] [27].

As mentioned before, computer science has a high-element interactivity. This leads to a high intrinsic load, which increases the general cognitive load. Since presenting information in a clear way can reduce that cognitive load, it is relevant how programming is taught to children. Additionally, keeping that in mind, giving them clear and useful feedback could potentially help with the reduction of load.

2.2 Feedback

As previously discussed, feedback can be useful in programming education. It can come in many forms, but overall it holds the general purpose of making the learner understand or correct their mistake. It is, in essence, information regarding the performance of someone. Programmers already receive feedback when an error occurs, namely in the form of compiler messages.

2.2.1 Error messages

When a programmer makes a mistake, it is in most cases shown by an error message produced by the compiler. These messages aim to point the coder to the problem, which would, ideally, allow them to fix the bug. However, they are often difficult to understand, especially for novice programmers. In some cases, the message can even be misleading or it can suggest an incorrect solution for resolving the issue. As can be seen in Figure 1, the arrow points at the double quotation mark, which could lead to a child or novice programmer removing it, rather than adding one on the other end. Besides that, it does explain the error in the lower line, but it uses complicated terms that can be confusing to novice programmers and children.

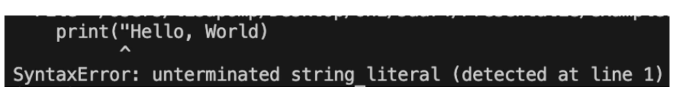

Figure 1: An example of an error message

Many studies show that the complicated compiler error messages have a negative effect on students learning to code. For example, Flowers, et al [16] developed a program called Gauntlet that identified syntax errors in code and explained them in easier to understand language. Additionally, it could find common semantic errors made by novice programmers. In their study, they found that Gauntlet improved the code written by the students and decreased the workload for the instructor, as less assistance was required. This shows the positive impact of easily understandable error messages.

Another similar example is by Schorsch [31]. Schorsch studied the impact of using the Code Analyzer for Pascal (CAP), which is a tool that finds errors in code and provides easy to understand feedback. The results, much like the study on Gauntlet [16], show an increase in quality of the code of the novice programmers, and a decrease in time spent assisting and grading by the teachers.

2.2.2 Corrective feedback

A subcategory of feedback that is particularly effective in terms of helping students in their coding learning process is referred to as corrective feedback. This is defined by Shute [32] as: "information communicated to the learner that is intended to modify his or her thinking or behavior for the purpose of improving learning". To specify further, in the context of coding, feedback that can help students to understand and correct their mistakes.

Van Amerongen [38], building upon the work of [35] and [4], put six distinguishable strategies within the category of corrective feedback into the context of coding education. The list of strategies made by Van Amerongen [38] is as follows:

1. **Explicit correction:** the exact mistake is pointed out and an exact solution is provided.
2. **Recast:** a mistake is implicitly corrected by reformulating
3. **Clarification request:** it is emphasized that a mistake has been made by asking for clarification
4. **Metalinguistic clues:** a comment or question is posed hinting at the required theory to correct the mistake
5. **Elicitation:** an unfinished comment or question is posed that the student has to fill in, which gives a clue for how the mistake can be corrected
6. **Repetition:** a comment is repeated in its entirety with the mistake pointed out

Furthermore, Van Amerongen considers two different types of corrective feedback. First, there is explicit feedback. This type of feedback is specific in terms of indicating the exact error. Then, there is implicit feedback, which does not provide a learner with the exact mistake.

According to Ellis, Loewen & Erlam [14], who analysed previous studies on those two different types of feedback, determining which of the two results offers greater advantages remains challenging. Although numerous studies indicated slight advantages for explicit feedback, others concluded the opposite. This could be due to the fact that individuals differ and things that work best for some, may not for others. For this reason, using feedback in the form of different types for different people, can be very useful. In other words, the use of personalized feedback could improve learning processes.

2.2.3 Personalized feedback

As mentioned in the previous section, the most effective type of feedback for one's learning process may vary per individual. Pérez-Segura, Ruiz, González-Calero & Cózar-Gutiérrez concluded in their study that the use of personalized feedback had a positive impact on the learning process of listening and reading skills. More specifically, they saw a bigger improvement of receptive skills than in the control group [29].

Another study done by Kochmar et al [21] also showed a positive effect on the student learning outcomes, supporting the hypothesis that personalized feedback can improve education. This study is further discussed in the next section on Artificial Intelligence in education.

Although not specifically tested in the computer science field, similar influences on children learning to code can be anticipated. This is especially the case, considering the difficulties that come with learning to code, and the complicated error messages that are currently the main form of feedback in this field, as previously discussed in the sections on difficulties of learning to code (2.1.2) and error messages (2.2.1).

Personalized feedback could be implemented for coding education in many ways. It can be specifically useful in the context of educational games.

2.3 Digitalization in education

Providing students with personalized feedback is very labor-intensive, and can for that reason benefit from digitalization. An existing and increasingly popular manner of teaching concepts is educational games.

2.3.1 Educational games

Educational games are becoming an increasingly popular medium for teaching children how to code. Although there is little certainty what the exact reason for this is, many studies show positive effects of learning through play. De Freitas [11] analyzed the perspectives of education science, game science, neuroscience, and information science on educational games. In the education science discipline, Piaget [28] first proposed the importance of play as early as 1962. Although few studies explore the underlying reasons, many studies have shown positive effects of learning through the use of games, in all of the researched disciplines [11].

Green & Bavelier [18] suggested gradual increase of difficulty, motivation and feedback type as possible factors that, if maximized, can positively affect learning progresses. Both a gradual increase of difficulty and feedback types can be relatively easily implemented and maximized in games. Furthermore, the effect of gamification on motivation has been widely researched. Several studies, including the one by Alsawaier [3] show increase in motivation through the use of games. Although some, including those by Dahlstrøm [10] and Buckley and Doyle [9], show different effects on different people, the general consensus is that gamification enhances motivation.

The motivational characteristic of educational games not only makes it a beneficial medium for learning, but also for experiments, as it makes sure that subjects actively participate. Additionally, games are convenient for studies, as they allow for many factors to be in the control of the researcher (for example: the degree of difficulty, the exact explanation, etc.).

Aside from the learning and research benefits, educational games are an efficient medium for personalizing education, since it can be used labor-free, in contrast to physical teachers. An especially useful tool for personalization is Artificial Intelligence.

2.3.2 Artificial Intelligence in education

Personalized feedback can be implemented in programming education in several ways. One way that is especially promising in the context of educational games is the use of AI. Currently, Artificial Intelligence is already being used in education with various applications. For example, many studies focusing on using it to function as a digital tutor, described by Beck, et al. [7] as digital tutoring systems. These systems aim to be the replacement of human teachers.

A study done by Kochmar, et al [21] looked into the effects of personalized feedback generated by AI. They used a platform called Korbit, on which children watch short video lectures and then do exercises. The platform uses machine learning to provide personalized help and explanations for students. In their case, they alter the hints given to students based on their performance, changing the amount of detail in the hint and the level of difficulty. The results showed that the personalized feedback positively improved the learning outcomes of the children.

2.3.3 Relevance of Artificial Intelligence

As discussed in previous sections, understandable and useful feedback can have a positive effect on the learning process of a student. Thus, even though programmers already receive feedback in the form of error messages, it barely helps them.

Besides the compiler messages, most of the feedback that students receive is from a teacher. This is very labor-intensive for multiple reasons. Firstly, a person has to be trained in a subject themselves, which can take years. Furthermore, they are not available every moment of every day, as they require basic needs such as eating and sleeping. Lastly, even an expert in the field requires time to identify a problem and come up with a solution. A computer does not encounter the same problems. It is able to provide feedback at any given moment, regardless of the time or day, and it can pinpoint the errors in mere seconds.

For these reasons, it is relevant to research the possibilities of AI in education, specifically in the area of feedback.

2.3.4 Possibilities of Artificial Intelligence

AI has been used to produce feedback for students already, as described by Zhai, et al [40]. Even research for the use of AI to produce personalized feedback already has been done [21], as discussed in section 2.3.2. There is, however, an absence of research regarding pedagogical theories in connection with automated feedback, whether personalized or not, using ai.

An AI program could alternate between various feedback strategies based on collected data regarding the learning process of a child. Such a program can use machine learning to track the students progress and mistakes, and based on that it can change the feedback to be more effective for that specific person. For example, if one feedback strategy was used after a previous assignment, but was barely effective, a different one could be tried out.

This is more challenging than it might seem. The reason for this is that the feedback needs to be accurate, useful and easily understandable at the same time. To elaborate, it needs to be accurate in terms of leading to the correct solution. Additionally, it needs to be useful in terms of actually helping a person get to the solution. Lastly, it needs to be understandable in a way that the used language, the structure of the sentence or the grammar does not get in the way of the meaning that the hint wants to convey.

Even though the study by Kochmar, et al [21], as discussed in the section on personalized feedback (2.2.3), does use Artificial Intelligence to personalize feedback, it does not keep the different feedback types in mind.

3 Methodology

For this research on personalized feedback, an experiment was conducted. In this study, children, between the ages ten to twelve, were asked to interacted with an adjusted pre-existing educational game. Each time they made a mistake, they received feedback generated by the generative model, ChatGPT. This experiment is further explained in the sections below.

3.1 Participants

Child	Age	Gender	City	Coding experience
1	12	F	Groot-Ammers	None
2	10	F	Utrecht	Once, without lessons
3	11	F	Utrecht	None
4	12	F	Delft	Multiple lessons
5	12	F	Delft	Multiple lessons
6	12	F	Delft	Multiple lessons

Table 1: Background information on the participating children

For this study, qualitative data was collected. Table 1 provides an overview of the background information of the participants, including information on their previous programming experience. As can be seen in the table, six Dutch children between the ages of ten to twelve participated in the experiment. All of the children were female. Out of the six children, one lived in Groot-Ammers, two in Utrecht and three in Delft. Two of the participants had never programmed before, one had once independently experimented with Scratch, and three had received lessons in programming in the past.

Too ensure that the puzzles would not be too easy or too difficult, the inclusion criterion for this study was that the participants had to be between the ages of ten and twelve. Four of the participating children were found through personal connections with the researchers and two through connections with the first four children.

3.2 Materials

In order to execute the experiment, two main things were required: a platform for children to learn how to code and a program that can generate personalized feedback. Therefore, an educational game was used in combination with the ChatGPT generative model from OpenAI.

3.2.1 The game

For this research, an already existing game was used. Van Deursen [12] developed this game in order to study the role of instruction methods in programming education. For the purpose of their research, two versions of the game were made, one with puzzles and one without. For this study, only the version with the puzzles was used.

In the game, the student takes on the role of a zoo owner. The player gets to build the zoo from scratch, while learning to program. They can earn money by solving coding puzzles, with which they get to buy enclosures and animals for in the zoo. Additionally, the players can alter the names of the animals and take care of the basic needs of the animals by changing the values of variables.

In the puzzles, the children learn to code in the programming language Python. New programming concepts are gradually introduced by a short explanation, before they are used in the puzzles. The exercises become progressively more difficult to keep the student engaged and to improve their coding skills.

Figure 2: The first puzzle in the game, in which the child has to display the text: "Hallo!"

As can be seen in Figure 2, the first puzzle is relatively easy. The child needs to display "Hallo!" and the print-statement is already given. The child only needs to add quotation marks to solve the puzzle.

Figure 3: The fifth puzzle in the game, in which the child has to display the value of the variable "kommagetal"

For the fifth puzzle, as can be seen in Figure 3, the child needs to be able to understand that the value of a variable can be printed by putting the variable name inside of the print-statement. This shows the increase in difficulty in later puzzles.

3.2.2 Feedback in the game

In the underlying code of the game, a system was implemented for the purpose of this study, which keeps track of the mistakes made by the child and which feedback type to use. The exact working of the system is explained in Section 3.2.3.

Because of the system, each time a child makes a mistake in one of the puzzles, the game displays a red message saying the solution is incorrect. It also includes an abbreviation representing the feedback type that should be used. It is abbreviated, so that the text does not confuse the child, but still allows the researcher to understand which feedback type should be used. The used abbreviations are: EC for Explicit Correction, Rc for Recast, CR for Clarification Request, MC for Metalinguistic Clues, El for Elicitation, and Rp for Repetition. An example of such a message is shown in Figure 4, which translates to: *That's not quite right yet, try again! If you're not sure, click on example or help (EC)*

Figure 4: An example message that is displayed in the game after a mistake is made

3.2.3 The code

The game was made in the Godot game engine and used GDscript as the programming language. A system was added in order to keep track of mistakes made by the child.

In the system, points were assigned to the individual feedback types, starting with one point for each. Then after each mistake and correct answer, the points were updated. After a mistake, points were added and after a correct answer, points were subtracted. To decide which feedback type to use, the system picked the first one with the least points assigned to it.

The formula used when the child made a mistake was as follows:

$$new = old + 0.2 * numt \tag{1}$$

where *new* is the new updated score for that specific feedback type, *old* is the old, not yet updated, score for that specific feedback type and *numt* is the number of times that the child has attempted that specific exercise.

The reason for making the penalty higher for each try at the puzzle, is because the child receives a hint after every failed attempt. The more feedback the participant has available to them, the more they could combine to come to the correct answer. Thus, it should be more likely for the child to finish the puzzle correctly after each new given feedback. Therefore if the feedback did not lead to the correct answer the punishment on the feedback type should be higher than for the previous attempt.

The formula used when the child correctly finished a puzzle was the following:

$$new = old - \frac{1}{numt} \tag{2}$$

The reason for the lower subtraction number after each attempt is again because the child receives feedback after every mistake. If the child has received feedback already, the new hint will contain less new information, and the child reaching the correct solution will more likely be because of a combination of the previous feedback then solely the new one. Therefore, the number of points subtracted from the current feedback type, should be lower than the number of points that would be subtracted from the previous type.

For example, the array containing the points per feedback type could look as follows:

$$[1.5, 1, 1, 1, 1, 1] \tag{3}$$

where the positions in order represent: Explicit Correction, Recast, Clarification Request, Metalinguistic Clues, Elicitation, and Repetition.

As can be conducted from Equation 3, the child has a higher score for the feedback type: Explicit Correction. This means that the system would now make sure to pick the next type in line, to use if the child makes a mistake, in this case that would be Recast. If the child would then solve the puzzle on the second try, after receiving the Recast feedback, points would be substracted from the second position. The amount of points can be calculated using Equation 2, as follows:

$$new = 1 - \frac{1}{2} = 0.5 \tag{4}$$

The new array would then look like the following:

$$[1.5, 0.5, 1, 1, 1, 1] \tag{5}$$

3.2.4 ChatGPT

Multiple ways of using AI to provide personalized feedback were explored. The two main possibilities were the use of AI in deciding the feedback type and the use of it in generating the feedback. However, the first option proved to be outside of the scope for this study.

In order to generate personalized feedback for the children, ChatGPT was used. Specifically, the study made use of the GPT-4o model by OpenAI. This was done to produce feedback using different feedback types. The reason for using ChatGPT, rather than other generative models, was because the GPT-4o model is both capable of understanding code to an adequate level and to producing relatively human-like text. Besides this, it is easy to use, which allowed for the researcher to sit between the AI and the child, while simultaneously observing. Additionally, the only requirement was a free account. There was, however, a limit on the free usage of the GPT-4o model, but this did not pose much of a problem, as the limit was high enough to finish at least one entire session without exceeding it. When exceeded, the model automatically switches to GPT-4o mini, which is still an adequate model.

The model was given a starting prompt in which the used feedback types were explained extensively. This was done to ensure, that the model would use the same definitions for feedback types as defined in this research. The starting prompt was originally in Dutch, but the translation is as follows:

"Here are the possible feedback types I will refer to later. It is not necessary to explain them or do anything with them at this point; I simply want to list them: Explicit correction: the exact error is pointed out, and an exact solution is provided. Recast: an error is implicitly corrected through rephrasing, so the exact solution is not given. Clarification request: it is emphasized that an error has been made by asking for clarification, so the exact solution is not given. Metalinguistic clues: a comment or question is made that hints at the theory needed to correct the error, so the exact solution is not given. Elicitation: an incomplete comment or question is posed that the student must complete, providing a clue about how to correct the error, so the exact solution is not given. Repetition: a comment is fully repeated with the error emphasized."

As can be seen, the prompt starts out by stating that it does not need to do anything with the given information yet, to ensure that it does not respond extensively to the starting prompt.

Each time the participant made a mistake, the model received a new prompt containing the correct code, the code written by the child and the feedback type it was supposed to use. The prompt specifically contained the words "short", "clear", and "child friendly" to make sure that the response would not be too long and overly complicated, and would remain appropriate for children. An example of a prompt, translated from the original Dutch version to English is the following:

"The correct code is: print("Hallo!"). This is the code the child wrote: print(Hallo!). I would like to ask you to provide short, clear, and child-friendly feedback. Use the feedback type: Explicit correction. Please make sure to apply the correct feedback type."

It then produced feedback that was passed on by the researcher to the child, to ensure it to be child friendly. An example of feedback given by ChatGPT, translated from Dutch to English, using the previous prompt is as follows:

"Your code isn't working quite right because you forgot to put the text in quotation marks. The correct code is: print("Hallo!"). Try again with this solution!"

3.3 Experiment

The experiment was approved by the ethical committee, before it was conducted. Additionally, all of the participating children signed an informed assent form and their parents an informed consent form. All of the sessions were done at the homes of the children, with a parent present at all times.

The children participated individually in the study. First, they started out by answering questions about their name, age, gender, school, the grade they are currently in and the city they live in. It also included questions about there previous programming experiences. This part of the research took up around five minutes per child. Second, after having answered those, they were introduced to the game by the researcher and by following the starting instructions in the game. Third, they played the game for around half an hour, and each time they made a mistake in a puzzle, the researcher interfered to give them their feedback. Finally, the researcher interviewed them for around five minutes to ask them how they experienced the game, the puzzles and the feedback. A few questions were prepared beforehand, but it was mostly a conversation in order to asses their opinion, based on observations during the experiment.

During the experiment, the researcher switched between two separate roles, the observer and the coordinator. At the beginning, the researcher took on the role of the coordinator to ask the background questions and to introduce the child to the game. After that, the researcher became the observer. That is, until the child made a mistake, because then the researcher switched temporarily back to the coordinator role. Whenever that happened, the researcher looked at the displayed feedback type and put the prompt with that type in the model. After the model was done generating the feedback, the researcher first ensured that the feedback was child-friendly, before letting the child read it. The focus was specifically on making sure that the produced text was not offensive in any way to the child. After having done that, the researcher switched back to the observer role. At the end of the experiment, the researcher became the coordinator again to conduct the interview with the child.

While the researcher was observing, they wrote observations down in four different categories: performance, comprehension, attention and enthusiasm. The first category contained observations such as: "the participant made a similar mistake for the third time", and "the participant quickly knew the correct answer". The comprehension category had observations such as: "the participant looks confused", and "the participant makes a sound suggesting they understand the question". The third category, attention, contained observations such as: "The participant is reading the instructions carefully", and "the participant is looking away from their screen". The last category, enthusiasm, included observations such as: "the participant is looking frustrated", and "the participant makes sounds suggesting excitement".

3.4 Analysis

Firstly, the answers on the background questions containing the basic information of the children and their previous programming experience, were pseudomysed. After that, they were converted to a table. Then, the collected qualitative data was analyzed using a thematic analysis to find patterns in the data. This analysis was done on the combination of the observational data and interview responds of each child. It was done in different stages based on the article by Braun and Clarke [8].

The main purpose of the first stage was for the researcher to familiarize themselves with the data. This consisted of reading through the data and noting down interesting ideas. In the second stage, initial codes were generated, with a total of thirteen (for example: feedback in relation to future mistakes, enjoyment of game, etc.). Then, in the third stage, themes were found based on the codes. This was done by grouping similar codes together to form an overarching theme (for example: enjoyment of game, enjoyment of programming, mood during experiment and focus on program, formed one theme). The fourth stage is meant to perfect the formed themes, so this is when the researcher reviewed them. It was investigated how the themes differ from each other and in what way they are similar, essentially finding the overall story of the data. In the fifth stage, the themes were named based on their fundamental characteristics (for example: the theme from the last example became: the satisfaction and enjoyment from solving puzzles and achievement). Lastly, a report was produced, for each child separately in Section 4.3. This was when the theme names were altered, as some already contained a form of interpreting which does not align with the purpose of the results section (for example: the theme from the last example became: engagement and enjoyment).

The experiment was done in Dutch, thus all of the quotes that can be found in the results, were translated to English.

4 Results

The first part of this results section will focus on the general findings during the experiment. After that, the results in relation to the exploration of the use of Artificial Intelligence will be addressed. Finally, the outcomes of all of the children individually will be discussed, through a thematic analysis.

4.1 General results

Several general results and observations will be addressed below. This includes information about the duration of each of the experiments, an overview of the mistakes per puzzle, and some unforeseen difficulties that occurred.

First, for each session, the participating child played the game around 30 minutes, as planned. The only exception was Child6, because she struggled with solving the last puzzle, as can be seen in Figure 9 and Table 8, which led to a longer playing time.

Second, Figures 5 to 10 show the amount of mistakes made before solving the puzzles for each of the children. Most puzzles were solved in a relatively similar amount of tries, with an exception of puzzle 5. As can be seen in Figure 9, it took Child6 ten tries to come to the right solution, which is further discussed in the section on Child6 in the thematic analysis per child (4.3.6). All of the children completed puzzles one to five, but some additionally finished the sixth one. The children that did not reach Puzzle 6 are not included in Figure 10.

Third, there was one mistake that multiple children made that required an intervention from the researcher. This was because the children were not aware that the computer differentiates between using single quotation marks twice in a row, and using double quotation marks once. This mistake only occurred in the first puzzle and after the intervention none of the children made the mistake again. This intervention is indicated in Figure 5 as half a mistake and is indicated in the Tables later on with the abbreviation "Int".

Finally, some children had the tendency to quickly click the check answer button multiple times after making a mistake. This led to unwanted changes in the scoring of the feedback types.

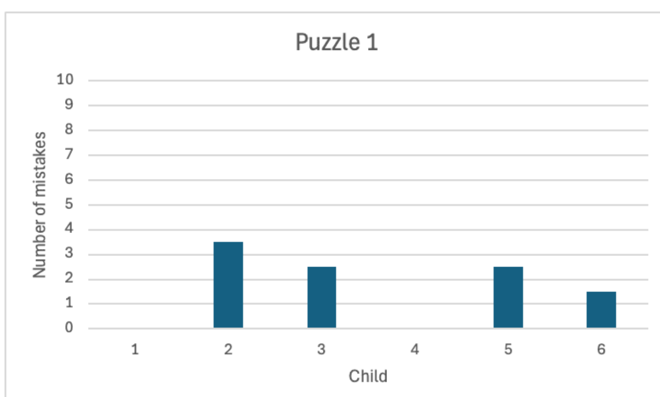

Figure 5: The amount of mistakes each child made before solving the first puzzle

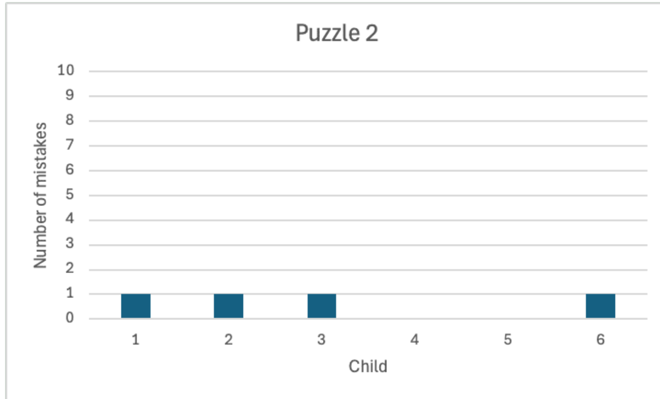

Figure 6: The amount of mistakes each child made before solving the second puzzle

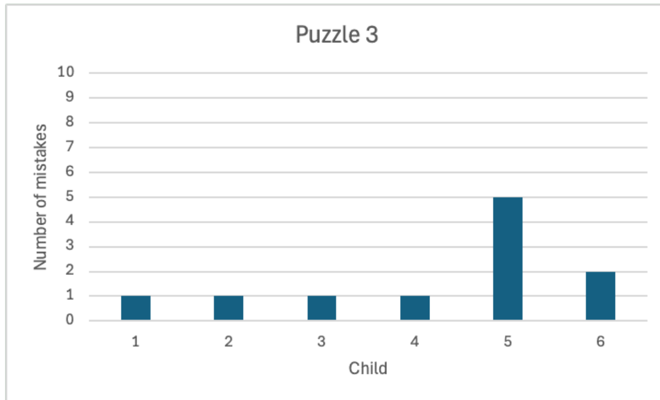

Figure 7: The amount of mistakes each child made before solving the third puzzle

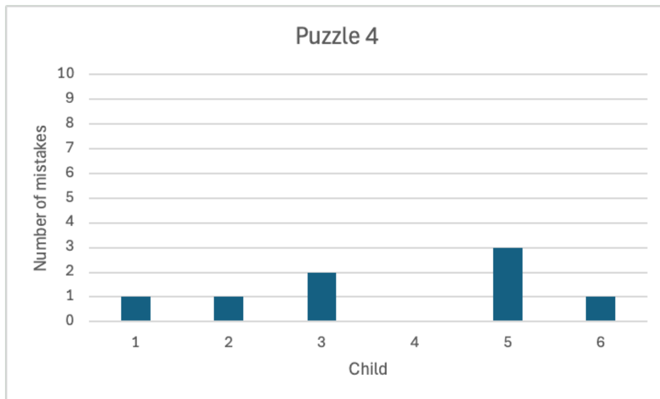

Figure 8: The amount of mistakes each child made before solving the fourth puzzle

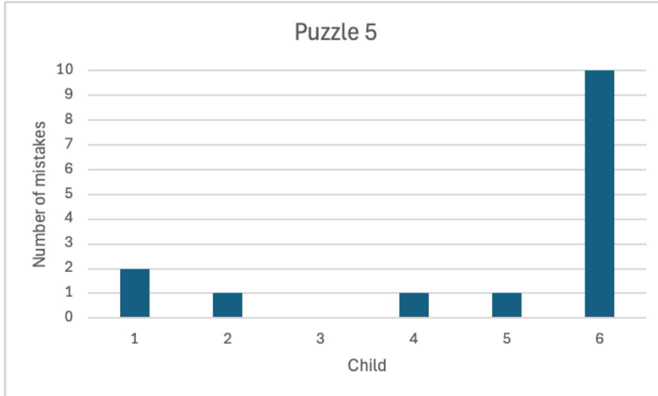

Figure 9: The amount of mistakes each child made before solving the fifth puzzle

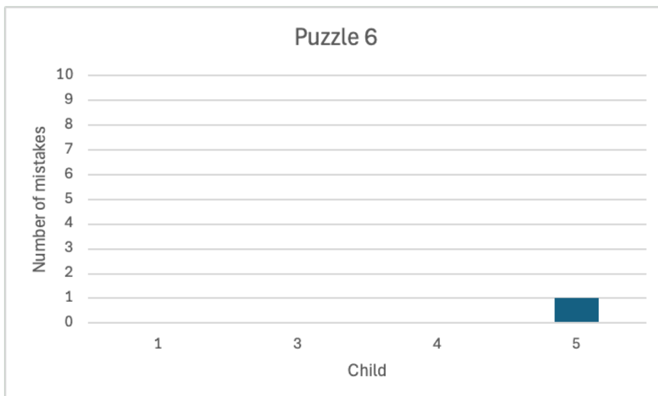

Figure 10: The amount of mistakes each child made before solving the sixth puzzle

4.2 Artificial Intelligence

This section discusses the experience of using Artificial Intelligence for personalized feedback in this way, including several general observations and an issue with the used model.

First, the researcher did not observe any offensive language in the generated feedback. All of it was child-friendly, and therefore there was, at no point during any of the experiments, a need for the researcher to intervene. Second, because the generative model was separate from the game, the prompts needed to be filled in manually, which took up around fifteen seconds each time. Third, although the feedback was overall clear and accurate, there were exceptions. An example of unclear feedback is shown in Figure 11. The prompt translates to: *The correct code is: 'naam = "Jan"'. This is the code the child wrote: 'naam = ('Jan')'. I would like to ask you to provide short, clear, and child-friendly feedback. Use the feedback type: Explicit correction. Please make sure to apply the correct feedback type.* and the feedback translates to: *Good try! But you need to use double quotation marks instead of single ones. The correct code is: naam = "Jan". Try it in this way!* As can be seen, the incorrect quotation marks are specifically explained, while the misuse of the parentheses is not mentioned. Even though the exact solution is given, the child can easily miss that the parentheses are removed.

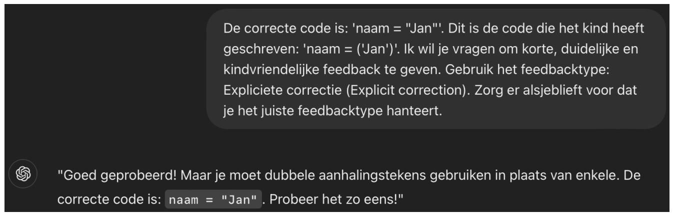

Figure 11: An example of unclear feedback

Fourth, the researcher observed that all of the provided feedback followed the form of the required feedback type accurately. Finally, the last child received some feedback from the GPT-4o mini model rather than the GPT-4o model. This occurred, because the sessions with Child5 and Child6 were done back to back. Since they both required quite some feedback and there is a limit on the free usage of the GPT-4o model, it switched models towards the end of the experiment.

4.3 Thematic analysis per child

4.3.1 Child1

Puzzle number	1	2	3	4	5		6
Feedback type	-	EC	EC	CR	MC	Rp	-

Table 2: The different feedback types that the child received per puzzle, where EC stands for Explicit Correction, CR for Clarification Request, MC for Metalinguistic Clues, and Rp for Repetition

As can be seen in Table 2, the child completed puzzles 1 to 6 and she received feedback a total of five times.

A few times following a mistake, the child pressed the check answer button multiple times in a row, rather than just once. This led to the program registering it as a new mistake, leading to unwanted changes in the scores for each feedback type in the underlying system.

The child consistently used the, by the game provided, examples and additional instructions, often copying the exact example without changing the value.

The clarity and usefulness of feedback The participant mentioned that some of the feedback worked better for her than others. Overall, she considered the feedback to be “pretty good” and helpful. Table 2 shows that the child mostly solved her puzzles in one try after receiving feedback, with the exception of puzzle 5. She did specifically voice her confusion after receiving the Metalinguistic Clue, saying that she did not find the hint clear. The researcher observed that the child had more trouble understanding the implicit feedback than the explicit ones.

When asked about the effectiveness of the feedback over time, she mentioned a decline. Furthermore, the feedback seemed to have little impact on future puzzles, as the participant often made similar mistakes.

The general comprehension and difficulty The participant said to have found the puzzles a little challenging. With the exception of the first and last puzzle, this was also observed during the experiment, as the child often made use of the examples and extra instructions that were available in the program.

The engagement and enjoyment The child expressed her enthusiasm about both the game and the programming side of the puzzles. During the experiment, the researcher observed that the enthusiasm slightly decreased for a while, until a new animal was unlocked. Even when the excitement was temporarily somewhat lower, the child remained focused on the game the entire duration of the experiment.

The willingness to continue The child expressed a clear enthusiasm both for re-engaging with the game and for programming again in the future. When asked if she wanted to end the experiment, she seemed reluctant and replied that she first wanted to buy another animal.

4.3.2 Child2

Puzzle number	1				2	3	4	5
Feedback type	EC	MC	El	Int	Rp	El	Rc	Rc

Table 3: The different feedback types that the child received per puzzle, where EC stands for Explicit Correction, Rc for Recast, MC for Metalinguistic Clues, El for Elicitation, and Rp for Repetition. Int represents an intervention by the researcher as explained in Section 4.1

As can be seen in Table 3, the child completed puzzles 1 to 5 and she received feedback a total of eight times.

A few times following a mistake, the child pressed the check answer button multiple times in a row, rather than just once. This led to the program registering it as a new mistake, leading to unwanted changes in the scores for each feedback type in the underlying system.

The clarity and usefulness of feedback The participant found the feedback good and helpful. She also stated that she experienced the feedback to be increasingly more helpful over time. The first puzzle took the child slightly longer due to the fact that she was not aware that the computer differentiates between a single quotation mark and a double quotation mark. The child completed the rest of the puzzles quite smoothly. As can be seen in Table 3, the rest only required feedback once. Furthermore, the researcher observed that similar mistakes were no longer made after receiving feedback on them.

The general comprehension and difficulty The participant considered the puzzles to be quite difficult. However, she did note that she felt that she should have read the instructions more carefully.

The engagement and enjoyment The participant noted her enjoyment of the game, and when asked what she enjoyed most, her answer was: "The puzzles were fun and the zoo was fun!". The child had her full attention on the game the entire duration of the experiment and seemed to be excited and interested most of the time. The researcher did find that the participant looked a little defeated and frustrated at times, after she had made a mistake. However, the child did not note any feelings of frustration or defeat.

4.3.3 Child3

Puzzle number	1			2	3	4		5	6
Feedback type	Rc	MC	Int	Rc	Rc	Rc	CR	-	-

Table 4: The different feedback types that the child received per puzzle, where , Rc for Recast, CR for Clarification Request, and MC for Metalinguistic Clues. Int represents an intervention by the researcher as explained in Section 4.1

As can be seen in Table 4, the child completed puzzles 1 to 6 and she received feedback a total of seven times.

A few times following a mistake, the child pressed the check answer button multiple times in a row, rather than just once. This led to the program registering it as a new mistake, leading to unwanted changes in the scores for each feedback type in the underlying system.

The clarity and usefulness of feedback The participant found the feedback useful the entirety of the experiment and did not note any differences over time in terms of usefulness. The researcher observed two types of mistakes that were made repeatedly by the child, even after receiving feedback on it and fixing it correctly. The first one was having unnecessary spaces in spots, which this specific program was sensitive to. The other one was printing the value of the variable directly, instead of using the variable name in which the value was already stored.

The general comprehension and difficulty The participant noted that she found the puzzles to be quite difficult, which she explained to be mostly due to her not fully comprehending the instructions. The child found it difficult to know what to do exactly in the beginning, but as she progressed further through the game, she began to understand programming better. As can be seen in Table 4, she completed the last two puzzles in her first try without needing feedback.

The engagement and enjoyment When the participant was asked for her opinion on the game, she answered that she did not quite now what to do in the beginning yet, but later on she found it really enjoyable, because she liked that she could solve puzzles and then use that to buy new things for the zoo. The researcher noticed an increase in excitement later in the game, supporting her statement.

4.3.4 Child4

Puzzle number	1	2	3	4	5	6
Feedback type	-	-	EC	-	EC	-

Table 5: The different feedback types that the child received per puzzle, where EC stands for Explicit Correction

As can be seen in Table 5, the child completed puzzles 1 to 6 and she received feedback a total of two times.

This child had previous experience in coding with the programming language Python, which she had received lessons for, one time a week, for half a year. She mentioned that she had not really enjoyed it, but it was a mandatory part of a program that she followed in school.

The clarity and usefulness of feedback The participant noted that the two times she received feedback it showed her exactly how to do it. As can be seen in Table 5, the child received Explicit Correction twice, which points out the exact answer and gives the correct solution. This allowed her to complete the puzzle immediately. She also added that it helped her understand how to answer similar questions later on. Therefore, she found the feedback helpful. The observer also noted that the child had clearly understood and learned from her mistakes.

The general comprehension and difficulty The participant had little trouble going through the puzzles. She noted that she did not read all of the questions carefully enough, but that the rest of them were correct on the first try.

The engagement and enjoyment The participant found that the game was fun and mentioned that once you start, you do not want to stop, because you want to reach 100% on the progress bar. She noted that she did not find any part of the game frustrating. She did, however, say that she would have found it frustrating if she had not been able to play to 100%.

The willingness to continue When the participant was given the option to stop, she did continue playing the game itself for a while, by buying new enclosures and animals for example. However, when asked if she would consider playing the game again, she answered the following: "Not in my free time, because I prefer watching series, and stuff." She did add that she would consider doing it again if it is for school.

4.3.5 Child5

Puzzle number	1			2	3					4			5	6
Feedback type	EC	CR	Int	-	CR	Rp	CR	EC	MC	MC	CR	El	EC	EC

Table 6: The different feedback types that the child received per puzzle, where EC stands for Explicit Correction, CR for Clarification Request, MC for Metalinguistic Clues, El for Elicitation, and Rp for Repetition. Int represents an intervention by the researcher as explained in Section 4.1

As can be seen in Table 6, the child completed puzzles 1 to 6 and she received feedback a total of thirteen times.

A few times following a mistake, the child pressed the check answer button multiple times in a row, rather than just once. This led to the program registering it as a new mistake, leading to unwanted changes in the scores for each feedback type in the underlying system.

The clarity and usefulness of feedback Overall, the participant found the feedback helpful and understandable. The child noted however, that in the beginning she did not quite comprehend the tips yet. She found that the feedback became increasingly clearer, and therefore increasingly more helpful. As can be seen in Table 6, the child needed much feedback for puzzle 3, but required it only once for the last two puzzles.

Although, the child often repeated similar mistakes, they were resolved in less tries than before.

The general comprehension and difficulty The child found the game fairly difficult, because she kept "forgetting quotation marks and brackets" and noted that it might have been easier if she had remembered them better.

The engagement and enjoyment The participant expressed great enthusiasm about the game. She thought it was very enjoyable, because, in her words: "It felt like I could keep going endlessly".

Somewhat at the beginning of the experiment, the child asked the researcher, what she was allowed to do in the game. The researcher explained that she was allowed to choose herself, which led the child to enthusiastically choose a puzzle. She also noted to not have perceived any part of the experiment as frustrating. The researcher, in contrast, did observe slight feelings of frustration, after repeatedly making a mistake in the fourth puzzle.

The willingness to continue The participant expressed with great excitement that she would gladly play the game again some time and she was quite disappointed when the time for the experiment was up.

4.3.6 Child6

Puzzle number	1		2	3		4
Feedback type	EC	Int	EC	EC	Rc	Rc

Table 7: The different feedback types that the child received for puzzles 1 to 4, where EC stands for Explicit Correction, and Rc for Recast. Int represents an intervention by the researcher as explained in Section 4.1.

Puzzle number	5									
Feedback type	Rc	Rc	CR	MC	El	EC	Rc	CR	MC	Rp

Table 8: The different feedback types that the child received for puzzle 5, where EC stands for Explicit Correction, Rc for Recast, CR for Clarification Request, MC for Metalinguistic Clues, El for Elicitation, and Rp for Repetition

As can be seen in Table 7 and Table 8, the child completed puzzles 1 to 5 and she received feedback a total of sixteen times. Some of the last feedback that the child received in puzzle 5 came from the GPT-4o mini model instead of the GPT-4o model, since the limit of free usage was reached.

The clarity and usefulness of feedback The participant found that the feedback helped a little. According to her, the tips were "sometimes a bit unclear, but at other times, they were clear." The child did add that they became more helpful over time. Additionally, the child noticed that the feedback helped her avoid similar mistakes later on, which the researcher had observed as well.

As can be seen in Table 8, the child received feedback ten times for puzzle 5. All of the possible feedback types were given at least once, but it took a while before she managed to solve the puzzle.

The general comprehension and difficulty The participant considered most of the puzzles to be "not very difficult", except for the last one, which she found extremely hard. This can be seen as well, when looking at Table 7 and Table 8. The child additionally noted that at the end of the experiment she felt that she understood coding better.

The engagement and enjoyment Although the participant found it frustrating when she kept making mistakes in the last puzzle, she enjoyed the game regardless. She noted that this is because she likes programming in general and because she also liked the animals in the game. The child exclaimed "Yes, finally!", when finishing the final puzzle.

The full attention of the participant was on the game the entirety of the experiment. This even remained when she was frustrated.

The willingness to continue The participant mentioned that she would like to do the game again in the future, partly because she feels that she understands it better now.

5 Discussion

The aim of this research was to study the influence of personalized feedback on the coding learning process on children between the ages of ten to twelve, by using an educational game. This was partly explored by looking at how an educational game can provide personalized feedback to a child. Specifically, the use of Artificial Intelligence was studied. Additionally, this was researched by looking at how personalized feedback could enhance the learning progress and experience of a child in the field of computer science.

The results of the experiment showed that every child found the feedback at least a little helpful. Three out of the six children noticed an increase in the helpfulness of the feedback, two did not feel it changed in a specific direction, while one saw a decrease. Out of the participants, four considered the puzzles to be quite difficult, one found it considerably easy and one found only Puzzle 5 very difficult. Overall, everyone enjoyed the game. Some found the game entertaining, because of the possibility for progress in the game, others because of the puzzles and a few for both of those reasons. Three of the children expressed their enthusiastic want for re-engaging with the game in the future and one would not consider doing that in their spare time.

The results will be interpreted in this discussion and this will be done in relation to the following research questions:

RQ: In what way can personalized corrective feedback using an educational game influence the coding learning process of children around the age of ten to twelve?

SQ1: In what way can an educational game be used to provide personalized feedback to a child?

SQ2: In what way does personalized feedback enhance the learning progress and experience of a child in the field of computer science?

5.1 Educational game

To research in what way an educational game could be used to provide personalized feedback, an existing educational game was used and build further upon.

5.1.1 The engagement and enjoyment

All of the children said to have enjoyed the game and only one admitted to frustration during the experiment. The child noted that they got frustrated after repeatedly making mistakes in the same puzzle. The researcher, however, did observe little feelings of frustration in other participants. This contradiction could be due to multiple reasons. For one, the researcher could have misinterpreted the body language and signs that the child was showing. Another explanation could be the rosy retrospection bias of the child. This bias can be defined by the tendency of people to recall memories in a more positive light than how they viewed them at that moment of time, which was first researched by Mitchell, et al. [26]. Furthermore, as discussed by Hall et al. [19], children have the tendency to respond overly positive when evaluating in a research.

5.1.2 The willingness to continue

Although all children said to have enjoyed the game, only half of them expressed a want to re-engage with the game and one even specifically mentioned they would not do so in their free time. The reason for this contrast could be related to the participants comment about wanting the progress bar to be at 100%. During goal-oriented tasks, humans release dopamine, which has the neurological effect of an internal reward [1]. The first prove of dopamine release in humans, during such tasks was given by Koepp, et al. in [22]. Because of this internal reward, the child that did not want to re-engage with the game could have experienced the game to be enjoyable while it lasted, even though they might not have found the game itself particularly fun. The child did also mention their disinterest in programming based on past experience. That being said, this does imply that an educational game can be used to make the experience of learning to code enjoyable, even when programming does not interest the person.

Furthermore, multiple children commented on the desire to continue the game to achieve new things or to reach 100% on the progress bar. This again could be explained by the dopamine releases in the brain. Therefore, it could be interesting to further research the role of dopamine in the experience of a child related to learning activities and if that can be leveraged to additionally increase their learning progress.

5.2 Artificial Intelligence

To explore the possibilities of Artificial Intelligence in providing personalized feedback, this experiment used a combination of an educational game and a separate AI model. This section interprets the results of that implementation, including the ethical concerns, the specific set-up and the clarity.

First, for this experiment, there was a researcher in between the AI and the child to ensure that potential harmful generated text would not be seen by the child. Although this is important, it does take away the labor-free advantage of using Artificial Intelligence for personalized feedback. During this experiment, all off the generated feedback by ChatGPT was evaluated as child-friendly by the researcher. However, this does not ensure that it will not happen in any future interactions between children and AI. Research done by Firmino Pinto [15] showed situations in which children were exposed to generated text that was not child-friendly, even when it was mentioned that the interaction included a child. The study was done on different models than the OpenAI GPT-4o model. For those models, it was unclear what data the LLM (Large Language Model) was trained on. Nevertheless, the risk cannot be entirely excluded.

Second, because it took time to fill in the prompts each time, despite the fact that it was only around fifteen seconds, the work flow and focus of the child was continuously interrupted. A study done by Yuan & Zhong [39], showed an increase in Cognitive Load when participants were interrupted during their tasks. As talked about in the related work section on Cognitive Load Theory (2.1.3), the cognitive load cannot be more than the capacities of the working memory allow. If it does exceed those capacities, it negatively impacts the learning process [34] [27]. As previously discussed as well, learning to code already requires a high cognitive load, so a further increase should be prevented. For this reason, later research should study programs in which the AI is directly linked to the educational game. This does, however, bring back the ethical concerns as discussed in the previous paragraph.

Third, the feedback was not always clear. Therefore, further research could look at the possibility for the child to actively interact with an AI through conversation, since this allows for the child to ask questions. However, similar as before, this introduces ethical challenges, regarding the lack of ensured child-friendliness by the model.

Finally, the feedback was always in the form of the requested type, which would have been difficult for a human to produce in the little time that the AI required. As discussed in the section on possibilities of Artificial Intelligence in the related work (2.3.4), there is an absence of research that combines pedagogical theories, such as feedback types, with automated feedback. These results suggest advantages of using Artificial Intelligence to combine these two concepts.

5.3 Personalized feedback

As discussed in the related work section (2), previous research shows the importance of feedback on the learning process of children, especially in the field of computer science. Additionally, because the most effective type of feedback can differ per person, it is interesting to look at the impact of personalized feedback. This section will discuss the interpretations of the results regarding the personalized feedback, including the clarity and usefulness, and the general comprehension and difficulty of the feedback.

5.3.1 The clarity and usefulness of feedback

First, the researcher observed that for two of the participants, the feedback did not help to prevent similar mistakes in the future. For one of the participants, the feedback helped slightly, as she did resolve her similar mistakes faster than the first time she had made them. The other three showed clear improvement over time. A possible reason for the seemingly increased coding skills of some of the children, could be that the feedback reduced the cognitive load. As talked about in the Cognitive Load Theory section of the related work (2.1.3), presenting information in a clear way, which in this case was the feedback, reduces cognitive load [27].

One of the children, for which the feedback did not seem to prevent future mistakes, had the tendency to copy the exact solution when given. Furthermore, she had trouble understanding the feedback, if the correct solution was not provided to her. A possible explanation for this is that the explicit feedback might not have motivated her to truly understand the correct answer, leading to similar mistakes being made in the future. In other words, because the feedback type changed whenever she made a mistake, it could have led to a preference for short term gain, at the expense of long-term improvement. Further research could look into ways to promote long-term benefits for learning using personalized feedback and how that impacts the rest of the learning experience.

Second, three out of the six children mentioned an increase in the helpfulness of the feedback, two did not feel it changed in a specific direction, while one saw a decrease. However, when looking at Tables 2 to 8, four of the children required a similar amount of feedback for all of the puzzles, not taking in account Puzzle 5 of Child6, and only two showed an actual decrease in the number of mistakes per puzzle. The overly positive view on the feedback could, once again, be explained by the rosy retrospective bias [26] of the children, and their tendency to respond more positively when evaluating in a research, as discussed previously in the section on the educational game (5.1). A possible reason that the effectiveness of the feedback rarely changed over time, could be, because some of the children messed up the underlying point system for the feedback types, by pressing the "ready" button too often.

Finally, all of the children did mention in the interview that overall they perceived the feedback as helpful or useful. However, none of the children noted that the feedback enhanced the general experience of the game for them.

5.3.2 The general comprehension and difficulty

First, four of the children mentioned in the interview that they found most puzzles difficult. One mentioned that they found only the last one difficult and one had little trouble with any of the puzzles. Out of the four children that found the puzzles difficult, one child received relatively little feedback compared to the other children. Although this could suggest that the feedback was more helpful for her than for the others, it is likely caused by her common usage of the "help" and "example" button.

Furthermore, most of the children made a somewhat similar amount of mistakes for each of the puzzles, with the exception of Child6 during Puzzle 5, for which they required feedback ten times. However, this likely had little to do with the feedback itself, as they had received all possible types at least once, and since some of those contained the exact correct solution. The actual reason for the problem, presumably, has more to do with the child itself and the cognitive processes in the brain. A possible theory is that all of the new concepts and information led to a cognitive overload as discussed in Section 2.1.3.

5.4 Limitations

One of the main factors that could have influenced the results of this research, was that some of the children pressed the check answer button multiple times in a row, after making a mistake. This caused the underlying point system for the personalized feedback to change undesirably. This occurred, because the researcher could not step in quick enough to prevent it. Even when specifically asked not to do this, the situation kept occurring, likely due to reflexes of the children. In the future this could be prevented by implementing a bigger warning in the program or by integrating a timer that prevents multiple button presses in a row.

Furthermore, in terms of variety in the participants, this study was limited, as all of the children were female, most of them were twelve years old and most of them were already studying in secondary school.

For this study, the researcher made use of a separate AI model to generate personalized feedback. This took up extra time during the experiment and interrupted the general work flow of the child more than needed. For this thesis, it was out of scope, however, future research could benefit from integrating AI into the game itself.

Lastly, this was a qualitative study done on a small amount of children. For that reason, they were all given the exact same game. However, if done on a bigger group, part of the participants could receive a game with only non-personalized feedback, so that the results can be compared. Furthermore, this study could in general be done on a larger scale.

5.5 Conclusion

This thesis aimed to study the influence of personalized feedback on the coding learning process of children between ten to twelve. Specifically, this was done focusing on how an educational game can be used to provide personalized feedback to a child, and in what way personalized feedback enhances the learning progress and experience of a child in the field of computer science.

The results showed that all of the participating children enjoyed playing the educational game, with some even expressing a desire to re-engage with it in the future. Furthermore, only one child perceived a decline in helpfulness of the personal feedback, which could be due to the child choosing short-term benefits over long-term ones.

To answer **SQ1**: This study indicates possible advantages using Artificial Intelligence in an educational game to provide personalized feedback, although there are still ethical concerns that should be studied further. Later research should also explore the effects of integrating the AI with the educational game. Additionally, the research suggests that the use of achievements in the game, can lead to more overall enjoyment. Because of this, the children are more likely to re-engage with the game, thus interacting with the personalized feedback more and improving their coding skills.

To **SQ2**: Some of the participants did seem to learn, as they ended up making similar mistakes less. However, this research did not find clear evidence that the personalized feedback was the direct cause of the improvement in the learning progress of some of the children. Similarly, there is little evidence to be found in the results that the personalized feedback caused an enhanced experience for the child that reached further than the game and the puzzles itself.

Finally, to answer the main research question, combining the answers to the subquestions: This research suggests advantages of using an educational game to provide personalized feedback, specifically through the use of AI. Especially, the use of achievements in the game can enhance the experience for the child. This study did seem to show an increase in the coding skills of some of the children, allowing for the possibility to remain that the personalized feedback caused that. Although inconclusive, these results indicate that further research into implementing personalized feedback, specifically through the use of AI, could lead to new positive insights.

References

[1] Dopamine: What It Is, Function & Symptoms — my.clevelandclinic.org. https://my.clevelandclinic.org/health/articles/22581-dopamine, 2022.

[2] Abesadze, S., and Nozadze, D. Make 21st century education: The importance of teaching programming in schools. *International Journal of Learning and Teaching 6*, 3 (2020), 158–163.

[3] ALSAWAIER, R. S. *THE EFFECT OF GAMIFICATION ON STUDENTS'ENGAGEMENT AND MOTIVATION IN THREE WSU COURSES.* PhD thesis, WASHINGTON STATE UNIVERSITY, 2018.

[4] Astia, M. Corrective feedback in english class. Master's thesis, Mulawarman University, Samarinda, 2018.

[5] Bakker, T. Hedy programming language: Introducing the gradual feedback model (gfm). Master's thesis, Leiden University, LIACS, Leiden, 2022. Thesis Master ICT in Business and the Public Sector.

[6] Barone, R. What is coding for kids in 2024?: Benefits, best programs classes, Apr 2021.

[7] Beck, J., Stern, M., and Haugsjaa, E. Applications of ai in education. *XRDS: Crossroads, The ACM Magazine for Students 3*, 1 (1996), 11–15.

[8] Braun, V., and Clarke, V. Using thematic analysis in psychology. *Qualitative Research in Psychology 3*, 2 (2006), 77–101.

[9] Buckley, P., and Doyle, E. Gamification and student motivation. *Interactive Learning Environments 24*, 6 (2014), 1162–1175.

[10] Dahlstrøm, C. Impacts of gamification on intrinsic motivation. *Education and Humanities Research* (2012), 1–11.

[11] De Freitas, S. Are games effective learning tools? a review of educational games. *Journal of Educational Technology & Society 21*, 2 (2018), 74–84.

[12] Deursen, A. v. A. The role of instruction method in an educational game, 2024. Thesis Bachelor Informatica.

[13] Du Boulay, B. Some difficulties of learning to program. In *Studying the novice programmer.* Psychology Press, 2013, pp. 283–299.

[14] Ellis, R., Loewen , S., and Erlam , R. Implicit and explict corrective feedback and the acquisition of l2 grammar. *Studies in Second Language Acquisition 28* (06 2006), 339 – 368.

[15] Firmino Pinto, S. *AI Friend? Risks, Implications, and Recommendations on Generative AI for Children.* PhD thesis, Geneva Graduate Institute, 10 2024.

[16] Flowers, T., Carver, C., and Jackson, J. Empowering students and building confidence in novice programmers through gauntlet. In *34th Annual Frontiers in Education, 2004. FIE 2004.* (2004), pp. T3H/10–T3H/13 Vol. 1.

[17] Gomes, A., Carmo, L., Bigotte, E., and Mendes, A. Mathematics and programming problem solving. In *3rd e-learning conference–computer science education* (2006), pp. 1–5.

[18] Green, C. S., and Bavelier, D. Exercising your brain: a review of human brain plasticity and training-induced learning. *Psychology and aging 23*, 4 (2008), 692.

[19] Hall, L., Hume, C., and Tazzyman, S. Five degrees of happiness: Effective smiley face likert scales for evaluating with children. In *Proceedings of the the 15th international conference on interaction design and children* (2016), pp. 311–321.

[20] Kaplancali, U. T., and Demirkol, Z. Teaching coding to children: A methodology for kids 5+. *International Journal of Elementary Education 6*, 4 (2017), 32–37.

[21] Kochmar, E., Vu, D. D., Belfer, R., Gupta, V., Serban, I. V., and Pineau, J. Automated personalized feedback improves learning gains in an intelligent tutoring system. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21* (2020), Springer, pp. 140–146.

[22] Koepp, M. J., Gunn, R. N., Lawrence, A. D., Cunningham, V. J., Dagher, A., Jones, T., Brooks, D. J., Bench, C. J., and Grasby, P. Evidence for striatal dopamine release during a video game. *Nature 393*, 6682 (1998), 266–268.

[23] Learning Resources. 5 reasons why coding is important for young minds, 2018. Accessed: 2024.

[24] Miller, G. A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review 63*, 2 (1956), 81.

[25] Mironova, O., Amitan, I., and Vilipõld, J. Programming basics for beginners: Experience of the institute of informatics at tallinn university of technology. In *2017 IEEE Global Engineering Education Conference (EDUCON)* (2017), IEEE, pp. 735–739.

[26] Mitchell, T. R., Thompson, L., Peterson, E., and Cronk, R. Temporal adjustments in the evaluation of events: The "rosy view". *Journal of Experimental Social Psychology 33*, 4 (1997), 421–448.

[27] Paas, F., Renkl, A., and Sweller, J. Cognitive load theory and instructional design: Recent developments. *Educational psychologist 38*, 1 (2003), 1–4.

[28] Piaget, J. *Play, dreams, and imitation in childhood.* W.W. Norton, 1962.

[29] Pérez-Segura, J. J., Ruiz, R. S., González-Calero, J. A., and Cózar-Gutiérrez, R. The effect of personalized feedback on listening and reading skills in the learning of efl. *Computer Assisted Language Learning 35*, 3 (2020), 469–491.

[30] Saeli, M., Perrenet, J., Jochems, W. M., and Zwaneveld, B. Teaching programming in secondary school: A pedagogical content knowledge perspective. *Informatics in education 10*, 1 (2011), 73–88.

[31] Schorsch, T. Cap: an automated self-assessment tool to check pascal programs for syntax, logic and style errors. In *Proceedings of the Twenty-Sixth SIGCSE Technical Symposium on Computer Science Education* (New York, NY, USA, 1995), SIGCSE '95, Association for Computing Machinery, p. 168–172.

[32] Shute, V. Focus on formative feedback. *Review of Educational Research 78* (03 2008), 153–189.

[33] Storte, D., Webb, M., Bottino, R., Passey, D., Kalas, I., Bescherer, C., Smith, J., Angeli, C., Katz, Y., Micheuz, P., et al. Coding, programming and the changing curriculum for computing in schools. In *Report of UNESCO/IFIP TC3 Meeting at OCCE, Linz, Austria* (2019).

[34] Sweller, J., Van Merrienboer, J. J., and Paas, F. G. Cognitive architecture and instructional design. *Educational psychology review 10* (1998), 251–296.

[35] Tedick, D. J. Research on error correction and implications for classroom teaching. Master's thesis, University of Minnesota, Minnesota, 1998.

[36] Turan, S., and Aydoğdu, F. Effect of coding and robotic education on pre-school children's skills of scientific process. *Education and Information Technologies 25*, 5 (2020), 4353–4363.

[37] van Algemene Zaken, M. Digitale geletterdheid op school, Apr 2024.

[38] van Amerongen, R. Error message needs of novice programmers in gradual programming languages, 2023. Thesis Bachelor Informatica & Economie.

[39] Yuan, X., and Zhong, L. Effects of multitasking and task interruptions on task performance and cognitive load: considering the moderating role of individual resilience. *Current Psychology* (2024), 1–11.

[40] Zhai, X., Chu, X., Chai, C. S., Jong, M. S. Y., Istenic, A., Spector, M., Liu, J.-B., Yuan, J., and Li, Y. A review of artificial intelligence (ai) in education from 2010 to 2020. *Complexity 2021*, 1 (2021), 8812542.