Understanding the Strategy Selection of Primary School Students in a Block Based Programming Environment

by

Hasan Ertuğrul Çinar

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of
Master of Science

Sabancı University
July 2024

Acknowledgements

First an foremost, I want to express my sincere gratitude to Prof. Dr. Yücel Saygın with my respectful regards for guiding me throughout my Master's experience along with being kind to me at all times. It was a pleasure to work with him on this thesis.

For providing me the opportunity to cooperate with them and helping me on this thesis, I also want to thank Prof. Dr. Seda Ertaç Güler, for their support and understanding in this process. I also want to thank Prof. Dr. Hüsnü Yenigün for sparing their time to join my thesis defence.

For encouraging me to decide for myself, and being there for me when I need it, I want to thank my family starting with my mother Emine Çinar, my father Yavuz Çinar, and my brother and best friend Alparslan Çinar. Along with my family, I am grateful for my friends Ekin Savaş, Furkan Akkurt, Tarık Deniz, Alper Serinkaya, İlker Aşık, Enes Baktır, Emre Yıldız, Mert Eliçelik, Turgay Çetinkaya, Ozan Savaş who have been like a family to me as well, on putting their faith in me. I always feel like I am the luckiest man on earth to have these people in my life.

And lastly, I want to thank my first teacher Suzan Zengin, who always believed in me and wanted me to be the best in anything I work on and guided me to be where I am today.

Understanding the Strategy Selection of Primary School Students in a Block-Based Programming Environment

Hasan Ertuğrul Çinar

Computer Science, Master's Thesis, 2024

Thesis Supervisor: Yücel SAYGIN

Abstract

Block-based programming is one of the most used methods for programming education for kids. Its simplistic nature makes it suitable for teaching students the fundamentals of coding along with computers working principles. With different platforms and game setups, students are generally free to solve problems by their strategies. To understand the student's approaches to problems, a dataset containing the event sequence of students in a block-based programming environment is used. Furthermore, an algorithm for evaluating the student's understanding of the given education is an important task that can facilitate programming education using these technologies by providing feedback to both students and teachers. By comparing students' code and action sequences for solving problems with expert approaches based on the subject shown in class, it has been examined to see whether students have implemented the given methods or chose to play their way. The carried out research shows that using end-code comparison is a promising method that can help to determine the strategy selection of students when used with clustering techniques. The experiments with the developed method showed that the students who chose to go write code in the best way they knew by not implementing the newly shown concepts performed better than the students who tried to use them. Along with the coding strategies of students, it is also crucial to evaluate the performances of male and female students to determine if there exists any significant difference to provide a system that focuses on students' strengths and weaknesses to enhance their learning experience. The statistical analysis has shown a difference between male and female students where males have a higher mean score than females. The effects of game setup are also taken into consideration in this thesis by conducting experiments based on a competitive game mode to analyse the effects of competitiveness. The results demonstrate that competitive game setup does not change the effect of strategy selection on performance. By creating a method for code similarity in block-based environments and analyzing the performance of students from various perspectives, we aim to propose a robust method for these evaluations.

Keywords: block-based programming, cluster analysis, coding education, strategy selection

Öğrencilerin Blok Temelli Programlama Ortamında Strateji Tercihlerinin Algılanması

Hasan Ertuğrul Çinar

Bilgisayar Mühendisliği, Yüksek Lisans Tezi, 2024

Tez danışmanı: Yücel SAYGIN

Anahtar Kelimeler: blok temelli programlama, küme analizi, kodlama eğitimi, strateji seçimi

Özet

Blok tabanlı programlama, çocuklar için programlama eğitiminin en çok kullanılan yöntemlerinden biridir. Basit yapısı, öğrencilere kodlamanın temellerini ve bilgisayarların çalışma prensiplerini öğretmek için uygundur. Farklı platformlar ve oyun düzenlemeleri ile öğrenciler genellikle kendi stratejileriyle problemleri çözmekte özgürdür. Öğrencilerin problemlere yaklaşımlarını anlamak için, blok tabanlı bir programlama ortamında öğrencilerin olay dizisini içeren bir veri seti kullanılır. Ayrıca, öğrencilere verilen eğitimi anlamalarını değerlendiren bir algoritma, öğrencilere ve öğretmenlere geri bildirim sağlayarak bu teknolojileri kullanarak programlama eğitimini kolaylaştırabilecek önemli bir görevdir. Öğrencilerin sınıfta gösterilen konuya dayalı uzman yaklaşımlarıyla problemleri çözme kodları ve eylem dizileri karşılaştırılarak, öğrencilerin verilen yöntemleri uygulayıp uygulamadıkları veya kendi yollarını seçip seçmedikleri incelenmiştir. Yapılan araştırma, son kod karşılaştırmasının, kümeleme teknikleriyle kullanıldığında öğrencilerin strateji seçimlerini belirlemede yardımcı olabilecek umut verici bir yöntem olduğunu göstermektedir. Geliştirilen yöntemle yapılan deneyler, yeni gösterilen kavramları kullanmaya çalışan öğrencilere göre, bildikleri en iyi şekilde kod yazmayı seçen öğrencilerin daha iyi performans gösterdiğini ortaya koymuştur. Öğrencilerin kodlama stratejileri ile birlikte, erkek ve kız öğrencilerin performanslarını değerlendirmek, öğrenme deneyimlerini geliştirmek için öğrencilerin güçlü ve zayıf yönlerine odaklanan bir sistem sağlamak amacıyla herhangi bir önemli farkın olup olmadığını belirlemek açısından da önemlidir. İstatistiksel analiz, erkek öğrencilerin ortalama puanının kız öğrencilere göre daha yüksek olduğunu göstermiştir. Oyun düzenlemesinin etkileri de bu tezde rekabetçi bir oyun modu temel alınarak yapılan deneylerle değerlendirilmiştir. Sonuçlar, rekabetçi oyun düzenlemesinin performans üzerindeki strateji seçiminin etkisini değiştirmediğini göstermektedir. Blok tabanlı ortamlarda kod benzerliği için bir yöntem oluşturarak ve öğrencilerin performanslarını çeşitli açılardan analiz ederek, bu değerlendirmeler için sağlam bir yöntem önermeyi amaçlıyoruz.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis aims to develop a method for analyzing students' log data in a block-based programming environment to differentiate students' approaches to completing game levels. The research also aims to analyze the data to discover the underlying factors that determine success. The motivation behind the research will be explained in 1.1. The overview regarding how the research is shaped for this subject and what are the key findings is given in 1.2. The related works about methods to analyze block programming environment data are examined in 2. The methodology for the developed methods and the analysis done are in sections 3.2 and 4. A brief conclusion section for summary and future work is discussed in 5.

1.1 Motivation

Block-based programming has become an important tool for programming education, along with its successful demonstrations such as Scratch and Blockly. It is a useful method for teaching kids and young students how to code by utilizing visual blocks representing code pieces to generate basic functions. By combining the blocks which are some actions in the environment, students create a series of actions, which does not have a strict syntax to follow. This way, student engagement is increased and the teaching process is facilitated by gamification. It also simplifies the process by getting rid of syntax requirements and encapsulating multiple actions that would normally be taken into one block, which makes the learning process less intimidating for non-coders. Along with the simplification of code itself, the interfaces are also prepared carefully for students, especially for kids, to easily use to create their environments for fun or complete generated levels that would lead them to learn different concepts regarding coding through them.

The employment of block-based programming tools has increased in the past years, with its benefits considered in both coding education and the development of critical thinking skills for problem-solving. Scratch, developed by MIT Media Lab, is arguably the most successful among all the block-based programming tools by having over 130 million registered users. Most of the users are between the ages of 9 to 18, which also shows its popularity among young people. It has been found as an effective learning method for students to enhance their critical thinking and problem-solving skills in [1] for students at age 12. Therefore, Scratch is a beneficiary tool for student's programming education. However, it is only one tool that emphasizes block programming methods. Each tool may have its approach to facilitating the process.

When block-based programming is examined by researchers, they generally try to understand the effects of these tools on the student's success when compared to other students who haven't used it, such as [1]. Some research also considers the motivation of students along with logical thinking and coding performances[2]. However, the importance of analyzing the students' methods when approaching solving problems is overshadowed. It is a crucial task to distinguish different strategies for further expanding the benefits by tailoring the environments to help them in their learning experience. If a student seems to struggle with a given concept, by not implementing it even if it is needed, it might be an indicator of not understanding it in the first place. Therefore, being able to identify the needs becomes a crucial task.

The main motive for using these tools for education is to support students' learning process by providing extra features to deliver crucial skills regarding programming and critical thinking that would be difficult to teach with traditional methods. However, it is also important to note that these tools are not miraculous devices that enhance students' understanding of these concepts directly. There must be a feedback cycle while a student is constantly trying to learn and improve their skills.

Without the proper process of tracking and supporting based on the needs, the effectiveness of these tools might be lower than they are intended to be. Thus, an automatized tracking and support mechanism is a lacking feature for programming education with block-based coding environments.

When it comes to analysing the students' choices in a block-based programming environment, there also exists some research that tries to understand the effective features to predict the students' success. Generally, they emphasize the number of various blocks that are used to analyze along with background information of students.[3] uses the number of blocks used, the number of block sequences and the count of system interactions for clustering student actions. Even though it can be a viable solution for distinguishing different solutions, it may not be enough to understand the written code but rather work with minimum requirements.

The need for correctly understanding the difference between two code pieces that are written in the same environment for categorizing solutions is the key aspect of this research. In this thesis, our main goal is to solve this problem. The proposed methods are explained in detail in the next sections.

1.2 Overview of Methodology and Contributions

This thesis aims to develop a method for understanding these various methods that are implemented by students to overcome the obstacles they encounter during a gamified block-based coding platform. The main approach consists of trying the effectiveness of two methods, where the first one is an action sequence-based similarity calculation, while the second one is an end-code-based edit distance method. The research mainly focuses on individual gameplay to understand different methods and their effectiveness on success, which is collecting as many points as they can to earn a reward. The research questions are formed in a way to examine these features in detail along with the importance of gender and game setup. Statistical tests are applied for hypothesis testing to ensure that the findings have statistical significance.

Main take aways can be listed as:

1. Lower event sequence distances do not correlate with higher scores in the game environment, however, the method is not as useful as end-code-based distance calculation.

2. Lower-end code distances do not correlate with higher scores in the game environment, however, there is a positive correlation between higher end-code distances and higher scores. The more direct approaches implemented receives higher scores.

3. Male students are more successful than female students in average, however, gender does not have any statistical significance when it is used for binary classification regarding to understanding whether a students belongs to a cluster of strategies or not

4. Playing the game in competitive mode does not change the effectiveness of strategies observed in individual setup, where the more direct approaches obtain more points.

The section 3.2 explains how the clustering methods are implemented along with the analysis methods we have used. After that, the effectiveness of these methods and results of the tests are explained in 4.

Chapter 2

Related Work

There is much research on block-based programming environments that focuses on various subjects such as the effectiveness of the system, extraction of knowledge states and clustering of students based on their action logs. But in general, all this research aims to develop an architecture that can facilitate the learning process for students by creating a positive feedback loop that can lead them in the right direction. We have examined similar works before starting to develop our model to see state-of-art techniques that are implemented by previous research.

2.1 Effectiveness of Block-Based Programming

As a start, there exists some research that examines the effectiveness of the block-based programming environment. Some research claims that block-based programming tools are useful at teaching critical thinking concepts to students, but they are not contributing much to students' coding skills[2]. However, some other research indicates that along with logical and critical thinking skills, these tools can also help students to get familiar with coding concepts which leads to an enhancement in programming skills including conditional statements and copying programming[4]. In another research that compares two groups, where one of the group utilizes block-based programming and the other one uses text-based coding, methods, it has been seen that the block-based programming users have achieved a higher computational thinking score than the other group while having a higher interest rates[5].

These environments such as Scratch and Blockly, also have their advantages of teaching by fostering the engagement of students and providing different options to program. Research comparing the effects of diverse environments, one that works with female students claims that Blockly increased students' interest in programming tasks more than Scratch [6]. However, Scratch is considered a more freedom centered and beginner friendly application that is useful for teaching basic programming concepts[7]. At the same time, Blockly is a better choice for diving into more complex ideas and transitioning into regular text-based coding [7]. Some other comparisons between Alice and Scratch stated that Scratch is far more effective at increasing students' understanding of reflective thinking[4], even though both platforms successfully facilitate the learning process.

2.2 Methods to Extract User Pattern Data

With the effectiveness of the block-based environment being recognized, there have been many methods proposed to analyze student actions in these environments. Some methods utilize the numeric features of user actions that are stored for each user separately to understand their action patterns. One of the proposed methods in this fashion [3], incorporates the number of used blocks per block type, number of actions and student information to apply clustering. This method seems to lack the amount of information to properly understand whether the used blocks serve any purpose, or do they have any meaning to be included in the code. Some other methods [8] that try to classify the different patterns to create a feedback mechanism propose to categorize patterns into high-performing (HP) and low-performing (LP) by mining the frequent patterns of high-performing and low-performing students in the given environment. This method aims to extract patterns to predict the likelihood of a code achieving a higher programming performance score, which is found successful at predictions. However, it might require a higher sample of students and their performance classifications to be done before extracting the frequent patterns to obtain a more robust solution that can be generalized for other block-based environments.

Another research about how the action logs can be used as metrics in strategy identification [9]. They have assigned students to expert-given skill levels by examining their coding processes. This research indicates that the time, number of block deletions and adjustments that change the number of block types are metrics that are found to be correlated with the assigned skill levels. However the sample size is considerably low to analyze the patterns that are found, therefore it would be good to be expanded by a method to automatically assign students to skill levels without the need of experts to follow each student's processes to be able to increase the number of students for the sample size.

2.3 Knowledge State Identification

Knowledge states are another hot topic in educational fields, where the main objective is to examine students' knowledge levels based on given concepts. It is also a highly researched topic for online education since understanding whether students grasp a concept totally would be crucial to providing feedback on their requirements. Therefore, it is important to generate methods to automatically assess students' knowledge states in block-based programming environments too. Some research suggests manual frameworks to analyze students' proficiency and understanding of programming concepts [10]. The proposed framework collects data of students to determine their knowledge level on topics including sequencing, selection and iterations. The group knowledge states as novice, intermediate or advanced based on their skill level. The evaluations are carried out based on performance on challenge completion, code structures and their ability to explain it. Even though their framework provides a valuable baseline to develop an automated system, they completed assessments manually based on their criteria. This causes it to fall short of assessing large numbers of students in a given system. A need for an automated knowledge assessment system based on evaluation metrics would need to be developed as well.

Another research that focuses on knowledge states proposes a fully automatic system that acknowledges knowledge states[11]. In this system, they implement Chained Hidden Markov Models(CHMM) to create a probabilistic approach to mine hidden knowledge states of students based on a problem solving puzzle game played in a block-based coding environment. They have predefined knowledge states where each student's knowledge state is mapped into, where the defined states are Trial and Error, Systematic Testing, Implement Solution and Generalize Solution. Their approach is based on a probabilistic method that incorporates sequence analysis to mine action sequences. They use Multivariate Gaussian distributions based on the selected features to calculate the most likely outcome. By applying CHMM, they found out they students tend to loop in their current knowledge state rather than transitioning. This method is highly useful to extract the knowledge states and can be applied to different environments which contain a curriculum and needs tools to asses knowledge states to provide feedback based on students' current situation.

2.4 Strategy Selection

Similar to knowledge states, strategy selection is another important aspect in block-based programming, where understanding the strategy choices of students could be useful in estimating the needs of a student to comprehend the newly encountered subjects. Strategies demonstrate the students' comfort in applying a concept in the game, which can help to measure students' proficiency in a given topic. Thus, it is another hot topic for research regarding block-based programming. The strategy selection is generally important for feedback production for students. In [12], a hybrid approach is implemented for evaluating students' completion of subgoals generated by experts and a data-driven model that uses past data, is developed for understanding which tasks are completed by students to provide feedback on their progress to help them move forward. The important part is that their approach aims to provide immediate feedback, which is a better option for accelerating the learning process. However, they do not distinguish different student strategies automatically, which would normally help their system to provide dynamic feedback based on the student's method for solving the problem that is represented.

Some other research solely focuses on strategy selections. Research [13], suggests a Bayesian Hierarchical Model to predict the performance of students in a given programming test after they interacted with the given environment, PRIME, in their case. Their experiment was conducted on university students and had 99 participants. They developed a method to see the similarity between an expert's solution to levels, by comparing the number of each block in the expert solution and students' solutions, and by taking the difference between them, they calculated the difference. Their results have shown a promising mean square error (MSE) score when the model used negative binomial distributions compared to other distributions such as normal and Poisson. The proposed method is a good example of understanding whether students have implemented the expected solution or not, but not checking the order of used blocks, but only considering the number of required blocks, it misses whether the main problem is about using the right amount of each statement, or the order of the statements that are used. However, this method is a good demonstration of understanding whether the student has the knowledge of coding to solve upcoming challenges or if they need help.

Key Takeaways

By looking at these resources, we have seen that block-based programming is an important tool for both programming education and the development of critical thinking. Moreover, it is important to work on students' data in these systems to evaluate coding performances to discover problems that students face, whether not understanding concepts totally or failing to implement them in a reasonable time, to facilitate the learning process. To do so, clustering students based on their actions or applying techniques to mine the knowledge states of students are available options. By looking at these, we have decided to come up with a metric that would help us to determine the strategy selection of students, and then analyze the performance of found strategies based on clustering them together with similar students' actions.

Chapter 3

Problem Definition and Methodology

This chapter aims to provide a definition for the main problem that laid the foundation for this thesis in Section 3.1. Methodology and the dataset that is used in this thesis will be discussed in Section 3.2.

3.1 Problem Definition and Research Questions

The problem definition for this thesis will be given in Section 3.1.1. After that, the research questions that is aimed to be answered by this research will be listed in Section 3.1.2

3.1.1 Problem Definition

The main goal of this thesis is to understand the different strategies that are deployed by primary school students and their effects on overcoming basic programming problems that they encounter in a block-based programming environment. The study also focuses on shedding light on the various other parameters that might affect the selection of a strategy selection such as gender and game setup (individual/-competitive). The research process starts with working on basic concepts such as understanding the correlation and causation between different events and progress towards painting a picture that shows how to differentiate coding strategies, then the students' preferred actions based on the setup, gender and the outcome of combining all these different factors on the task they have been assigned to.

3.1.2 Research Questions

This section aims to provide the research questions that shaped the research process. The project mainly focuses on examining student choices based on related parameters and the results they obtain. In this context, two concepts are focused on, the understanding and usage of coding concepts to define the students' strategies, and analyzing the effect of strategies, gender and game setup on the selection on the outcomes.

The two parts of the research, calculating students understanding and usage of coding concepts and analysis of obtained data brought different research questions. The strategy extraction part revolves around the action stream and resulting code blocks, the analysis tries to find answers for possible correlations.

Research questions:

- **RQ1:** Do lower event sequence distances with an optimally written code imply higher scores for the given tasks?
- **RQ2:** Do lower end-code distances with an optimally written code imply higher scores for the given tasks?
- **RQ3:** Is there a relation between gender and the selection of a strategy?
- **RQ4:** Is there a relation with the game setup and the selection of a strategy?

3.2 Methodology

In this methodology section, the methods used to extract information from the raw dataset and the analysis part of the project is provided. In first section 3.2.1 the game environment the data is obtained is explained. Section 3.2.2 covers the structure of the data, as well as the number of instances that are used for the analysis. The data cleaning for getting rid of unnecessary parts is detailed under 3.2.3.

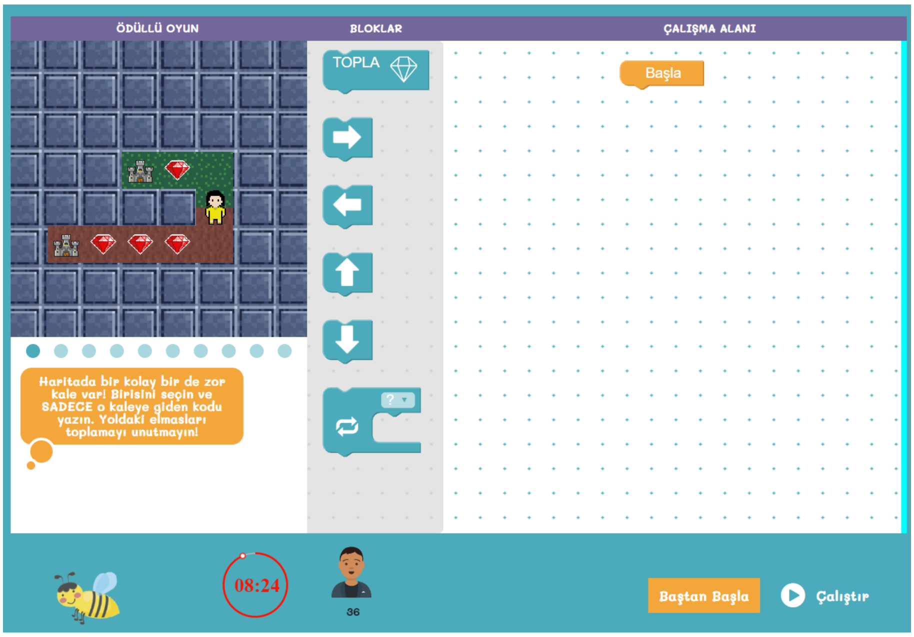
.

3.2.1 Game Environment

The data used in this thesis come from the project “Improving Gender and Immigrant Outcomes through the Social Malleability of Attitudes: Randomized Interventions on Peer Interactions in an Educational Setting”, funded by the ERC Consolidator Grant (Grant number:866479), led by Dr. Seda Ertac (Koc University). Specifically, we use the game-play data from a single game in the educational coding platform developed for this project. It contains five stages where the first and the last stages are quizzes related to basic questions regarding the outcome of a code block or finding the correct code block for a missing part. It aims to calculate the effects of provided lessons regarding coding concepts and students’ understanding of these concepts after completing the coding stages.

There are 10 practice levels in stage 2 and 10 competition levels in stage 3 for coding challenges. The more important sections are the coding sections, where the second stage is a warm-up for students to get familiar with the setup of the game and does not provide any points for students, which has been told to students before they start the game. However, the third stage determines whether the student wins a prize or not. It contains two different roads to two different castles, where one of the roads is shorter, but yields less diamonds, while the other one is the complete opposite.

There exist 6 different code blocks a student can use to traverse from the starting position to the castle to which they have decided to go. In general, students need to complete levels as fast as possible since there is a time limit to complete the whole stage. Moreover, they need to collect diamonds to win the prize. Normally, to write the code in the shortest amount of time, utilizing the loops, which is called a repeat button in the game, helps to eliminate unnecessary block placements for movement. Since levels are designed specifically to grant an advantage to repeat block usage with repetitive road patterns. Students also need to manually collect diamonds while the avatar is on them, otherwise even if they have passed over the diamond, they do not get any points.



Haritada bir kolay bir de zor kale var! Birisini seçin ve SADECE o kaleye giden kodu yazın. Yoldaki elmasları toplamayı unutmayın!

Figure 3.1: An example image of stage 3

3.2.2 Dataset

The raw dataset used for this research consists of event logs that are recorded for each play-through of students. Each student in the dataset plays at most one game and one mod, which are individual, cooperative and competitive. Therefore if a student exists in an individual group, they cannot be found in other game mod data. The experiment only considers individual and competitive game mods and does not include any data points from cooperative games. Even though there exist different game mods, the log data is recorded almost the same way for each student's actions. For individual games, the data contains student information, game setup and actions that are taken by the student in each stage of the game as a JSON document. It can be illustrated as:

Table 3.1: Dataset Fields for Student Action Logs

Data Section	**Contains**
home	student_id grade school game_mode
quiz_stage_events	question_no answer
code_stage_events	block_type code event_type start_time

- **student_id:** Student's unique id in the given classroom and school
- **grade:** Student's classroom information
- **school:** Name of the school that student studies
- **game_mode:** The type of game student plays (individual, competitive, co-operative)
- **answer:** Student's selected answer for given quiz question
- **block_type:** For a given log, gives the type of block affected
- **code:** Resulting code after the change
- **event_type:** Type of the action that has been done, such as move or delete
- **start_time:** Logs time record

While starting the game, students' info, which is a school name, grade and ID, is entered into the system. Then students proceed to complete levels one by one. For each stage, students' actions are recorded and presented in the dataset. For the quiz stages, it contains the answer given by the students along with start and finish times. In the code stages, each level has its action sequence where the affected block, event type and the resulting code chunk are given. The end code after any action is in XML format which can be parsed as an object.

3.2.3 Data Cleaning

The dataset used for clustering operations has been created after taking mandatory data cleaning processes. These processes include clearing out duplicates, empty records, unrelated game types and inserted values.

Inside the dataset, some session records contain the same students' records, which normally cannot happen due to the format of the experiment that collected the data in the first place. Each student is supposed to play the game once. Therefore, duplicate values are erased from the resulting dataset. The same is done for empty records, which are created mistakenly during the data collection stage. This situation may be caused by unnecessary logins to the system for test reasons or students' wrong selection of their user ID.

The other unnecessary part of the data is the records regarding to cooperative game mode. Since cooperative game plays are not included in this research's scope, the data coming from this game mode are cleaned entirely from the dataset used for the clustering and analysis. For the other two game modes, their data has been separated for analysis separately to understand the changes between competitive and individual playthroughs.

3.2.4 Data Pre-Processing

To start analyzing the data, it had to be preprocessed to obtain additional information regarding students' approaches to problems. Even though the dataset mainly consists of students' actions, it does not have a structured way of presenting the action set for analysis. Moreover, the dataset does not contain some sort of performance metric directly, therefore, the metrics need to be set and added to the dataset as they correspond to students' actions.

First of all, to examine students' action flow, the action logs are restructured to contain only the block type and action type as a whole and create a queue, where every action is in the order that they have taken place. This is a mandatory operation for conducting distance-based comparisons with different coding styles. The methods applied to the generated structure will be talked about in detail in upcoming sections.

After generating the sequence data, since there exist two different ways students can choose, and this affects the way they approach the problem, data extraction from the formed sequence data is applied to understand which road is chosen by the student. This is both mandatory and useful for analysis, since without knowing the chosen road, it would not make sense to compare students with a given sequence which does not match the same objective. Moreover, the chosen roads represent students' tendency to select the easy or the hard one to solve. Therefore, it also contributes directly to the analysis of strategy selection in terms of coding.

Once the road and sequence data are obtained, several objectives collected through the road are also required for having a performance metric. Even if a student cannot finish a level, we can consider how far they have reached based on their performance metric along with the two other metrics we extracted from the raw data. To accomplish this, a simulation algorithm is created for reproducing the results of written codes of students for each level, and find out how far they have come.

3.2.5 Dynamic Time Warping Model For Event Sequences

In this thesis, we used several methods to calculate their effectiveness in understanding student strategies. The first model we came up with is the Dynamic Time Warping Model (DTW), for aligning two event sequences to see the difference between the building process of the end code. DTW is a method that is mostly used for comparing two different sequences and generating a distance between those two based on the events that are present in those sequences. It is a powerful technique which does not need sequences to have the same length. It stretches or squeezes the time axis of the sequence data to find the optimal alignment.

DTW is especially useful in fields where a time series or sequence data is examined. The main use cases in different fields for this method include speech recognition, bioinformatics for gene analysis and pattern recognition. The bioinformatics example is similar to our case since they represent gene sequences as blocks to compare two sequences for the differences and their effects. Therefore, for comparing different data with a list of items representing events or objects in a given time, DTW is a highly used method for comparisons.

Table 3.2: Event Sequence Name Convention

Blocks	**Actions**	**Example**
right	create	create-right
left	move	move-down
up	delete	delete-collect
down		
collect		
repeat		
repeat	click	click-repeat
	change	change-repeat-2

For the our sequences, the events are stored as a single string for comparisons to see if they are same. For this purpose, a naming convention for actions is generated following this rule. Each event is directly connected to its block to represent the action taken as a string. As an example, the move-right represents moving the ‘right‘ block somewhere else than its position. The convention provided in the 3.2 can be followed for all blocks. Some blocks cannot take other blocks’ events, therefore, blocks like repeat has their special actions as well, such as ‘change‘ which sets its number of iterations. Before starting to compare sequences, the data is prepared in this way.

Our method includes comparing the students’ action sequence data with the expert’s action sequence data for each level to observe the difference between them, and how this difference affects the student’s way of solving problems. However, the cleaned data requires extra work to be processed by the DTW model that we created. Since the event sequence of the students does not contain numerical values representing the current move they chose for the given event, the data needs to be turned into a list of vectors, where each vector corresponds to a certain event. Once the events are mapped into vectors using One Hot Encoding, the DTW can be applied to the new sequence that is generated.

The output of DTW is stored for each student's distance metric for each level to be used in clustering, which is explained in 3.2.7. In general, students' event sequence distance is high, which is probably due to their need to adjust their code according to the mistakes they have made, while the expert finishes levels with ease when compared to the students. Even though this is problematic in terms of categorizing the strategies based on their approach and execution of code writing process, it still helps us to see that most students tried many different actions before reaching the solution, which shows the general flow of the learning process.

3.2.6 End Code Comparison Model

The other model that is used for finding out the similarity between the optimal code and the students' solutions is the end code comparison model. This model aims to match two XML code chunks with each other by calculating the amount of actions needed to convert one into another. To calculate the cost of this operation, the model uses edit distance, which is also called Levenshtein distance. Levenshtein distance is a useful metric in various fields such as spell-checking algorithms, plagiarism detection and DNA sequence analysis. The common requirement of these fields is the need to calculate the required changes for converting A into B so that it can be examined if they are similar based on the amount of work that needs to be done. When the requirement of understanding whether a student's code is similar to what is expected, can help to distinguish different strategies that are applied by students.

Normally, the Levenshtein distance method is used for converting strings into each other, by counting the number of change, add and delete operations to find the distance between two different words. In our case, since the code blocks are predefined due to the number of actions being limited, they can act like a letter in a string as well. Some of the blocks and their name convention can be seen in table 3.3.

Block	Name Convention
Up	U
Down	D
Right	RI
Left	L
Repeat1-2-3-4	RE1-2-3-4
Start	S
Collect	C
Next	N

Table 3.3: Code Block Name Convention

To use coding blocks instead of strings, we need to turn the XML tree that keeps code into a flat array of blocks. While creating this array, the XML tree needs to be parsed just like an Abstract Syntax Tree to keep code integrity, since we have nested blocks inside repeat operations. To ensure that the Next(N) blocks are used to identify if something is inside a block or after the block. By separating the code using this method, it can be ensured that the code sequence can be compared in the right way.

Once the flattened end code chunk is ready, the comparison operation takes place, where for each action, the algorithm can choose between changing, deleting or adding a code block in student code. While deciding which operation to choose, it uses a dynamic programming approach to keep the least distance in a matrix for each action, by taking the minimal distance in each step, similar to greedy algorithms. Therefore, for any difference between two codes, the least amount of change is selected to turn one into the other. By applying this approach, it calculates the edit distance between two code chunks.

3.2.7 Clustering

Clustering is an unsupervised machine learning method which groups similar data points into the same groups to assign them labels to distinguish them from each other. It is a preferred method in data analysis when the groups of data are not predetermined and need to be labelled based on their characteristics. It can be used for market segmentation, user profiling in social networks and gene and protein analysis in bioinformatics. The use case of the clustering also affects the chosen clustering methods. For instance, for a small dataset where the number of clusters can be defined before the clustering operations, a basic clustering method such as K-means can be used, while for hierarchical structures hierarchical clustering can be chosen. For our case, since we will be clustering students based on their code similarity and number of actions, applying a soft clustering method such as GMM and DBSCAN, a density based clustering algorithm, is applied to see if there would be a difference between clustering algorithms.

Gaussian Mixture Models

Gaussian Mixture Models(GMM) is a soft clustering method that takes advantage of probabilities while assigning data points to give clusters [14]. Since it does not directly assign any point to a cluster but takes probabilities into consideration, it is a soft clustering method. It is also a good choice when the clusters are expected to overlap each other. Due to its probabilistic nature, GMM handles it better than other hard clustering models such as DBSCAN, which expects points to fit into its density criteria, otherwise labels them as outliers, for datasets where it cannot be guaranteed for a dataset that displays the characteristics properly for each point.

The working principle of GMM is to generate a probability density function as:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k) \tag{3.1}$$

The K is the number of clusters,π_k are the mixing coefficients and $\mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)$ is the distributions with mean and covariance. The model then tries to maximize the expectations (EM Step) by computing the probabilities for each data point and adjusting its parameters accordingly to maximize the probabilities for data. Details about how the GMM model is used for clustering will be explained in the next section.

Clustering Models in Game Data

Before starting clustering using clustering models, we separated the sequence and end-code distance datasets to examine them. Afterwards, We applied normalization and standardization for all quantitative features in our dataset. Other than that, categorical variables such as school and gender information are excluded from the datasets as well. Once the data is ready to be served to the clustering model, we need to understand the optimal number of clusters. To find out the optimal clusters, we have to apply different tests and consider their results together. The applied tests are BIC, AIC and Silhouette Scores, and both BIC and AIC scores should be low, while the Silhouette score needs to be high for the given number of clusters to consider it as the perfect fit.

For trying different methods for clustering, we also implemented the DBSCAN clustering method, which is a density-based algorithm. It is expected to provide clusters that are more strictly separated since it considers dense areas as clusters and considers points which do not fall into these zones as outliers. We will be examining the performance and pros and cons of both methods in this way.

DTW Clustering Process

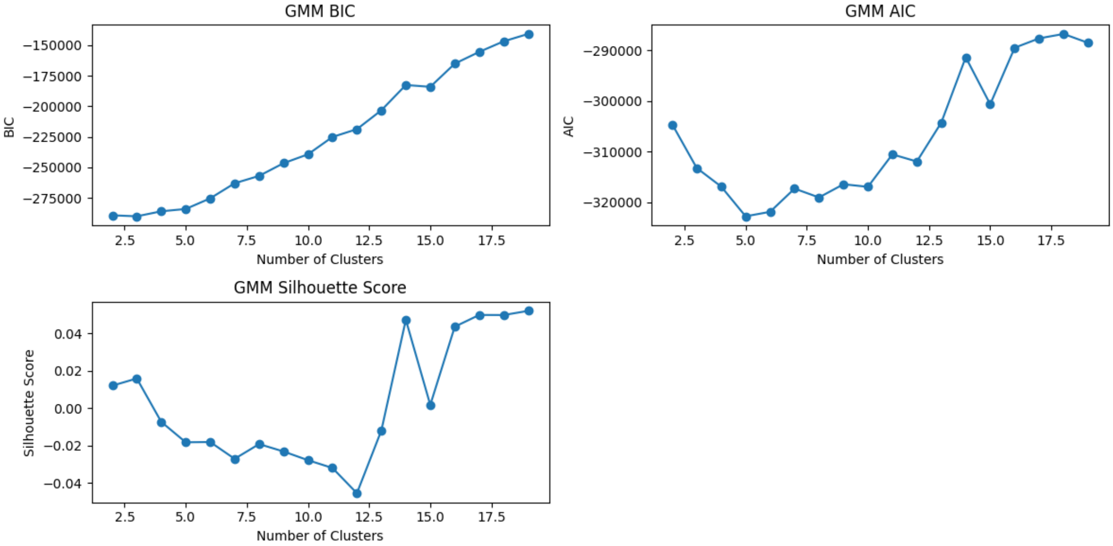


Figure 3.2: DTW Clustering Test Results

For DTW approach, the test results are as displayed on 3.2. BIC scores generally increase along the number of clusters, which means that a lesser cluster number is desired by BIC. For AIC scores, it diminishes until 5 clusters, and afterwards, it starts to increase, which means that the number of clusters should be between 1 to 5 according to BIC and AIC scores, and preferably 5, since until 5 it does not change much for BIC but for AIC it undergoes a drastic change when compared to lesser number of clusters. However, Silhouette's score gives us a contrary situation where it needs to be less than 3 or more than 13 since we expect to have a positive Silhouette score for correctly assigned data points and it needs to be as high as possible for more well-defined clusters. Therefore, when all of these are taken into consideration, 3 is the selected amount of clusters for DTW approach to have a clean clustering process.

End Code Clustering Process

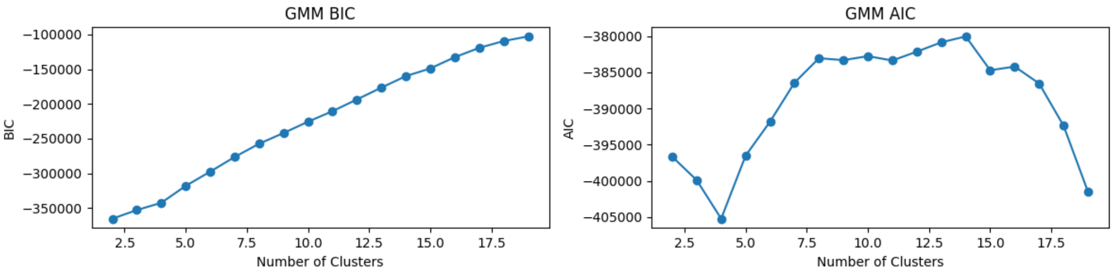


Figure 3.3: End Code Clustering Test Results

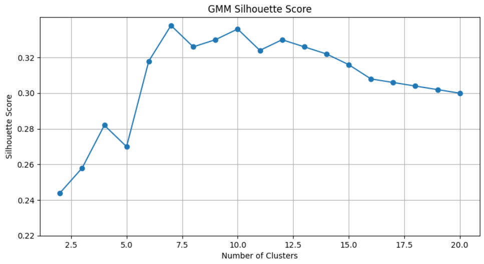


Figure 3.4: End Code Clustering Silhouette Results

The encode clustering process also requires the selection of an optimal cluster number since it uses the same clustering method. In this case, the results given in 3.3 demonstrate the values for the tests. Firstly, the BIC score increases as the cluster number increments, therefore, a lower value is preferred for BIC. On the other hand, the AIC test demonstrates that after the point of 4 clusters, the AIC score starts to boost, making it less selectable for our clusters. By looking at these two tests, 4 is the optimal number. When the Silhouette score is taken into consideration, even though all values are positive in this case, the higher the Silhouette score, the better-defined clusters are created. Hence, we select 4 as the optimal cluster number since values after 4 do not contribute to both AIC and BIC and 4 has the highest local value for the Silhouette score as well.

DBSCAN Clustering Method

DBSCAN is a density based clustering method that forms clusters based on the zones that contain dense data points, in other words, where the data points are closer to each other, it accepts those zones as clusters. In this method, there is a threshold, ϵ, for the distance between data points to not exceed, in order to accept them in the same zone, or cluster. Due to being a hard clustering method, it is expected to separate clusters more clearly. However, DBSCAN can struggle with high amount of data and high dimensionality.

The DBSCAN method require its hyperparameters, ϵ and min-sample to be tuned for providing useful clusters and not just noise. To analyze this, both elbow method and silhouette scores are taken into consideration.

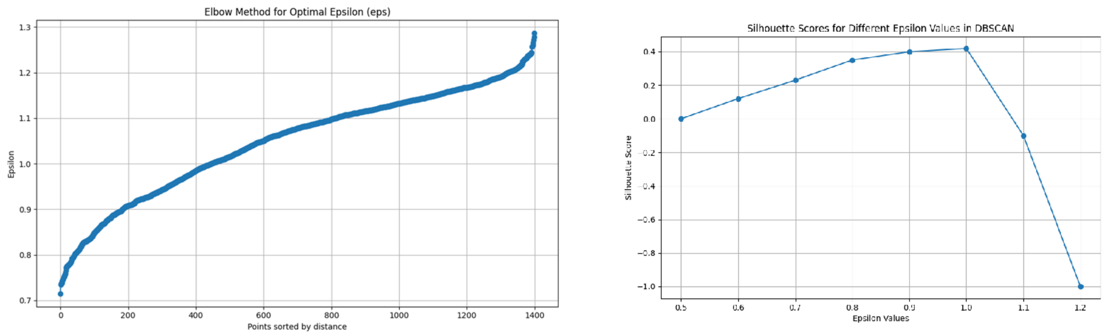


Figure 3.5: End-Code Distance Clustering with DBSCAN

Normally, by considering the elbow method, we would pick the epsilon at the point where a rapid increase starts, which is around 1.2 in 3.5, however, when we apply the silhouette score test, we can see that number of clusters drops down to one, once the epsilon past the limit of 1.05, which does not help us to distinguish different points. In order to see the clusters in this case, we selected 0.95 as the epsilon value to generate clusters. Similar to GMM values for the silhouette test, the values are a bit low, which indicates that the clusters are close to each other and there are data points that overlap.

Statistical Analysis Methods

In this thesis, we conducted statistical analysis to apply hypothesis tests for validating our findings regarding the research questions. In this context, some statistical tests are utilized for correlation and significance of related variables. For all statistical analyses, the confidence level is set to 95% for getting a robust outcome.

Spearman Correlation Test

Spearman Correlation is a test that computes the correlation between two variables while also considering the ranking between different variables. There exist many tests and methods that are available for different tests, such as Kruskal-Wallis and ANOVA. Some parametric tests, such as ANOVA expect a normal distribution while Spearman does not require those conditions but needs to have an adequate number of samples to work. Even though the Spearman test is mostly used with ordinal data for ranking, it can be used with continuous data to find monotonic relations between variables. When these are considered, using Spearman for our correlation and significance tests is a valid approach. For DTW and end-code methods, we decided to apply the Spearman test with the clustering operations data for understanding the relations between the distance features and the performance of students, where the results can be seen in 4.2 sections.

Spearman correlation ρ can be calculated as follows:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

where:

- d_i is the difference between the ranks of each pair of observations.
- n is the number of observations.

After obtaining the correlation value ρ, the t value for statistical analysis can be calculated using the formula:

$$t = \frac{\rho\sqrt{n-2}}{\sqrt{1-\rho^2}}$$

By using the t-value, the t-distributions table can be examined to find the corresponding p-value, which determines the statistical significance checking if it is less than the significance level. Having a p-value that is less than the significance level means that the null hypothesis can be rejected.

Logistic Regression

The other method we employed for gender significance while deciding on the membership of a student to a cluster is logistic regression. Logistic regression is a widely used method for binary classification which makes it a perfect candidate for analyzing the membership and gender relationship. Logistic regression generates a logistic function that works as an activation function known as a sigmoid. The function maps the given variables into a binary value for deciding the results. Since in our research, we want to understand if gender plays any role in being a member of a cluster, with the help of one hot encoding for our clusters we obtained by our two different distance metrics, DTW and End-code, logistic regression is a valid approach to determine the significance of gender values. In our research, we mapped males to 0 while mapping females to 1 as a categorical variable for these operations.

Welch's t-test

Additionally, Welch's t-test is used for evaluating the significance of the average performance scores for male and female students. Welch's t-test is a method that compares two groups' mean scores to find out if the differences between them are statistically significant. It is a robust version of the Student's t-test by not requiring equal variances between two groups. While applying a hypothesis test for the relation with gender performances, the t-test is chosen as the comparison method.

The Welch's t-test statistic (t) is calculated as:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Similar to the Spearman test's t-value, we can use the t-value to determine the corresponding p-value and check if it is less than the significance level.

3.2.8 Experiment Methodology

The experiments aim to answer the research questions along with providing general information hidden inside the raw dataset that is used in this research. The main focus of the thesis is on the distance measurement algorithms and the clusters that are formed based on the distances that are provided along with other parts of essential data. Therefore, first, we try to understand the basics of the dataset that is used, then move forward to answer the research questions provided in 3.1.2. To find the answers, the clustering analysis and correlation evaluations are done for both algorithms. After that, the relationship between gender and game setup is examined based on the success criteria.

Starting with descriptive analysis, we examined the main differences in the dataset. The overall success of all students, success comparison of students based on gender and schools, and lastly based on their quiz scores are calculated. To understand if there exists any statistically significant change between comparison groups, we applied Welch's t-test to compare the mean scores. Since our sample set is a subset of a large dataset, by taking central limit theorem[15], we can apply statistical tests since we can assume they approximate normal distributions. The confidence interval is set to 95%. Statistical tests such as t-tests help us to understand if the evidence we have is strong enough to reject the null hypothesis. By using these tests, we can carry out hypothesis tests to make sure the findings are strong enough to support our claim. For the t-tests, we separate our data into two groups to compare their means to see if we have evidence to reject the null hypothesis, since our confidence interval is 95%, our p values must be less than 0.05 to indicate a strong difference between the two groups.

Following the descriptive analysis, The main experiments for clustering with two different approaches examine the effectiveness of the distance calculation algorithm and its usefulness in determining the similarity between two codes. To understand, we applied statistical tests with Spearman correlation to see if we could see any relation between the distances obtained and the collected diamonds.

For the first method using DTW, we generated the sequence of actions data, then applied clustering along with the quiz scores, number of blocks, and chosen road counts. Then the clustering scores for BIC, AIC and Silhouette scores to for clusters that are informative, less prone to overfitting and compact when GMM is used as the clustering method. By considering these scores, we generated the clusters and analysed each cluster's characteristics. After cluster analysis, the Spearman correlation test is applied to each distance value of levels to see their relations with the total diamonds collected. Through this experiment, we aim to see the positive or negative correlation of each level's distance data and its usefulness in informing us about the total score of students. Also, each score's statistical significance is calculated and examined to see their importance for showing as evidence.

After completing the analysis of clusters based on DTW data, we moved on to the end code-based distance data to generate clusters. DTW distances data is replaced by the end code distances, by keeping other essential information to help determine student strategies. For the cluster numbers, the same procedures are applied for end code data as well. After completing cluster number calculations the clusters are generated, and cluster analysis is done. The correlation test for each level's distance is carried out to understand the effect of the distance data concerning the student's success. Moreover, statistical tests for correlations are applied similarly to the DTW distances. The resulting values are used for the hypothesis test. For the DBSCAN, we examined the optimal epsilon and moved forward to cluster analysis just like the GMM.

Once the clustering processes were completed and analysis was done, we started to examine the effect of gender on binary classification for each cluster, where we focused on its effectiveness while using logistic regression and whether it has any statistical significance on this prediction process. To do that, we calculated the importance of gender along with other features in the logistic function to determine if a student belongs to a cluster or not. The cluster labels are converted to binary values by applying one hot encoding. A logistic regression test is used here for hypothesis testing.

For our last research question, the data of competitive matches have been used clustering based on end code distances. The distances are treated similarly to the individual versions. The clustering process is also done by following the same steps to ensure there are no other differences in the process other than the data itself. Cluster analysis and significance tests are completed according to the process that has been done to individual game data.

These are all the methods and the flow of the operations that are performed through the course of the research process. All selected methods intend to answer selected research questions.

Chapter 4

Evaluation

In the evaluation section, the results obtained by the experiments conducted according to the methods explained in 3.2 are discussed. The experiments aim to answer the research questions provided in 3.1.2. The section includes sequence analysis-based experiments, 4.2.1, and end-code distance analysis-based experiments in 4.2.2 for their relation with the strategy selections and the strategies' effect on success in the game. Moreover, gender-based strategy selection and the game mode-based strategy selection experiments are also conducted to find out if there exists any relation.

4.1 Descriptive Analysis

Before analysing the outcomes of the clustering methods and applying hypothesis testing for our research questions based on those results, we first carried descriptive analysis of the dataset we obtained and expanded. The descriptive analysis is a beneficial tool for grasping the crucial aspects of the data that is used for the research. It helps to direct the research methods and brings out questions concerning the subject.

The descriptive analysis for this thesis includes quantitative data regarding our sample, gender-related features, school-based performance analysis, quiz and coding stage performance correlations. This analysis aims to facilitate to understanding of the player's general characteristics along with the game game environment.

General Performance Examination

There are 1324 students in total in the dataset we used for our project. The dataset contains two different variations for each game, and only the ’b’ variation is used for the tests due to the difference of games between variations. Some student data contains false information due to wrong recordings in game logs, where some have achieved scores that are not feasible to obtain. Due to this kind of problem, the student number is decreased to 1214 for the research. Students come from various schools across Istanbul, and their personal information is excluded from the dataset to preserve privacy. Some personal variables such as school information are mapped to codes for later analysis.

Count	**Mean (Comp. Lev.)**	**Std. Dev. (Comp. Lev.)**	**Mean (Diamond)**	**Std. Dev. (Diamond)**
1214	6.85	2.75	26.62	14.43

Table 4.1: Statistics of completed levels and diamonds. Mean values are reported with their corresponding standard deviations.

The student’s general performance is examined to understand the mean score to determine the success of students that belong to various clusters, and whether they are more successful on average than the average of the whole population in the sample. When the performance values based on level completion and collected diamond numbers are inspected, the result shows that the mean score of the students is 6.85 for level completion in 10 levels of stage 3, while their mean diamond score is 26.62 out of 45 diamonds can be collected if students have consistently chosen the hard path to complete levels.

Performance Based Gender Analysis

Gender-based statistics can help to distinguish the behavioural patterns regarding coding education. It is important to understand different aspects of the dataset to provide a comprehensive analysis. Therefore, differences between genders are looked into for both clusters and in general.

Gender	**Count**	**Mean (Comp. Lev.)**	**Std. Dev. (Comp. Lev.)**	**Mean (Diamond)**	**Std. Dev. (Diamond)**
Male	625	7.28	2.63	28.65	14.42
Female	589	6.41	2.80	24.47	14.13

Table 4.2: Statistics of completed levels and diamonds based on gender. Mean values are reported with their corresponding standard deviations.

For the whole dataset, we have conducted the research on 625 male and 589 female students' data. The performances of the students show that males performed better than females in this game as it is shown in 4.2, when the collected diamond scores are taken into consideration. The average of males is 28.65 diamonds and 7.28 accomplished levels, while female students have 24.47 diamonds on average and 6.41 completed levels. As it is seen, males have earned more diamonds around 17.08% while completing around 13.57% more levels in average.

Table 4.3: T-test Results for Gender Performances

Test	**Means**	**Result**
Male,Female	(28.65,24.47)	Reject Null Hypothesis

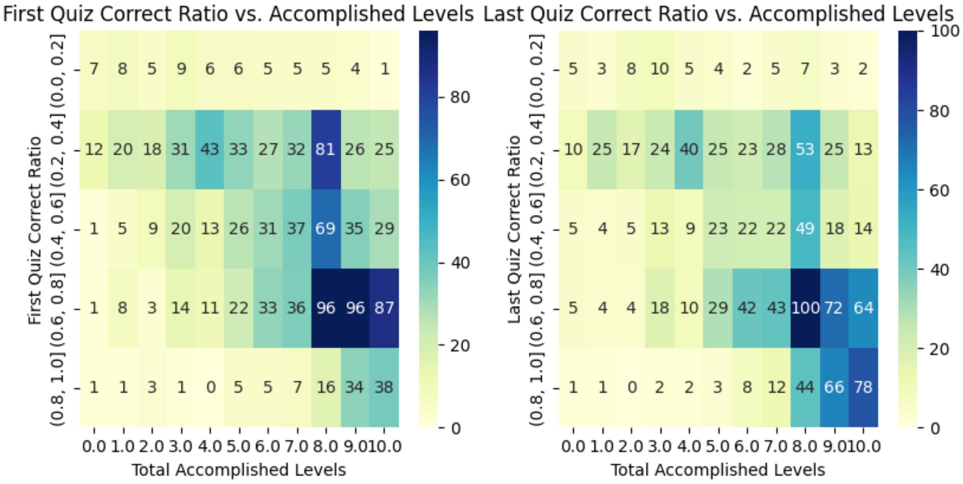
When Welch's t-test is applied to evaluate the significance of the change between two mean scores, it shows that there exists a significant change that is enough to reject the null hypothesis of not having any significant difference between the mean scores.

Quiz Analysis

There are two quiz sections prepared in the game environment. The quizzes aim to see the difference in students' ability to analyze a code piece and evaluate how it works, or what needs to be added in order to make it functional. To measure students understanding, quizzes are placed at the beginning and the end of a gaming session. There are 8 questions per quiz, and the first quiz's questions are relatively easier than the last quiz's questions. It is also important to analyze whether students have improved their quiz scores. Because it might indicate that the coding stages have helped students to comprehend what is taught about the programming fundamentals in that week.

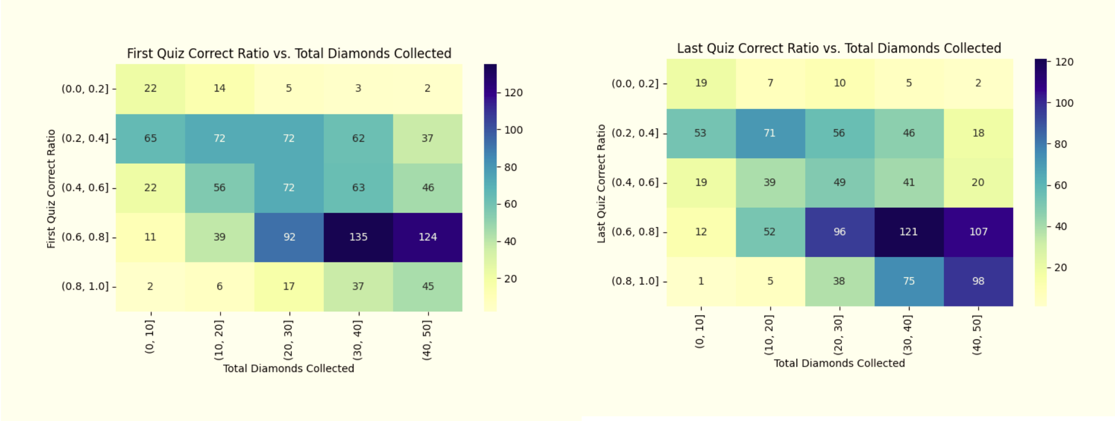
When we examine the quiz scores, we see that there is a positive correlative trend between the quiz scores and the obtained points in the competition stages. Students who are successful in first quiz, have completed more levels in general. Moreover, their total scores in stage 3 are also higher than the students who got a lower score on quiz questions. For the last quiz stage, students who scored high, received even higher scores on the last quiz, even though it was a bit more challenging than the first quiz section.

Figure 4.1: Relation of levels passed in stage 3 with the quiz scores



The heatmap represents the number of students in given ranges for accomplished levels and the obtained percentage of correct answers to the total number of questions. We can see that, there is a trend between quiz scores and the accomplished levels, since, as the overall correct percentage increases, the number of accomplished levels also increases as well. For the last quizzes, there are more students who got into the highest overall quiz score bin, which indicates the increase in the quiz performance after the coding stages. Moreover, it also demonstrates the same trend, where the quiz scores increase, the accomplished levels increase as well.

Figure 4.2: Relation of diamonds collected in stage 3 with the quiz scores



The 4.2 demonstrates that coding stage scores also have a trend similar to accomplished levels, where the code scores increase as the quiz scores increase. For the first quiz, there are 259 students who has scored more than 30 while scoring 5 to 6 correct answers in the first quiz stage. There are only 82 students who got more than 30 points while getting 7 to 8 correct quiz stages. The situation changes when the last quiz stage points are examined, where the number of students who got 0.8 or more overall quiz scores and more than 30 points in the coding stage is obtained. This situation may indicate that the coding stages enhance students' performance and knowledge of students in terms of quiz questions. To test that, a null hypothesis that states there does not exist any effects of coding stages on the overall quiz performance is formed, and Welch's t-test is used to compare the two of mean scores of first and last quizzes of all students to see if we can see a significant change. Once the t-test is applied, it has been seen that there is a statistically significant change between these two mean values.

Table 4.4: Quiz Analysis

Sets	**Mean**	**Result**
(First vs Last Quiz)	(0.518,0.571)	Reject Null Hypothesis.

School Analysis

The dataset contains 37 schools for individual game setup, where school information is hidden due to privacy concerns. Due to the probability that the overall performance of students in a school might be an indicator of individual performance in our model, we examined school-based overall scores to gain an insight about this metric. All student's scores are grouped based on their schools, and school names are mapped into dummy variables. The top 3 and bottom 3 schools are given in the table 4.5.

Table 4.5: Overall Performances of Students Based on Schools

Group	**School**	**Mean (Desc.)**	**Count**
Top Completed Levels	AM	8.26	30
	AR	8.15	39
	AO	7.76	26
Bottom Completed Levels	BA	6.07	41
	AL	6	48
	BF	5.64	37
Top Diamond Collected	AY	35.59	37
	AC	32.37	29
	AR	31.92	39
Bottom Diamond Collected	AE	21.74	39
	AT	20.91	34
	BF	18.87	37

When students' performances are compared based on the schools, it is revealed that schools have a high difference when the mean scores are considered, which could influence the prediction of a student's class and performance. However, due to the change in the number of students for each school and the small sample size per school might not reflect the effect of the school properly. Since we want to focus on the code-related fields rather than categorical variables of a student such as personal information, we decided to keep it outside of our clustering dataset to even all students in the clustering operation.

Whole data about school-based performance also shows that as the number of completed levels increases, the total diamond increases as well. This is an expected behaviour since getting more diamonds than other students is only possible by selecting a harder path than other students that are compared or completing more levels to reach more diamonds to collect. Hence, the distribution of easy and hard paths is balanced among different schools.

This analysis displayed that schools' performances are not similar to each other and they are likely to be a factor om students' problem-solving skills.

4.2 Experiments

Four different main experiments are conducted, as it was mentioned. The first two experiments, sequence-based and end-code-based experiments aim to understand if the chosen strategy has any effect on the general success of the students. The success parameter is selected as the number of diamonds collected due to the gaming using the metric as a ranking mechanism for students to decide on who will get the reward. For these experiments, other data such as some used blocks of each type or quiz scores are kept the same while, the data obtained for action sequence distance or end code distance are added to the experiment one at a time to see their effect on determining their effectiveness based on success.

The last two experiments are based on the strategy clusters that are generated using both action sequence distance and end code distance, to understand the strategy selection preferences based on gender and game mode, where in respective experiments, they will be dependents. The variables used for these experiments to answer research questions are explained in table 4.6

Table 4.6: Experiment Variables

RQ. No	**Added Variable**	**Examined Dependent Variable**
1	Sequence Dist. Data	Ind. Success
2	End Code Dist. Data	Ind. Success
3	End Code Dist. Data	Genders' strategy selection
4	End Code Dist. Data	Comp. Success

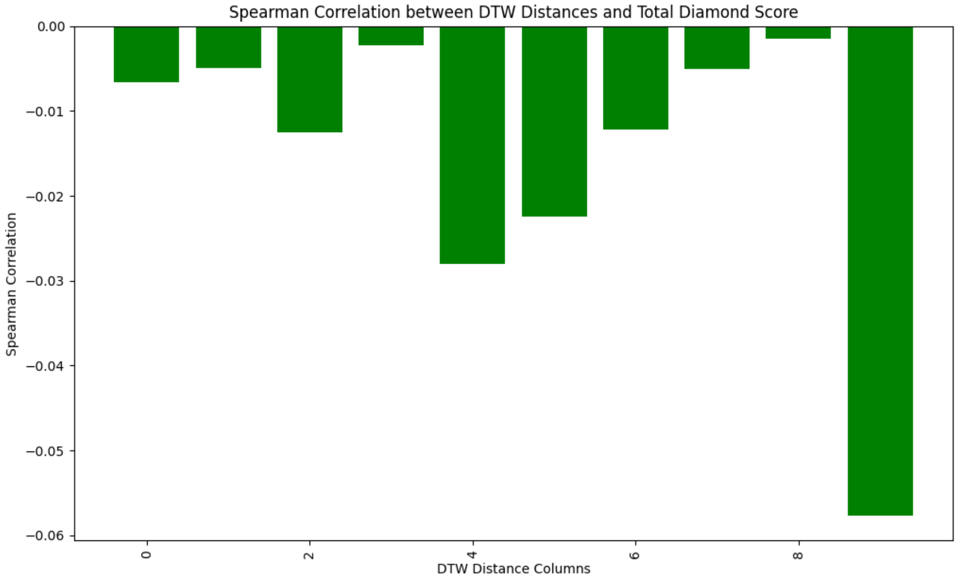
4.2.1 Action Sequence Based Experiments

RQ1: Do lower event sequence distances with an optimally written code imply higher scores for the given tasks?

To be able to understand if there exists a meaningful relationship between the event sequence distances and the success of students, the obtained results should be examined using statistical tests to see if they have any significant effect. The null hypothesis for this case is that there does not exist any relation between DTW distances and the success of the students. To understand the Spearman Test is applied to obtain correlation and p-value scores for statistical analysis.

The test results demonstrated that there does not have any strong correlation between DTW scores for each level and the success metric (total diamonds collected). The correlation scores varied between -0.01 to -0.05, which is considered to be a minor negative relationship, that does not imply any relation between given features and label. Moreover, the p-values for statistical significance also do not provide any statistically significant variable in this case as well.

Figure 4.3: Correlation scores obtained by Spearman Test regarding the relation between DTW distances and success metric



As it can be seen in the plot 4.3, even though later levels have a higher correlation compared to early levels, they still do not provide any significant correlation. Along with these values, p-values that determine the statistical significance of these variables do not indicate any evidence to reject the null hypothesis. Therefore, the event sequence distances that are obtained using dynamic time warping do not possess any value while trying to interpret the success of a student in the game environment.

To also validate our findings regarding the correlation between DTW and success in the game environment, analyzed the clusters that are generated using the DTW distances as well. By doing so, we can see if the clusters generated with this information provide any additional value about different strategies applied by students and their effectiveness.

Cluster	**Student Count**	**Avg. Comp. Levels**	**Avg. DTW. Dist.**	**Avg. Collected Dia.**
1	58	4.5	0.004	17.871
2	204	6.3	-0.09	24.20
3	952	7.11	0.2	27.67

Table 4.7: RQ1 Clustering Analysis

In table 4.7, the average distances are displayed with standardized values, therefore, as the value gets higher into positive, it means it has a greater distance than the average, and it is the opposite for negative values. By looking at the clusters and their characteristics, cluster 2 has the least average distance to the optimal event sequence, however, it is not low enough to distinguish itself from other clusters when their average distances are taken into consideration. Also, cluster distributions are not optimal since one cluster has too many members and its characteristics are not that much different than the other.

This situation also displays the insignificant correlation between dynamic-type warping distances and the success of students since we cannot come to a conclusion regarding the difference between the clusters. Therefore, we can conclude that the DTW method for distinguishing different strategies and their effects on the success of the students in the given game environment

4.2.2 End Code Distance Based Experiments

RQ2: Do lower end-code distances with an optimally written code imply higher scores for the given tasks?

The second method is end-code distance analysis which we try to implement to find out the different coding preferences as strategies. This method aims to analyze the correlation of end-code distances and the obtained diamonds, that is the success criteria, along the way. The correlation analysis is then backed by cluster analysis for examining different clusters and their characteristics, related to their play style and performance.

In this case, our null hypothesis is that there does not exist any relation between end-code distances and the success criteria. Contrary to the null hypothesis, the statistical analysis that is carried out by using the Spearman test indicates that there exists a moderate to strong relationship between end-code distances, which are obtained with the Levenshtein distance method, by having around 0.5 on average of all levels. Moreover, the p-values for all levels' end-code distances have statistical significance, which means they play a factor in the number of collected diamonds.

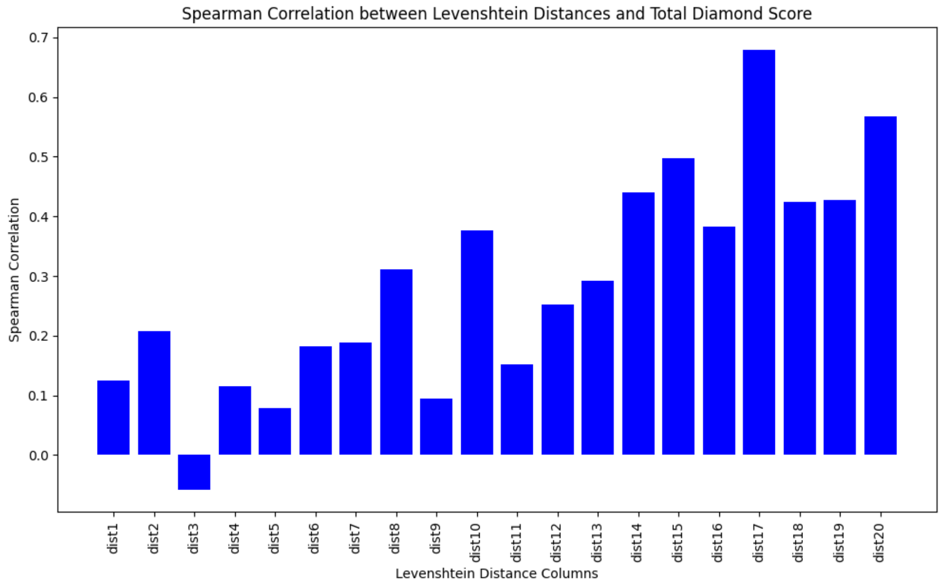


Figure 4.4: Correlation scores obtained by Spearman Test regarding the relation between end-code distances and success metric

The number of levels is provided as 20 for this correlation test since we also added the distances for practice levels to see their effect as well. As seen in the 4.4, the correlation between the number of diamonds collected and the average end-code distance is positive and increases along with the level number. This is probably caused by the fact that the advanced levels which start from level 15, are more difficult than the starting levels, therefore it becomes harder for students to write the correct code that is similar to the optimal one and most students probably used a more direct approach if they have made it to the higher levels.

DBSCAN Method

As the first approach, we use DBSCAN to cluster students' strategies. We expected to see well distinguished clusters which would give us a good amount of insight into the strategies and student's performance in applying them. Once the clustering is applied, the results are obtained as follows:

Clusters	Completed Levels	Avg. Lvn. Dist.	Collected Dia.	Student Count	Female	Male	Easy Count	Hard Count
0	5.71	7.91	19.12	447	234	213	3.91	2.70
1	8.19	14.21	38.34	415	168	247	3.20	5.18
2	6.70	4.68	22.38	365	193	172	3.60	3.19

Table 4.8: Cluster Analysis Based on Levenshtein Dist. Using DBSCAN

The cluster 0 is labelled as noise by the DBSCAN algorithm. The clusters seem to be formed in general by the average distance to the optimally written code. The algorithm shows that with a similar number of members, the students' who have higher average end-code distance have a higher number of diamonds, collected, thus, being more successful in the game on average. The cluster 1 contains more male students than cluster 2 when their density is considered. Even though several students do not have a high difference between clusters there is a notable change in the number of diamonds collected on average. According to the clusters' average level completions, we can also see that cluster 1 has the upper hand on finishing more levels, which is an expected situation since, to collect more diamonds, the best way for students would be to complete as much as levels as possible to be able to collect the diamonds in that levels. Being stuck in a level would considerably drop the number of collectables that a student has access to.

The main reasons behind having a lower number of accomplished levels and total number of diamonds collected for cluster 2 would be spending the most time on planning the algorithm for a longer period of time before acting, which would cause them to not see some levels and diminish the number of diamonds that they were able to collect.

Having around 1/3 of the population in the sample as a noise is not a good indicator, since the algorithm seems to not be able to add them into any clusters. However, we can see that data points labelled as noise also share the characteristics of being in the middle of the two clusters that we can separate as low and high distances clusters. Even though having those as noise causes, we still have two well-defined clusters that represent students who used the new techniques they learnt and the ones who chose to apply straight solutions to the problem. Therefore, by looking at their characteristics, we can also interpret them as the students who tried to mix and match both strategies but failed to do so in terms of success metrics.

GMM Method

Similar to the DBSCAN method, the GMM method is applied to create informative and well-structured clusters. The main difference expected is to see more intertwined clusters since the nature of GMM is a probabilistic method that leads to clusters that overlap due to assigning them based on their probability to belong to that cluster. With these in mind, the results obtained for GMM method are as follows:

Clusters	Completed Levels	Avg. Lvn. Dist.	Collected Dia.	Student Count	Female	Male	Easy Count	Hard Count
1	6.43	5.68	21.68	156	70	86	3.76	2.89
2	8.23	11.15	34.90	600	268	332	3.80	4.17
3	6.61	7.27	23.08	136	73	66	3.49	3.22
4	7.18	7.84	15.07	322	178	144	3.59	3.69

Table 4.9: Cluster Analysis Based on Levenshtein Dist. Using GMM

Due to low silhouette scores, we need to carefully examine the clusters to see if they provide any insight into the data and does clusters have anything different regarding strategy selection.

The clusters that are formed represent different characteristics than each other. The first attribute that is significant in our case is their average performance. The clusters can be grouped in three sections, high average distance, low average distance and medium average distance, since the average distance tells us the amount of difference between the optimal code that is generated according to what has been taught to the students in the previous steps. In this context, cluster 2 has the highest number of students along with the highest amount of collected diamonds. This situation also resembles the finding related to the connection between increasing distances and the collected diamonds. By looking at the number of students, we can also conclude that many students decided to follow a different coding method than the desired coding practice to easily pass the levels and obtain the prizes.

Cluster 1 in general has the lowest average distance to optimal code, which means that the students in this group selected a strategy that tried to write an optimal code for completing the levels, however, their overall performance indicates that they were not successful at implementing solutions that would require less blocks with the help of loops. On the other hand, cluster 2 represents students who have a much higher distance average to optimal code, yet they managed to collect more diamonds than any other cluster. This might highlight the situation where a more direct approach yielded more rewards for students. Cluster 3 seems to contain students who have a mix of both strategies, but their strategy did not increase their performance regarding collected diamonds.

For cluster 4, it seems like they tried a mixing method similar to cluster 3, however, they did not manage to successfully collect diamonds along the way. Moreover, they have the least amount of diamonds collected among all 4 clusters, therefore they contradict the finding about the positive relation with Levenshtein distance for end-code and the collected diamonds. However, since other clusters follow the positive correlation between those two features, and they make up 75% of the population of the sample, it is not enough to falsify the positive correlation.

By looking at all these findings with applying the Levenshtein distance method for distinguishing selected strategies, we obtained useful information to determine what kind of pros and cons each strategy might have, and can also provide special support for them based on their needs to develop their understanding of programming, so that they can easily adapt to more complex subjects of computer science in the future. Moreover, this method can be applied to later stages of this game, or other block-based programming games for providing a better learning environment.

4.2.3 Gender Based Experiments

RQ3: Is there a relation between gender and the selection of a strategy?

To understand if there exists any relation between the gender of the students and the cluster they belong to, we carried out some experiments including cluster analysis, along with statistical analysis to understand if gender plays any role in determining if a student belongs in a cluster or not. We first analyzed the distribution of each cluster based on gender. After that, by using logistic regression for binary classification for each cluster, tried to understand if gender is useful in determining whether a student belongs in a cluster or not.

When the 4.8 and 4.9 are examined, we can see each cluster's distribution of genders as well. For clusters 1 and 2 the male presence is more than the females, while the situation is the opposite in the rest of the clusters. By looking at the performance metrics of the clusters on average, cluster 2 has the highest number of diamonds collected per student, and it has a higher concentration of males. Also, in the descriptive analysis section 4.1, we also showed that males have a higher mean collected diamond per student than females. However, we cannot directly conclude that males have a tendency towards performing better than females since we cannot find significant evidence to reject the null hypothesis.

As for statistical analysis, the logistic regression method is used by reforming cluster values into binary values to make a binary classification task for each cluster in the end-code distance method that is shown in 4.8 and 4.9 to understand the effect of gender. Our null hypothesis is that gender does not affect when deciding if a student chooses a strategy. The obtained results demonstrate that, for clusters except cluster 1, we fail to reject the null hypothesis. The p-values do not satisfy the 0.05 threshold for accepting the gender variable for predicting whether the students belong to that cluster or not.

Clusters	Coefficient	Std. Errors	P-Values	Results
1	-0.433	0.179	0.015	Male > Female
2	-0.005	0.127	0.966	Fail to Reject
3	0.021	0.17110	0.901	Fail to Reject
4	0.211	0.015	0.09	Fail to Reject

Table 4.10: Logistic Regression Test

By looking at the table 4.10, it is seen that only cluster 1 has a statistically significant gender factor, and its negative coefficient shows us that males are more likely to be a part of this cluster. In other words, while trying to understand if a student is in cluster 1 if they are a male, they are contributing to the equation positively in the logistic regression equation.

When both the cluster analysis and statistical analysis are taken into consideration, the relation between gender and the strategy selection is weak apart from cluster 1's tendency towards male students. By looking at that, we might say students who are successful at implementing the optimal solution are a bit more likely to be male than female. Other than that, we cannot conclude performances and other strategy selections.

For the DBSCAN part of gender analysis, we have carried out the same method, the results are as follows:

Clusters	Coefficient	Std. Errors	P-Values	Results
-1	0.123	0.116	0.289	Fail to Reject
0	0.193	0.233	0.407	Fail to Reject
1	-0.214	0.151	0.156	Fail to Reject

Table 4.11: Logistic Regression Test

By looking at these results, we can see that we cannot reject the null hypothesis for the clusters we have generated by DBSCAN. Therefore, we cannot find any relation between gender and being a part of clusters when DBSCAN method is applied for clustering end-code distance data.

4.2.4 Game Type Based Experiments

RQ4: Is there a relation between the game setup and the selection of a strategy?

In this thesis, we also aim to analyze the effect of the game setup and its effect on students' strategy selection. As shown in previous experiments, students generally selected to implement a more direct approach by writing longer code chunks to finish levels, as can be seen in tables 4.8 and 4.9. We want to examine whether this situation continues in competitive game mode as well. To examine the effects, clustering operation has been applied to competitive game data. The main difference in competitive game mode is that students can see the progression of their competitor, therefore there is another factor which forces students to use their time efficiently. The clustering operation based on the end code method has been used for comparison since the end code method has been found as the better method for evaluating the similarity of students' code with the expert's optimal one.

Clusters	Completed Levels	Avg. Lvn. Dist.	Collected Dia.	Student Count	Female	Male
1	4.14	4.50	18.27	371	180	191
2	8.26	14.08	32.90	456	190	266
3	3.42	6.75	14.08	418	228	190

Table 4.12: Competitive Game Setup Clusering Analysis

When the results are compared, the cluster separation is higher than the individual performances. Students who chose to apply the direct approach had higher completion and total diamond scores, while the members of cluster 1 who tried to apply the methods shown in class performed poorly. Cluster 3 is in a position between the other two clusters, however, they got the least amount of average-level completion and diamond collection. These results demonstrate that for this experiment, students who decided to apply a more direct approach benefited the most. One of the reasons might be that, with the time limitation and the pressure of the competitors advancement through the game might have led students to underperform while trying to implement a code that requires fewer blocks but requires the understanding of the loop concept.

Summary of Results

The experiments explained above provide valuable insights regarding the student's approaches to coding challenges in a block-based programming environment and provide a method to distinguish different strategies. Among the two methods that are proposed, using dynamic time warping for action sequence comparison and end code distance method with Levenshtein distance have been examined. After comparing the results and discussing them, it has been shown that using end-code comparison is a better method for categorizing strategies implemented by students. It is evident that, the students who chose a direct approach and written a longer code performed well, while others struggled more to complete levels and collect diamonds. Regarding the relation between gender and generated strategy clusters, and seen that being male affects predicting whether a student is a part of cluster 1 or not. For other clusters, there does not exist any statistical evidence to reject the null hypothesis of not having any relation between gender and clusters. Moreover, the competitive game setup has been evaluated, where the results show similarities to individual setups. However, the number of clusters has been decreased to 3, yet the most successful cluster is again the one that has the highest average distance.

Chapter 5

Conclusion and Future Work

In this thesis, we developed two methods to analyze the students' coding preferences in a block-based game environment. The first method used dynamic type warping to compare the action sequence of students to an optimally written code's action sequence. The action sequences are processed to be compared and the calculated distances are used with other code features to cluster students. The method did not yield much value when the statistical analysis was done along with the correlation tests and the obtained distances did not show any resemblance with the students' obtained scores. Therefore, we have shown that the action sequence comparison is not a valid approach in this setup to create valuable feedback regarding to strategy selection.

The other method has implemented an edit distance method for the resulting codes of students for each level, to calculate the dissimilarity of the students' code with the expert generated one. In this approach, calculated similarities are used along with other related features to create clusters of different strategies applied by students. This method proved that the generated distances have a positive correlation with the success metric, the number of collected diamonds in the game. Also, the statistical analysis helped us to reject the null hypothesis and prove that there exists a monotonic relation between the end-code edit distance and the diamonds collected. This is an important finding for generating a feedback mechanism that can distinguish the applied strategy of the students and provide adequate support for the users in a block-based gaming environment.

The other findings are regarding the gender analysis, how they performed and whether it has anything related to strategy selection or not. For this question, we employed statistical analysis tools such as Welch's t-test and logistic regression. T-test showed that there exists a significant difference between the mean male and female scores based on the number of collected diamonds. For the strategy selection, the logistics regression for binary classification demonstrated that there is no relation between the gender and clusters that represent strategies, except for the cluster that has a lower edit distance than other clusters. In this cluster, males are more likely to be a part of, and gender is statistically important for this cluster to be used as a deciding feature.

For the game setup-related research question, similar processes are done with the competitive dataset, we have seen that students who follow a more direct approach and have a higher distance score, have a higher success score compared to the other strategies that would try to incorporate the concepts they have learnt such as loops.

As conclusion, they were able to distinguish the students who tried to use the shown concepts. Moreover, we have seen that there exist student groups, that did not use loop concepts, students who tried to use them on some levels, and the students who tried to solve problems using the loop concepts in general. By being able to decide who might be a part of which group, we can provide the required support for students who bypass the taught concepts to make sure they grasp them and not just continue to stay in their comfort zones.

Limitations and Future Work

The environment that the data has been collected is a specific purpose tool, that has been created for the "Improving Gender and Immigrant Outcomes through the Social Malleability of Attitudes: Randomized Interventions on Peer Interactions in an Educational Setting" project we have mentioned, and it does not have a high variety of choices when it comes to coding. The limited number of code blocks directs students to generate strategies based on using loops. This situation prevents us from testing to see if the students may have come up with strategies that have a high variety. Moreover, even though students can choose between two roads and can utilize loops in this step of the game, this mechanic does not exist in other games in the environment, therefore we are unable to see the strategy selection pattern of the students as a time series data. This situation leads us to only examine this part of the whole platform for understanding strategy selections.

Secondly, the found strategies may not be similar to other block-based coding tools. Since the used platform data does not contain much variety, but only works on seeing whether students may choose to go the hard way or the easy way, an environment that gives the ultimate freedom to users might have different characteristics for the strategies. The proposed methods are better suited for recognizing different strategies for a given curriculum and when it has an expected outcome that can be optimized.

Lastly, our dataset contained students between the ages of 9 to 13 who are not likely to have prior coding knowledge. Testing the proposed methods on freshman university students to see the differences in strategies they might come up with, and try to direct them to computer science-related areas based on their interests might be useful to extend the scope of the research.

In future work, the provided methods can be extended to calculate the dissimilarity with NLP techniques, where the code pieces can be turned into actual code scripts. Therefore the block-based codes can be compared with the regular scripts as well. This would open up a new perspective for research.

Moreover, for the block-based environments only, continuous research might be done on students in a different environment to see the development of understanding of the coding concepts and critical thinking skills of students based on their written codes in the environment. This would provide a far more advanced feedback advantage to deliver personal support for each student based on their learning patterns and knowledge states.

Bibliography

[1] O. Erol and N. S. Çırak, "The effect of a programming tool scratch on the problem-solving skills of middle school students," *Education and Information Technologies*, vol. 27, no. 6, p. 4065–4086, 2021. [Online]. Available: https://link.springer.com/article/10.1007/s10639-021-10776-w

[2] Saygıner and Tüzün, "The effects of block-based visual and text-based programming training on students' achievement, logical thinking skills, and motivation," *Journal of Computer Assisted Learning*, vol. 39(2), p. 644–658, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/jcal.12771

[3] A. Emerson, A. Smith, F. J. Rodriguez, E. N. Wiebe, B. W. Mott, K. E. Boyer, and J. C. Lester, "Cluster-based analysis of novice coding misconceptions in block-based programming," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 825–831. [Online]. Available: https://doi.org/10.1145/3328778.3366924

[4] J. Lee, S. Yunus, and J. O. Lee, "Investigating children's programming skills through play with robots (kibo)," *Early Childhood Education Journal*, 2023.

[5] D. Sun, C.-K. Looi, Y. Li, C. Zhu, C. Zhu, and M. Cheng, "Block-based versus text-based programming: a comparison of learners' programming behaviors, computational thinking skills and attitudes toward programming," *Educational Technology Research and Development*, 2024.

[6] M. Seraj, E.-S. Katterfeldt, K. Bub, S. Autexier, and R. Drechsler, "Scratch and google blockly: How girls' programming skills and attitudes are

influenced," in *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, ser. Koli Calling '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3364510.3364515

[7] U. AbdulSamad and R. Romli, "A comparison of block-based programming platforms for learning programming and creating simple application," in *Advances on Intelligent Informatics and Computing*, F. Saeed, F. Mohammed, and F. Ghaleb, Eds. Cham: Springer International Publishing, 2022, pp. 630–640.

[8] G. Gao, S. Marwan, and T. W. Price, "Early performance prediction using interpretable patterns in programming process data," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 342–348. [Online]. Available: https://doi.org/10.1145/3408877.3432439

[9] M. Kesselbacher and A. Bollin, "Quantifying patterns and programming strategies in block-based programming environments," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2019, pp. 254–255.

[10] Çakiroğlu and Çevik, "A framework for measuring abstraction as a sub-skill of computational thinking in block-based programming environments," *Education and Information Technologies*, 2022.

[11] T. Liu and M. Israel, "Uncovering students' problem-solving processes in game-based learning environments," *Computers and Education*, vol. 182, p. 104462, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360131522000331

[12] S. Marwan, B. Akram, T. Barnes, and T. W. Price, "Adaptive immediate feedback for block-based programming: Design and evaluation," *IEEE Transactions on Learning Technologies*, vol. 15, no. 3, pp. 406–420, 2022.

[13] A. Emerson, M. Geden, A. Smith, E. Wiebe, B. Mott, K. E. Boyer, and J. Lester, "Predictive student modeling in block-based programming environments with bayesian hierarchical models," in *Proceedings of the 28th*

ACM Conference on User Modeling, Adaptation and Personalization, ser. UMAP '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 62–70. [Online]. Available: https://doi.org/10.1145/3340631.3394853

[14] D. Reynolds, *Gaussian Mixture Models.* Boston, MA: Springer US, 2009, pp. 659–663. [Online]. Available: https://doi.org/10.1007/978-0-387-73003-5_196

[15] S.-G. Kwak and J. H. Kim, "Central limit theorem: The cornerstone of modern statistics," *Korean J Anesthesiol*, vol. 70, no. 2, pp. 144–156, Apr 2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5370305/