UNIVERSITY OF THE AEGEAN

DOCTORAL DISSERTATION

Development of a platform for learning programming using robots

Author: **Supervisor:**

Minas Rousouliotis **Dr. Ergina Kavallieratou**

A dissertation submitted in fulfilment of the requirements for the degree of Doctor of Philosophy (PhD) in the

Department of Information and Communication Systems Engineering

University of The Aegean

July, 2023

Είμαι ο αποκλειστικός συγγραφέας της υποβληθείσας Διδακτορικής Διατριβής με τίτλο «Ανάπτυξη Πλατφόρμας για εκμάθηση προγραμματισμού με χρήση Ρομπότ». Η συγκεκριμένη Διδακτορική Διατριβή είναι πρωτότυπη και εκπονήθηκε αποκλειστικά για την απόκτηση του Διδακτορικού διπλώματος του Τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου. Κάθε βοήθεια, την οποία είχα για την προετοιμασία της, αναγνωρίζεται πλήρως και αναφέρεται επακριβώς στην εργασία. Επίσης, επακριβώς αναφέρω στην εργασία τις πηγές, τις οποίες χρησιμοποίησα, και μνημονεύω επώνυμα τα δεδομένα ή τις ιδέες που αποτελούν προϊόν πνευματικής ιδιοκτησίας άλλων, ακόμη κι εάν η συμπερίληψή τους στην παρούσα εργασία υπήρξε έμμεση ή παραφρασμένη. Γενικότερα, βεβαιώνω ότι κατά την εκπόνηση της Διδακτορικής Διατριβής έχω τηρήσει απαρέγκλιτα όσα ο νόμος ορίζει περί διανοητικής ιδιοκτησίας και έχω συμμορφωθεί πλήρως με τα προβλεπόμενα στο νόμο περί προστασίας προσωπικών δεδομένων και τις αρχές Ακαδημαϊκής Δεοντολογίας.

Advising Committee of this Doctoral Dissertation:

Ergina Kavallieratou, Supervisor
Professor, University of the Aegean,
Greece

Efstathios Stamatatos, Advisor
Professor, University of the Aegean,
Greece

Charalabos Skianis,
Professor, University of the Aegean,
Greece

Approved by the Examining Committee:

Ergina Kavallieratou,
Professor, University of the Aegean,
Greece

Efstathios Stamatatos,
Professor, University of the Aegean,
Greece

Charalabos Skianis,
Professor, University of the Aegean,
Greece

Angelique Dimitracopoulou,
Professor, University of the Aegean,
Greece

Aikaterini Klonari,
Emeritus Professor, University of the Aegean,
Greece

Theodoros Kostoulas,
Associate Professor, University of the Aegean,
Greece

Papasalouros Andreas,
Associate Professor, University of the Aegean,
Greece

Declaration of Authorship

I hereby declare that I am the sole author of the submitted PhD Thesis entitled "Development of a platform for learning programming using robots". This Doctoral Thesis is original and was prepared exclusively for obtaining the Doctoral degree of the Department of Information and Communication Systems Engineering of University of the Aegaan. Any help I have had in its preparation is fully acknowledged and accurately referenced in the manuscript. Also, I accurately cite in the manuscript the sources I have used, and mention by name the data or ideas that are the product of the intellectual property of others, even if their inclusion in this work has been indirect or paraphrased. More generally, I certify that during the preparation of the Doctoral Dissertation I have strictly adhered to the provisions of the law on intellectual property and have fully complied with the provisions of the law on personal data protection and the principles of Academic Ethics.

Dedication

To my family, who stands by me with unwavering devotion.

Abstract

Motivating and fostering student engagement in their education can present considerable challenges, particularly within online learning environments. Educational Robotics has exhibited a multitude of benefits within the educational setting, encompassing not only the facilitation of teaching but also the development of various abilities such as creativity, problem-solving, and teamwork. In recent years, numerous methodologies have been developed, leveraging technological advancements to enhance the pedagogical process and augment students' aptitude for acquiring knowledge. Two educational approaches that have gained attention in recent years are game-based learning and gamification. These pedagogical approaches seek to optimize the learning experience by integrating game elements and concepts.

Learning to code has become increasingly important in the digital age. Technology's widespread usage made coding a desirable ability. It helps people understand and shape the digital world, offering many personal and professional growth opportunities. People are realizing the importance of teaching the next generation computational thinking and computer skills. Programmers are needed in software engineering, data analysis, academia, and many other fields of science. This demand has led to programming concepts being taught in most modern engineering programs. There is also agreement that Robotics Engineering programs should include software engineering.

This dissertation introduces an innovative, freely accessible teaching platform that aims to enhance the learning of Python programming and essential concepts in robotics. A novel Python programming framework was developed according to the Platform's requirements, allowing users to manage a robot remotely via the Internet. The adaptability of this framework allows for its possible utilisation with any robotic system capable of executing Python code. The robot can be operated through Python code or by using Blocky, a client-side open-source JavaScript library designed to develop block-based visual programming. The Platform can be used with any programming language Blockly supports with modest adjustments.

Literary sources influence the teaching approach utilised in the Platform and has resemblances to analogous endeavors, particularly with those that utilise virtual robotics.

However, a significant distinction exists in the Platform, which allows users to remote control an actual robot, facilitating a more creative, engaging, and pleasurable educational experience.

One of the objectives of this dissertation was to utilise the benefits of robotics inside the educational setting. Among these benefits is fostering creativity and innovation, as well as promoting computational thinking. This educational tool integrates elements of both game-based learning and gamification. Implementing this methodology amplifies the learner's satisfaction and active engagement in the pedagogical process while strengthening the comprehension of the fundamental concepts being taught. Furthermore, the employment of both Blockly and Python allows users to develop coding abilities in two separate coding environments, hence facilitating their familiarity and proficiency in the programming language being taught. Moreover, the Platform has the potential to be utilised in e-learning or blended learning environments, as well as in conventional classroom settings.

Among this dissertation's objectives is to evaluate the impact of employing the Platform in conjunction with two robots. The results of the conducted surveys indicate that the Platform intrigues its users and has a favourable influence on their views towards programming and robotics. Moreover, it exhibits a significant positive influence on their understanding of programming.

Τίτλος

Ανάπτυξη Πλατφόρμας για εκμάθηση προγραμματισμού με χρήση Ρομπότ

Περίληψη

Η παρακίνηση και η ενθάρρυνση της συμμετοχής των μαθητών στην εκπαιδευτική διαδικασία μπορεί να παρουσιάσει σημαντικές προκλήσεις, ιδιαίτερα σε διαδικτυακά περιβάλλοντα μάθησης. Η Εκπαιδευτική Ρομποτική έχει παρουσιάσει μια πληθώρα πλεονεκτημάτων όταν χρησιμοποιείται εντός του εκπαιδευτικού περιβάλλοντος, που περιλαμβάνει όχι μόνο τη διευκόλυνση της διδασκαλίας αλλά και την ανάπτυξη διαφόρων ικανοτήτων όπως η δημιουργικότητα, η επίλυση προβλημάτων και η ομαδική εργασία. Τα τελευταία χρόνια, έχουν αναπτυχθεί πολυάριθμες μεθοδολογίες, που αξιοποιούν τις τεχνολογικές εξελίξεις με στόχο την ενίσχυση της παιδαγωγικής διαδικασίας και της ικανότητας των μαθητών να αποκτούν γνώση. Δύο παραδείγματα εκπαιδευτικών προσεγγίσεων που έχουν κερδίσει την προσοχή τα τελευταία χρόνια είναι η μάθηση με βάση το παιχνίδι και η παιχνιδιοποίηση. Αυτές οι παιδαγωγικές προσεγγίσεις επιδιώκουν να βελτιστοποιήσουν τη μαθησιακή εμπειρία ενσωματώνοντας στοιχεία και έννοιες του παιχνιδιού.

Η εκμάθηση προγραμματισμού γίνεται όλο και πιο σημαντική στην ψηφιακή εποχή. Η ευρεία χρήση της τεχνολογίας έκανε τον προγραμματισμό επιθυμητό προσόν. Βοηθά τους ανθρώπους να κατανοήσουν και να διαμορφώσουν τον ψηφιακό κόσμο, προσφέροντας πολλές ευκαιρίες για προσωπική και επαγγελματική ανάπτυξη. Οι άνθρωποι συνειδητοποιούν τη σημασία της διδασκαλίας της επόμενης γενιάς υπολογιστικής σκέψης και δεξιοτήτων υπολογιστών. Χρειάζονται προγραμματιστές στη μηχανική λογισμικού, στην ανάλυση δεδομένων, στον ακαδημαϊκό χώρο και σε πολλούς άλλους τομείς της επιστήμης. Αυτή η απαίτηση έχει οδηγήσει στο να διδάσκονται έννοιες προγραμματισμού στα περισσότερα σύγχρονα προγράμματα σχολών μηχανικών πληροφορικής. Επίσης είναι κοινώς αποδεκτό ότι τα προγράμματα ρομποτικής μηχανικής θα πρέπει να περιλαμβάνουν και μηχανική λογισμικού.

Αυτή η διατριβή εισάγει μια καινοτόμο, ελεύθερα προσβάσιμη πλατφόρμα διδασκαλίας που στοχεύει να βελτιώσει την εκμάθηση του προγραμματισμού Python και βασικών εννοιών στη ρομποτική. Ένα νέο πλαίσιο προγραμματισμού Python αναπτύχθηκε σύμφωνα με τις απαιτήσεις της πλατφόρμας, επιτρέποντας στους χρήστες να διαχειρίζονται ένα ρομπότ εξ αποστάσεως μέσω του Διαδικτύου. Η προσαρμοστικότητα αυτού του πλαισίου επιτρέπει την πιθανή χρήση του με οποιοδήποτε ρομποτικό σύστημα ικανό να εκτελέσει κώδικα Python. Ο χρήστης μπορεί να χειριστεί το ρομπότ μέσω της χρήσης κώδικα Python ή μέσω του Blocky, μιας βιβλιοθήκης JavaScript ανοιχτού κώδικα από την πλευρά του πελάτη που έχει σχεδιαστεί για την ανάπτυξη οπτικού προγραμματισμού που βασίζεται σε μπλοκ. Με μικρές προσαρμογές, η πλατφόρμα μπορεί να χρησιμοποιηθεί με οποιαδήποτε γλώσσα προγραμματισμού που υποστηρίζεται από το Blockly.

Βιβλιογραφικές πηγές επηρεάζουν τη διδακτική προσέγγιση που χρησιμοποιείται στην Πλατφόρμα και έχει ομοιότητες με ανάλογες προσπάθειες, ιδιαίτερα με εκείνες που χρησιμοποιούν εικονική ρομποτική. Ωστόσο, υπάρχει μια σημαντική διάκριση στην Πλατφόρμα, η οποία επιτρέπει στους χρήστες να ελέγχουν από απόσταση ένα πραγματικό ρομπότ, διευκολύνοντας μια πιο δημιουργική, συναρπαστική και ευχάριστη εκπαιδευτική εμπειρία.

Ένας από τους στόχους αυτής της διατριβής ήταν να αξιοποιήσει τα οφέλη της ρομποτικής στο εκπαιδευτικό περιβάλλον. Μεταξύ αυτών των πλεονεκτημάτων είναι η ενθάρρυνση της δημιουργικότητας και της καινοτομίας, καθώς και η προώθηση της υπολογιστικής σκέψης. Αυτό το εκπαιδευτικό εργαλείο ενσωματώνει στοιχεία τόσο της μάθησης που βασίζεται στο παιχνίδι όσο και της παιχνιδιοποίησης. Η εφαρμογή αυτής της μεθοδολογίας ενισχύει τον βαθμό ικανοποίησης και την ενεργό εμπλοκή του εκπαιδευόμενου στην παιδαγωγική διαδικασία, ενισχύοντας την κατανόηση των θεμελιωδών ιδεών που διδάσκονται. Επιπλέον, η χρήση τόσο του Blockly όσο και της Python παρέχει στους χρήστες την ευκαιρία να αναπτύξουν ικανότητες κωδικοποίησης σε δύο ξεχωριστά περιβάλλοντα κωδικοποίησης, διευκολύνοντας έτσι την εξοικείωση και την επάρκειά τους στη γλώσσα προγραμματισμού που διδάσκεται. Επιπροσθέτως, η πλατφόρμα έχει τη δυνατότητα να χρησιμοποιηθεί σε περιβάλλοντα ηλεκτρονικής μάθησης ή μικτής μάθησης, καθώς και σε συμβατικές τάξεις.

Μεταξύ των στόχων αυτής της διατριβής είναι να αξιολογήσει τον αντίκτυπο της χρήσης της πλατφόρμας σε συνδυασμό με δύο ρομπότ. Τα αποτελέσματα των ερευνών που διεξήχθησαν δείχνουν ότι η πλατφόρμα προκαλεί το ενδιαφέρον των χρηστών και έχει ευνοϊκή επιρροή στις απόψεις τους για τον προγραμματισμό και τη ρομποτική. Επιπλέον, επιδεικνύει σημαντική θετική επίδραση στην κατανόησή τους για τον προγραμματισμό.

Acknowledgements

To Dr. Ergina Kavallieratou, I offer my most profound appreciation and gratitude. Her guidance and support over the past few years have been invaluable.

Table of Contents

LIST OF FIGURES

List of Tables

Chapter 1. Introduction

The significance of incorporating coding education into the curriculum has significantly increased in the modern age of digital advancements. The extensive integration of technology has resulted in a significant demand for coding as a highly desirable proficiency. Coding is a highly specialized skill that requires learning how to produce code or instructions for computers or, in other words, providing computers with explicit instructions on how to carry out a task. In order to be successful in computer programming, you need to be able to think critically, figure out solutions to issues, and apply computational thinking. Coding equips individuals with the necessary resources to comprehend and influence the digital landscape, hence offering abundant opportunities for personal and professional growth. With humanity's growing reliance on technology, the demand for programmers has expanded across diverse sectors, encompassing software engineering, data analysis, academics, and the arts [1].

The progression of technology in recent decades has yielded resolutions to technical challenges that have obscured the progress of robotics. The progress in the field of robotics and artificial intelligence has led to the creation of robots that have the potential to be utilised across various domains of daily life, including but not limited to medicine [2], teleoperation [3], tourism, hospitality [4], and entertainment [5].

Educational robotics is a specialised field within the subject of robotics that focuses on utilising robots for educational purposes. The utilisation of Educational Robotics (ER) in educational settings has been found to provide numerous advantages, including the facilitation of teaching and the cultivation of a diverse set of skills, such as innovation, problem-solving, and teamwork. ER can serve as a valuable educational tool across several grade levels for the teaching of STEM disciplines, including Computer Science [6]. Moreover, the utilisation of robotic technologies in education has been found to enhance student engagement and motivation, as students are able to observe the concrete outcomes of their endeavours.

As a consequence of the global COVID-19 pandemic, educational institutions across the globe were compelled to suspend in-person operations and transition to remote learning platforms. Numerous students and educators had difficulties as a result of this situation. However, it has also accelerated the implementation of e-learning and showcased the approach's transformative possibilities in reshaping the education field [7].

E-learning, often known as distant learning, refers to a pedagogical approach that only occurs through online platforms, eliminating the need for a conventional face-to-face classroom environment [8]. The rapid progression of technology has greatly aided the implementation of distance learning. The course materials can be accessed by students at their own convenience and from any location through the use of different electronic devices [9]. The majority of the terminology commonly used in the field, such as online, web-based, computer-based, or blended learning, all possess the capability to make use of a computer that is connected to a network. When utilised appropriately, online learning can enhance the quality of education by fostering greater engagement, stimulation, and flexibility for individual learners [10].

On the other hand, blended learning is distinguished by incorporating conventional classroom teaching approaches alongside digital coursework. Blended learning has gained significant traction in the field of education due to its perceived ability to augment the learning experience and maximise the utilisation of the classical classroom environment.

Gamification and game-based education are two educational methods that aim to enhance the learning process by incorporating game elements and concepts. Despite sharing similarities, game-based learning and gamification are separate concepts. Gamification encompasses elements derived from gaming, such as score systems, badges, and prizes, to motivate individuals to engage in repetitive or potentially monotonous activities, such as homework. In contrast, game-based learning adopts a more holistic strategy. This approach entails employing real games as pedagogical instruments, incorporating fundamental game mechanics and concepts to enhance knowledge acquisition.

1.1 Objectives

This dissertation presents a novel learning platform for Python programming that integrates game-based and gamification features. The proposed Platform has considered various learning theories, including Constructivism, Constructionism, Behaviourism, Social Learning, and Cognitive Load theory. The suggested Platform has the potential to serve as a tool for facilitating long-distance learning and being adaptable for usage in blended learning environments or traditional classroom settings.

The main objectives and contributions of this dissertation are as follows:

- A novel teaching tool designed to facilitate the teaching of python programming and fundamental principles of robotics.
- A novel python programming framework that enables users to remotely control a robot through the Internet. The versatility of this framework enables its application with any robot that can execute python code.
- A learning tool that incorporates features of game-based learning and gamification. Adopting this approach enhances the learner's level of enjoyment and active participation in the educational process, hence strengthening the fundamental principles being taught.
- A free-to-use platform that can be used by anyone interested in learning Python programming.
- A friendly and appealing user interface.
- An automated error feedback system
- An automated and anonymous system for gathering data that can be used to improve the learning Platform.
- The conducted surveys indicate that the Platform elicits interest among users and positively influences their attitudes towards programming and robotics. Furthermore, it has a substantial favourable impact on their comprehension of programming. The surveys analysis addressed the primary research inquiries as follows:

  RQ1: Does the Platform facilitate enhancing the survey participants' creativity and motivation?

  RQ2: How does the Platform influence survey participants' attitudes towards programming and robotics?

  RQ3: What is the impact of the Platform on the programming comprehension and skills of the survey participants?

  RQ4: Are the platform and framework utilised considered to be versatile?

1.2 Organization of the Dissertation

This dissertation is structured into a total of eight chapters. Chapter 1 offers an introductory overview of the disciplines implicated in the subject matter. In this chapter, the dissertation outlines its objectives and provides an overview of its organizational structure.

Chapter 2 delves into the topic of educational robotics, whilst Chapter 3 analyses the current state of integrating coding education in academia. Chapter 4 of the dissertation discusses E-Learning and Blended Learning, while Chapter 5 elaborates upon Game-based and Gamification teaching approaches.

Chapter 6 introduces the El Greco platform. The El Greco platform refers to the integration of the robot named El Greco with the Platform discussed in this dissertation. This chapter presents an analysis of the characteristics of the Platform and El Greco. Additionally, the discussion at the end of the chapter focuses on the outcomes of the conducted survey.

Chapter 7 is dedicated to the EDUV platform. EDUV is an underwater vehicle integrated into the Platform proposed in this dissertation. Furthermore, Chapter 7 presents the findings of the second survey conducted.

This dissertation concludes in Chapter 8, where a comparative analysis of the two surveys is conducted. Additionally, the prospective directions for future work are discussed.

Chapter 2. Educational Robotics

2.1 Introduction to Educational Robotics

Midway through the 20th century witnessed the birth of the scientific field of robotics, which saw substantial development thanks to the work of researchers such as George Devol and Joseph Engelberger, who created the very first industrial Robot in the late 1950s [11]. These first robots established the groundwork for the technology that would one day be utilised in educational institutions like schools and universities.

Seymour Papert later developed the LOGO programming language. LOGO is a programming language developed in the 1960s by Seymour Papert and his colleagues at the Massachusetts Institute of Technology (MIT). LOGO was developed to be understandable by young students, and it pioneered the idea of employing a "turtle robot" as a teaching aid in the education process. Using LOGO code, the turtle robot could move around on the floor and draw whatever forms were instructed [12].

Seymour Papert expanded the Piaget Constructionism theory and suggested that learning is more effective when students create specific objects of interest [13]. According to research [14], children who program a robot to move can investigate spatial concepts, problem-solving, measurement, geometry, and metacognitive processes. According to Papert, educational robotics can help "externalize" students' thoughts and make mathematical subjects "more open to reflection" [15].

In the 1980s, increasingly advanced robotics technology entered classrooms, having progressed beyond the turtle. For the first time, students could use LEGO bricks to construct working machines complete with gears, motors, and sensors while practising their coding skills in the Logo programming language. LEGO MINDSTORMS is LEGO's continuation of its robotics technology for students in grades K-12. Both official and informal classrooms made use of these robots [16]. As an early example of educational technology and robotics, MINDSTORMS had a significant impact and scalability when it was released in 1998 and sold out in three months [17].

Robotics is a field of science that is bustling with activity. Developments in other fields like engineering and computer science have allowed the dramatic advancements in robotics that have taken place in recent years. These advancements resulted in robots that can be used in everyday life. Robots nowadays have diverse applications; amongst others are used in tourism, hospitality [4], teleoperation [3], [18], entertainment [5], [19], medicine[2], [20] and elderly care [21].

Especially after the year 2000, researchers and teachers worldwide began investigating the feasibility of utilising robots in the classroom. Research was done to determine how using robots in the classroom will affect students' education and what methods would be most effective. Because of their efforts, educational robotics is now an official field of study [16].

Several studies have concluded that using robotics for education positively impacts motivation, collaboration and computational thinking [22]. Robotics can not only be used to teach robotics, but they are a tool of great value in science, mathematics, engineering, technology and informatics learning [23]. Educational robotics are used primarily for domain–specific learning subjects like science and technology [24], but they also can enhance cognition and social skills [25]. Furthermore, new technology and robotics attract young people, which is beneficial in the learning process [19]. Robots in education can take up the role of tools, tutors or peer learners [26].

Educational robotics is a subset of Robotics in Education, a broad term that refers to what robotics can do for students in the classroom. A categorization of Robotics in Education can be accomplished by using the target user, the area of learning and how the Robot is used in the learning process [24]. A more recent paper by Scaradozzi [27] uses three kinds of categorization. The first one (Figure 1) categorized robotics in education according to the formality or not of the learning environment, the impact on the school curriculum, the integration of the robotics tools and the way the activities were evaluated. Another categorization of Educational Robotics suggested by Scaradozzi is based on the classification of the tools used (Figure 2). Finally, Scaradozzi proposes categorizing Educational Robotics activities according to evaluation features (Figure 3)

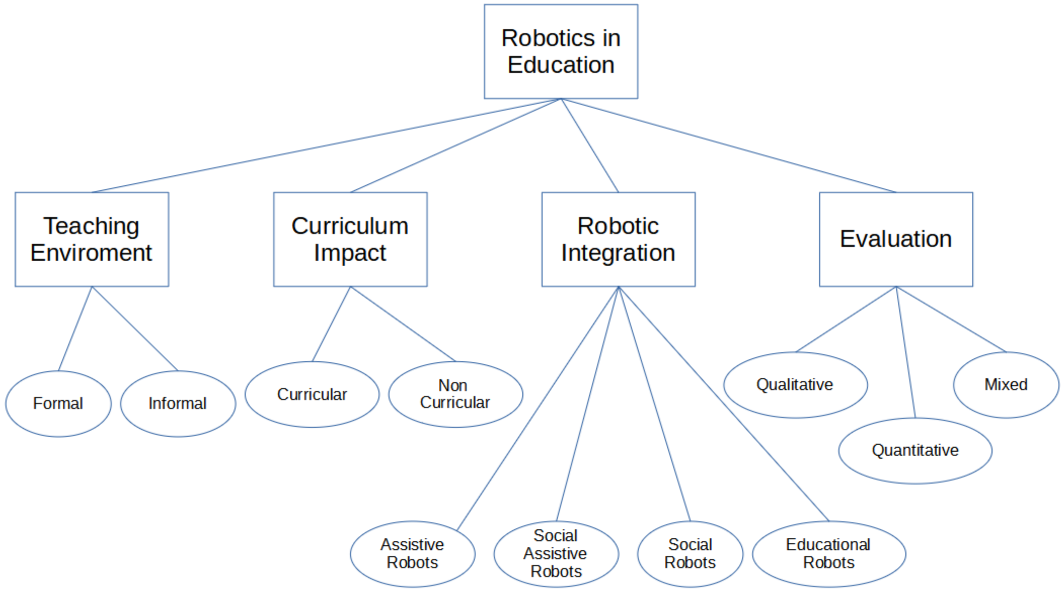


Figure 1: Categorization of Robotics in Education according to the formality or not of the learning environment.

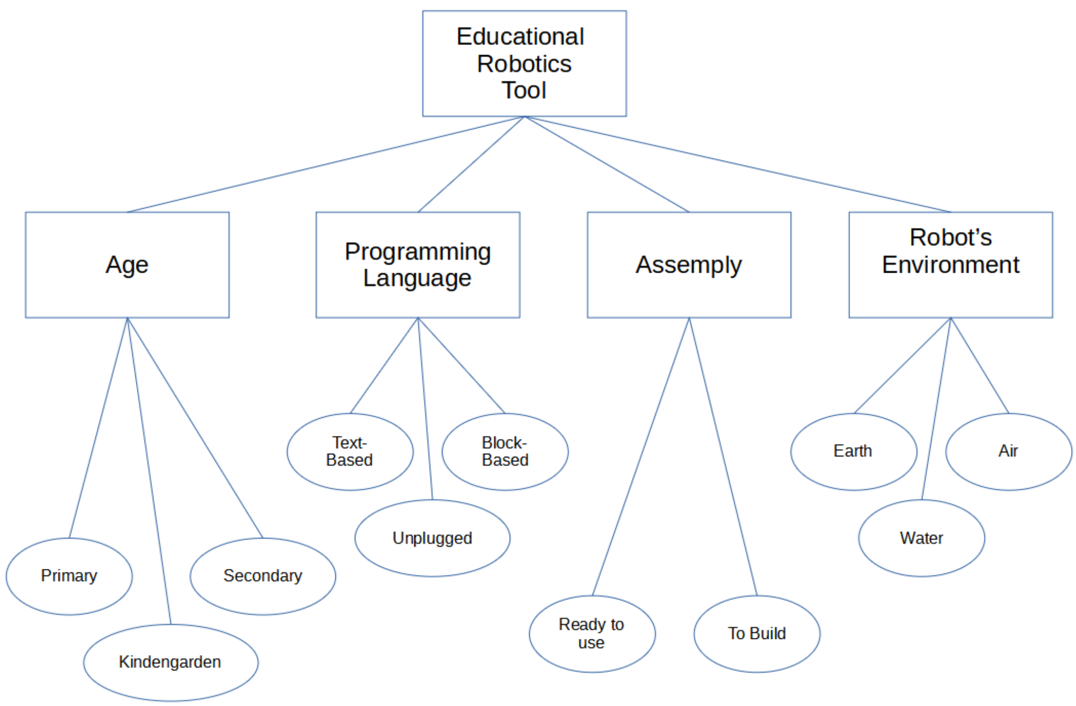


Figure 2: Categorization of Educational Robotics according to the features of the Robotic tool used.

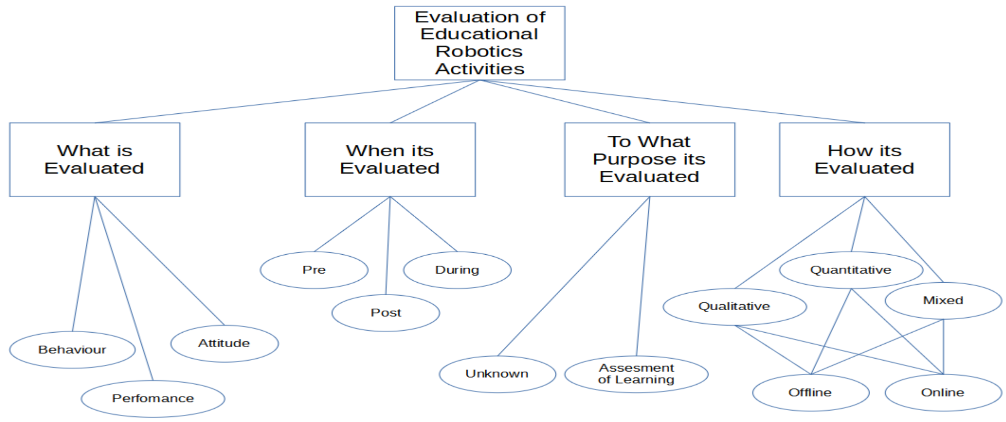


Figure 3: Categorization of Educational Robotics according to the features of the Evaluation of activities.

2.2 Educational Robotics and Learning Theories

Engaging in learning activities focuses on designing and constructing prototypes that allow students to acquire significant knowledge. These activities facilitate students' transition from a theoretical or abstract idea to a tangible or practical result [28]. Integrating educational robotics into the pedagogical process can be achieved through diverse teaching strategies, with one approach being prioritising robotics as a central component of educational programs [29]. The second approach involves utilising robotics as an educational tool [30], while the third approach entails using robotics to facilitate the advancement of learning [31]. The initial two methodologies prioritize the construction and programming of robots by utilising diverse components, sensors, and actuators. Additionally, they involve coding instructions in compliance with the syntax of a programming language [32]. The most current educational initiatives in educational robotics encompass these two approaches. These initiatives make a valuable contribution to education by developing teaching activities and providing laboratory courses [33]. The third strategy involves the integration of robots as instructional resources within educational environments. The utilisation of inquiry as a pedagogical approach has been identified as an appropriate approach for improving the process of learning [34]. Moreover, this particular approach considers the occurrence of errors as a valuable chance for learning [35].

This section of the dissertation will examine some learning theories connected to educational robotics.

2.2.1 Constructivism - Constructionism

According to Bravo and Forero [36], the primary source of material for educational robotics may be found in the study of constructivism and constructionism theories. Jean Piaget and John Dewey developed the theories of learning behind constructivism [37]. According to supporters of the constructivist approach to education, students should take an active role in their education rather than merely receiving information. Furthermore, students connect to what they already know to grasp new information better. Constructivism argues that each learner, drawing on their own experiences, constructs their own unique set of knowledge [38]. Through their interactions with others, students constantly challenge these beliefs. Different people have different mental models of how information is acquired and processed, shifting in this way the focus from the teacher to the student, who then uses various resources to direct his or her education [39].

According to constructivism, Individuals construct knowledge by expanding on their prior understanding and experiences. Constructivism can be used in educational robotics by giving students more chances to learn through direct experience. Designing, constructing, and programming robots provides students with a hands-on opportunity to learn about and apply STEM disciplines while they tackle real-world challenges [40]. Constructivism fosters knowledge building through active student participation by requiring students to work together, reflect, and revise their designs [41].

Seymour Papert expanded the Constructivism theory and suggested that learning is more effective when students create specific objects of interest [13]. Educational robotics has its roots in Seymour Papert's Constructionism learning theory and uses its basic principles. As a learning tool, it helps students explore and find new ideas. These ideas are tested on the Robot, and based on feedback from the robotic learning tool, children can expand their ideas or adopt new ones. In this way, knowledge is built on real-world experience, a fundamental principle of Constructionism [40]. As Papert claims, educational robotics can help "externalize" students' thoughts and make mathematical subjects "more open to reflection" [15]. Furthermore, Papert claims that an individual's level of knowledge is directly proportional to their interaction with the subject of their research [42]. Individuals can attain this level of interaction because educational robotics provides the means for them to do so. Finally, research [14] shows that children who program a robot to move can investigate spatial concepts, problem-solving, measurement, geometry, and metacognitive processes.

2.2.2 Behaviorism and Social Learning Theory

Similar concepts underpin the behavioural learning theory and the social learning theory. Both behavioural and social learning theories agree that social context influences people's actions. On the other hand, the social learning theory argues that individuals' thoughts and feelings also play a role in shaping their actions. Social learning theory argues that despite what students or individuals may observe being done, their internal ideas ultimately determine what kind of action response they exhibit [43].

Behaviourism does not focus on or even acknowledge the role of one's thoughts when it comes to behaviours. The learner is regarded as an inactive subject who only reacts to the various stimuli presented [44]. According to Skinner [45], a learner adapts his behaviour based on whether or not he is provided with positive or negative reinforcement by the instructor. Many aspects of behaviourist theory have been instrumental in developing significant educational technologies [46].

According to the social learning theory, behaviour is more complicated than the stimulus-and-response framework of behaviourism. It implies that pupils observe others' actions and deliberately mimic them to learn. Peer pressure and the need to fit in are two underlying emotions influencing behaviour [43].

In order to learn effectively, social interactions are highlighted as crucial in the Social Learning Theory. Collaborative activities and group projects in educational robotics can help students develop social skills. Together, groups may tackle problems, exchange insights, and boost each other's knowledge. Learning, talking, and working together are all enhanced by this method of interaction [25], [47], [48]. Teachers can also use elements of competition or friendly challenges to increase student enthusiasm and participation. This learning strategy is more commonly called gamification [43] and has attracted significant interest from the academic community in recent years.

2.2.3 Cognitive Load Theory

The primary concern of the Cognitive Load Theory is how the amount of mental labour required of students while being educated can affect the results of those students' education [49]. In the field of educational robotics, it is essential to consider the difficulty of the tasks at hand and the mental strain they put on the pupils. Educators can facilitate better learning by simplifying complex concepts and breaking them into smaller, more manageable parts. They can provide students with scaffolding and detailed instructions to help lead them through the learning process. Students can direct their mental resources toward developing problem-solving skills and grasping essential ideas when they have an effective strategy for properly regulating the cognitive load [50].

2.3. Educational Robotics Advantages

Educational robotics is a subfield of robotics that concentrates on using robots in education. It integrates aspects of the science, technology, engineering, and mathematics (STEM) curriculum with hands-on and interactive experiences that involve robots. This activity aims to provide students with a playful and exciting approach to learning the fundamentals of robotics, programming, and problem-solving. The main advantages of educational robotics are:

- Educational robotics provides students with hands-on opportunities to interact with robots, construct, program, and evaluate their creations. This method encourages knowledge-creation skills, logical reasoning, and problem-solving abilities [51], [52]. It has been shown that hands-on experience is essential for enhancing the understanding of fundamental engineering concepts [53]. Learning robotics challenges students to think critically, recognize issues, and create workable solutions. To program their robots, they learn to deconstruct complex problems into simpler ones and create step-by-step algorithms [48], [54], [55].
- Educational robotics fosters creativity and innovation. Students are encouraged to think creatively, generate original solutions, and implement their knowledge unconventionally [56], [57]. They can design and modify robots to complete specific duties or solve real-world issues [22], [58].

- The STEM disciplines of science, technology, engineering, and mathematics are integrated into robotics. Students gain knowledge of mechanics, electronics, sensors, and programming as they design, construct, and operate robots [59]. Educational Robotics can also be used as a teaching resource for computer science [6].
- Collaborating and working as a team is common in educational Robotics. Students are pushed to cooperate, exchange information, and take on a more significant role in group projects. Because of this collaboration, their social abilities are improved [25], [47], [48].
- Programming robots requires computational thinking, a vital skill for the twenty-first century [60]. Students are instructed in logical thinking, data analysis, decision-making, and algorithmic problem-solving [61]. These abilities apply in robotics and various fields outside of robotics. Students gain experience with real-world uses of technology through participation in educational robotics. They can investigate topics like automation, artificial intelligence, machine learning, and autonomous systems, which are increasingly important and considered 21st-century skills [62]. These skills are frequently associated with the accelerated development of technology and the transforming nature of the workplace [63]. Students interested in pursuing careers in science, engineering, technology, or related sectors can benefit significantly from studying robotics. It gives them competitive technical capabilities, critical thinking skills, and problem-solving aptitude [64].
- Students are naturally drawn to robots [65], so introducing them in the classroom is a great way to make studying exciting and interesting. It draws their attention, stimulates their desire to learn, and encourages a constructive outlook on STEM fields [66].
- Educational robotics is a varied field that may be adapted to fit students with varying learning requirements and abilities. It enables differentiated teaching, allowing all students to participate and learn at their own pace [67].

2.4. Educational Robotics Disadvantages

Educational robotics presents some disadvantages, but it is essential to keep in mind that these drawbacks can be minimized or eliminated with the help of appropriate planning, investments in resources and support, and an emphasis on educational practices that are inclusive and equitable.

- Educational robotic kits and equipment costs can be prohibitive for some institutions and individuals [68]. Some educational institutions or individuals may struggle to afford robotics hardware, software, and peripherals. In this way, access to resources and technical support for educational robotics may be restricted. Students and teachers can become frustrated when there is a lack of adequate resources for guidance, training, and problem-solving, making the learning experience less enjoyable [69].
- Educational robotic kits are often limited in their capacities and functionalities, which can frustrate students. Because of this limitation, students might be unable to investigate more advanced ideas or applications in the real world, potentially limiting their learning experiences [70].
- Educational Robotics, like any technology tool, has the potential to be addictive to students. When students depend more on technology to complete their tasks, they risk developing an addiction to those tools [71].
- Although robots improve the learning experience, they should not be viewed as a replacement for human interaction. Too much reliance on these devices by educators can result in educational violations that harm students more than benefit them [72].
- The field of robotics frequently uses advanced technologies such as computer programming, electronic engineering, and mechanical engineering. If a student or teacher does not have a firm basis in the STEM areas, they may have difficulty understanding and using these ideas [73].
- As with many other STEM fields, the field of robotics suffers from a lack of diversity among its practitioners. There are a variety of social, cultural, and systemic obstacles that may make it more difficult for students from

underrepresented groups, such as girls and minorities, to participate in robotics programs [73].

- Designing, constructing, and programming robots can take significant time [74]. When time is limited, as it often is in educational settings, incorporating robotics into the curriculum may require extensive planning and preparation from the instructor.
- Most robotics projects require cooperation and teamwork to complete. While this may be advantageous for developing social and communication skills, it may also present difficulties if there are disagreements within a group or if some students dominate the project, leaving others with fewer opportunities to participate and learn [75].

Chapter 3. Learning to Code and Education

In today's digital age, the importance of teaching students how to code has increased dramatically. The widespread adoption of technology made coding a highly sought-after skill. It gives people the tools needed to make sense of and shape the digital environment, providing numerous chances for personal and professional development. As humanity increasingly depends on technology, programmers are needed in various industries, from software engineering and data analysis to academia and the arts [1].

Coding is a highly specialized ability that involves learning how to create programs that a computer can execute. Coding refers to writing code or computer instructions, or in other words, just giving computers specific directions on how to do something. The ability to think critically, solve problems, and use computational thinking is necessary for success in computer programming. People are becoming more aware of how important it is to teach the next generation how to think computationally and improve their computer skills [60]. This demand has led to the inclusion of programming fundamentals in the curricula of nearly all modern engineering educational programs [76]. Furthermore, a widespread agreement exists that software engineering should be incorporated into Robotics Engineering curricula [77].

3.1 Advantages of incorporating coding into educational curricula

Several notable advantages are associated with integrating coding into educational programs. These advantages can be briefly described as follows:

- Reasoning and solving problems are essential in coding [78]. This skill is valuable in today's constantly developing technological landscape [79].
- The capacity to solve complex issues by decomposing them into smaller pieces and applying appropriate methods is a characteristic of computational thinking, which is an increasingly valuable ability in business and essential for gaining perspective on the world [57].
- Creating code can be a creative activity. Students must apply their creativity when finding coding solutions [57].

- There is a rising need for skilled programmers. One can enter a wide variety of fields after learning to code, including software engineering, data science, and web design [1].
- Students often need to collaborate on coding tasks to find solutions to challenges. Students can benefit from this by learning to work together and communicate more effectively. Collaboration and communication are essential 21st Century skills [79].
- The process of coding might provide difficulties, necessitating pupils to demonstrate resilience when confronted with obstacles. This intervention has the potential to facilitate the cultivation of students' determination capacity and foster their ability to effectively navigate and overcome obstacles [80].
- Students learning to code should be able to express themselves simply and concisely. Students can use this to improve their verbal and writing skills [81], [82].
- Acquiring coding skills is a valuable means of bridging the gap between theoretical knowledge and practical application, thereby mitigating the inherent abstraction often associated with STEM disciplines [83]. Students gain a sense of accomplishment and self-assurance when they witness the outcome of their hard work [84]. Furthermore, students develop a sense of ownership and pride in their education when they design and develop their own programs or apps [85]. Students develop a desire to succeed through this kind of teaching, which also inspires them to follow their curiosities and discover new things throughout their lives [86].

3.2 Integration of Coding into Traditional and Non-Traditional Classroom Environments

In the classroom, students can learn to code in various ways. Many new programs and materials have been developed in response to the rising need for coding education. Coding is no longer out of reach for potential learners of all ages because of free online resources like Code.org, Scratch, and Blockly. Many schools have also begun offering coding classes or incorporated coding into their already established curriculum [87]. People wishing to switch careers or advance in them may also benefit from the intensive training offered by coding boot camps and other forms of specialist coding academies.

3.2.1 LOGO

LOGO was the first attempt to make coding easier to learn by young pupils. It is a programming language designed specifically with novice programmers in mind. LOGO users learn programming fundamentals by straightforward words and instructions rather than memorizing theory or employing complicated programming techniques. LOGO is a simple programming language for young students that allows the instruction of an object, usually a turtle, to carry out a series of actions. LOGO's four main focuses are on helping kids get solid foundations in the following areas: critical thinking and problem-solving; communication; teamwork; inspiration and pioneering. Young users of LOGO can take an active role in the design and construction process with LOGO [12], [42]. Figure 4 depicts a modern LOGO setting [88].

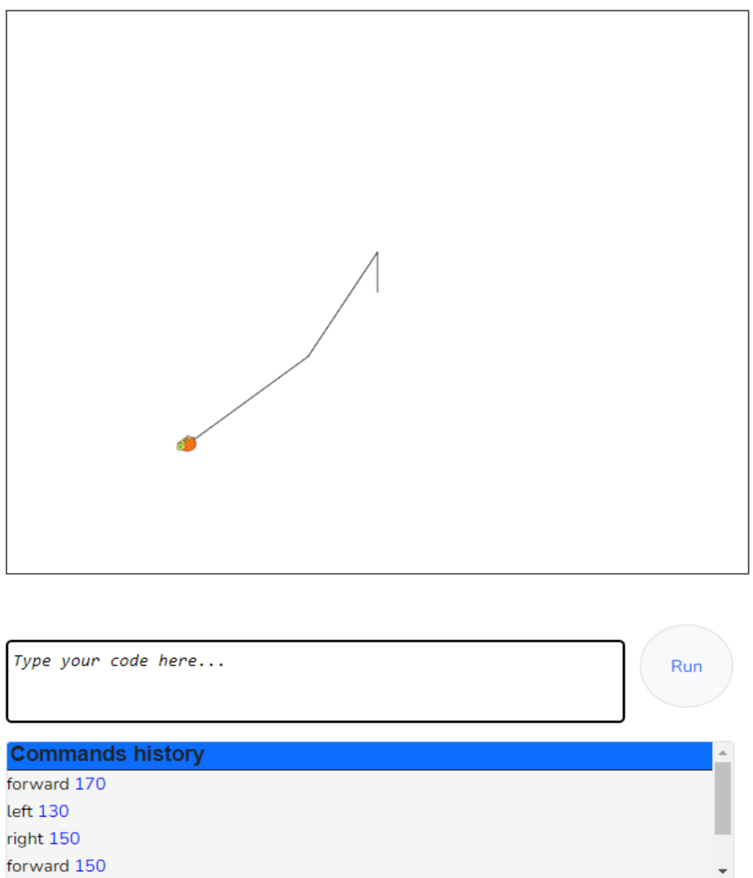


Figure 4: Logo environment of Turtle Academy [88].

3.2.2 Block-Based Programming

When it comes to introducing students to programming and computer science, block-based programming is quickly becoming the method of choice [89]. It has become a standard feature of the computer science curriculum thanks to the popularity of tools like Scratch and events like the Hour of Code hosted by Code.org. Many significant aspects of block-based programming set it apart from traditional text-based programming. Block-based programming utilises puzzle-styled blocks interconnected to each other, forming a list of blocks. Each block has a distinct functionality and corresponding colour.

3.2.3 Scratch

Scratch is a free programming language that allows users to create their own interactive stories, animations, digital games and stories [90]. Specifically, it features a scene in which the player produces "objects" (heroes and backgrounds) by picking them from a preexisting series or making their own. Depending on the user's predetermined behaviour, the scene's objects can engage in conversations with one another or with the user directly. The behaviour of the objects is achieved by dragging elements that imitate actions-commands, which refer to each object individually. Figure 5 exhibits the interface of scratch [91].

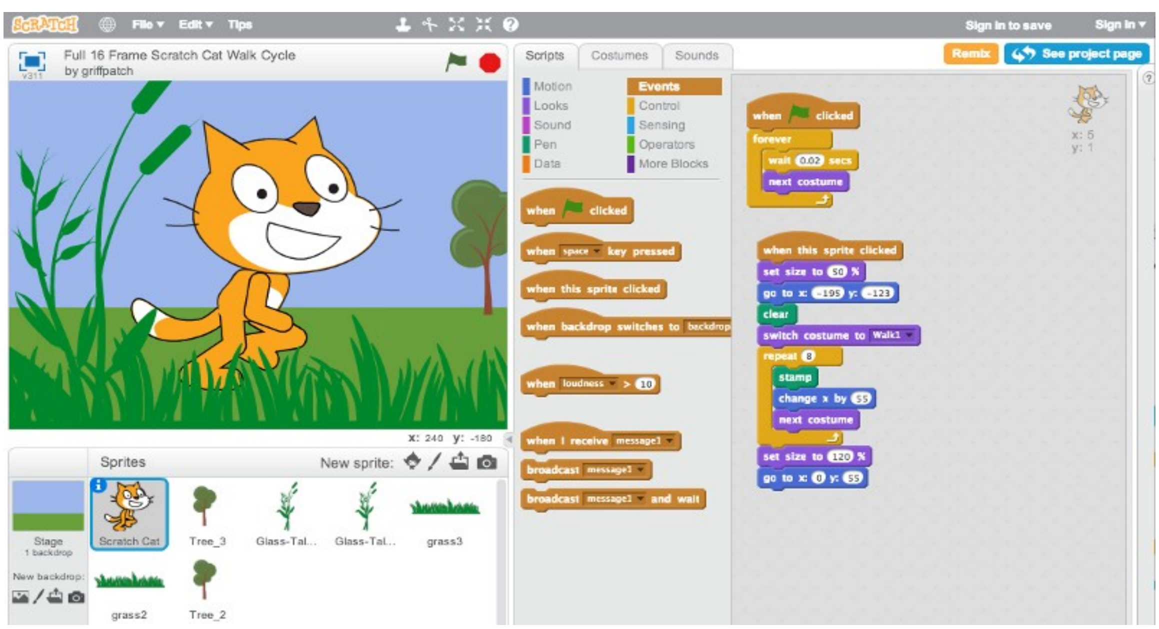


Figure 5: The interface of Scratch. [91]

3.2.4 Blockly

Blockly [92] is a client-side open-source JavaScript library for developing block-based visual programming and can be found in various platforms that teach computer programming. A scripting language like Python may be represented more comprehensibly using Blockly. In research, Blockly was used as a teaching tool to facilitate programming learning of a microcontroller called "M5Stack" [93]. By working with M5Stack devices, students may quickly get experience with Blockly Programming and then apply their newfound skills to the Internet by utilising the MQTT Protocol offered by the M5Stack graphical user interface. Students reported more comfort and enjoyment with programming after taking this approach. Figure 6 presents the Blockly interface [92].

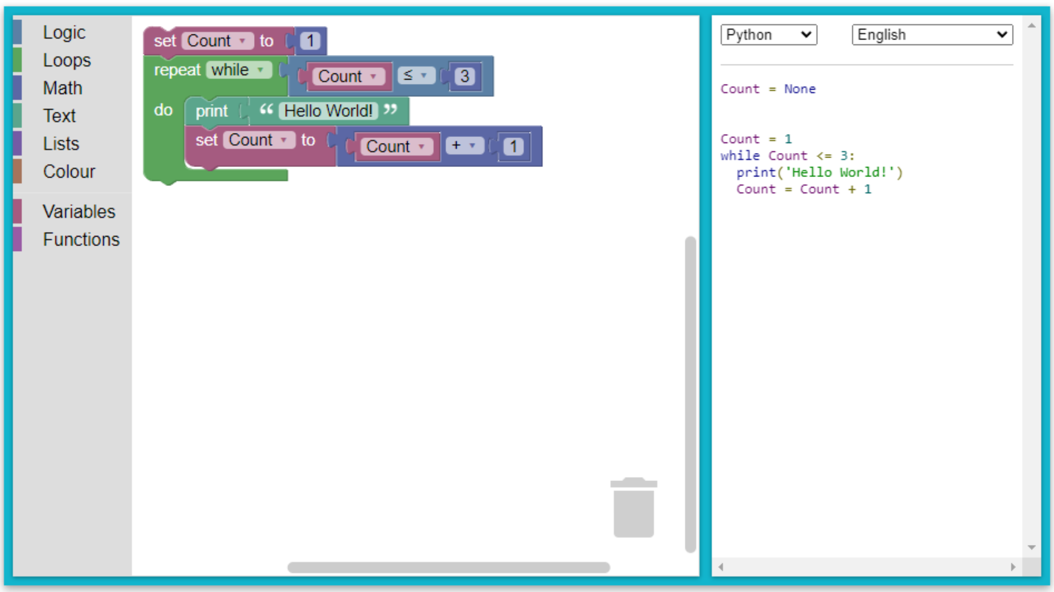


Figure 6: The Blockly interface [92].

3.2.5 PYTHON

Guido van Rossum developed Python, which was initially made publicly available in 1991. Over the years, Python's popularity has increased dramatically, making it one of the most popular languages for applications in web development, data analysis, artificial intelligence, automation, scripting, and many other areas. Python is a powerful and versatile language [94].

The Python programming language is highly suitable for educational purposes mainly for the following reasons.

- Python has been shown to help facilitate programming concept comprehension within a constructivist educational framework, particularly when used in the context of game-based learning. Using Python as a programming language in a game-based environment can dramatically improve programming concept acquisition [26].
- Python's straightforward syntax and robust capabilities render it appropriate for programmers of all skill levels. For this reason, it is highly recommended for use by beginners in computer science [95].
- Python's inherent flexibility renders it extremely useful for various applications, encompassing data analysis, web development, and robotic process automation. Consequently, Python appears as a versatile programming language that effectively introduces students to several domains within the field of computer science [96], [97].
- Python is an open-source programming language, meaning anybody can download and use it without cost [98]. For this reason, it is a fantastic choice for education because institutions and schools are financially constrained.
- Python has a large and productive community of users and programmers. As a result, there is a wealth of resources accessible to help those learning Python. Python's extensive standard library includes many useful pre-built modules and functions that save time and help students get started with programming right away. Learners are able to see immediate benefits, keeping them engaged and interested. Thanks to the popularity of the language, tutorials, manuals, and online courses are just some of the many instructional resources available for Python. The wealth of available resources is invaluable to students as they embark on their coding adventures [98].
- The Python language's interpreted nature allows instantaneous feedback during program execution. It is far simpler for students to spot and fix errors in their code when they can immediately view the results of their efforts [99].
- Python's support for object-oriented programming is a significant advantage. Teaching Object-Oriented Programming in Python helps students understand

fundamental software engineering principles [100] used in many contemporary programming languages.

- Python works well with various platforms and tools, like Raspberry Pi and microcontrollers. All the major robotics boards (Raspberry Pi, Arduino, Lego) may now be programmed in Python. Because of this interoperability, students can use what they learn in Python for things like creating robots and Internet of Things devices [94], [101].
- Python has visual programming tools like Scratch or Blockly, which offer a drag-and-drop interface, making it accessible to younger learners and those who are new to coding. These resources are great for beginners since they ease you into computer programming [89].

In secondary schools, Python is frequently used to teach students computer programming basics [102], [103]. Advanced programming ideas, such as data analysis and machine learning, are first introduced to students in higher education through Python [104]. People who like to study at their own pace can also learn Python through one of the many online classes that are available.

Chapter 4. E-learning and Blended Learning

E-learning, or distance learning, is a kind of teaching and learning that occurs entirely online, bypassing traditional classroom settings [8]. Rapid advances in technology have facilitated distance learning. Students can access the course materials from any location at their own pace using various electronic devices [9]. Most of the terminology (such as online, web-based, computer-based, or blended learning) share the capacity to utilise a network-connected computer. Online learning can potentially increase education quality when used correctly, making classes more engaging, stimulating, and flexible for each learner [10]. In other words, the focus of the teaching shifts more toward the students.

The development of online courses is an essential component of the digitalization of education because it enables educational institutions to fulfil the requirements of more recent forms of education, such as long-distance learning while overcoming the space-time constraints of traditional offline modes of teaching [105]. According to a study, there is no distinguishable difference in the learning outcomes that occur in a regular classroom compared to those that occur in an online classroom [106]. Despite this, the most essential aspect that contributes to students' success in online education is their level of motivation. Another study [107] highlights the factors influencing students' academic success in online education. In particular, when thinking about how the students conduct themselves, the only variables that have a significant effect are their levels of motivation and the setting they are in. The findings of this study indicate that behaviour factors such as the characteristics of the teachers, levels of student motivation, and organizational structures all appear to boost student performance. Similarly, the findings of a study [108] indicate that each service quality category has a positive association with the levels of motivation and overall satisfaction experienced by the students. In other words, there is a positive association between the level of motivation that students have and the academic achievement they achieve.

Blended learning is characterized by integrating traditional classroom teaching methods with digital courses. Blended learning is experiencing a rise in popularity as educators increasingly recognize its capacity to enhance education and optimize students' in-class time. Regarding this matter, a survey conducted after the completion of the course on "Data Structures and Algorithms" indicated that students not only expressed satisfaction with the teaching methodology employed but also demonstrated exceptional performance [109]. Additionally, it is noteworthy to add that there was a substantial decrease in the dropout rate, possibly due to the satisfaction students experienced with the assistance provided by their educators and the educational system. Another example, as described by [110], involved implementing a blended classroom approach in the context of the course "Principles of Chemical Engineering". This study demonstrated that using micro-videos and a blended approach of online and offline learning resulted in a notable enhancement in learning efficacy, as observed by the student's independent study. Blended learning encompasses using advanced educational technologies and digital resources while redefining students' learning ideologies and instructors' teaching methodologies [111].

Schools and institutions worldwide were forced to close their doors and turn to online learning due to the COVID-19 pandemic. Many students and educators struggled because of this, but it has also expedited the adoption of e-learning and demonstrated the potential of this approach to radically alter the educational landscape [7].

Chapter 5. Gamification – Game-based Learning

Gamification and game-based learning are two educational strategies that endeavour to augment learning by using game elements and principles. Although they exhibit similarities, game-based learning and gamification are distinct concepts. While gamification incorporates certain parts of gaming, such as score systems, badges, and rewards, to motivate individuals to engage in repetitive or potentially monotonous tasks like homework, game-based learning takes a more comprehensive approach. It involves utilising actual games as educational tools, integrating core game mechanics and principles to facilitate learning [112]. Thus, game-based learning goes beyond superficial incentives and engages learners in interactive and immersive gaming experiences, directly contributing to their educational development. The primary objective is to cultivate intrinsic motivation and foster a flow state in which the player acquires knowledge and skills without conscious awareness, as their attention is directed towards the act of playing [112].

Both methodologies have demonstrated favourable impacts on educational achievements; nevertheless, their efficacy depends upon various conditions. When deciding which teaching approach to follow, educators should seriously consider the individual learning objectives and the needs of their learners. In the remainder of this chapter, both approaches will be discussed.

5.1 Gamification

Gamification refers to using game-like elements and techniques in non-gaming contexts to enhance user engagement and motivation [113]. In recent years, educators have grown to accept the potential of gamification as a tool for enhancing the learning experience. Through the incorporation of gamification aspects inside educational environments, educators possess the ability to cultivate dynamic and engaging learning experiences that engage pupils and facilitate the development of a deeper understanding of complex concepts. Utilising this tool can enhance student motivation and retention, facilitate cooperation and teamwork, and individualize the learning experience [114].

Learning theories, such as behaviourism, constructivism, and cognitive load theory, provide a framework for a better understanding of how gamification can support learning and engagement. By incorporating these theories into the design of educational games, educators can create more effective and engaging learning experiences [115].

Some common gamification techniques include using points, badges, and levels, creating challenges and quests, using leaderboards and competition [116], and using social features to share their progress. Social features can include forums, chat rooms, and social media integration [117].

Gamification is a relatively new field, and there is still much research on its effectiveness. However, the evidence to date suggests that gamification can be a powerful tool for improving learning outcomes. It has been successfully applied in various fields, including marketing, work organizations, health, and environmental initiatives [8].

Gamification in education, specifically in the context of learning to code, has gained significant attention in recent years. The traditional approach to learning to code can be intimidating and discouraging for many students, leading to disengagement and hindered progress [118]. Gamification offers a powerful solution to this challenge, transforming the process of learning to code into an exciting and empowering journey for students of all ages. By incorporating game elements and mechanics, gamified coding platforms foster creativity, motivation, engagement, and critical thinking, while providing goal-oriented learning coupled with immediate feedback and personalized learning experiences. The gamification approach offers collaborative learning opportunities and ensures that students develop a love for coding, embrace challenges, and continue to pursue knowledge [119], [120].

Overall, gamification is a promising approach to improving learning outcomes. By harnessing the inherent appeal of games, educators can create immersive, interactive, and personalized learning experiences that enhance students' academic performance and foster a love for learning. As technology continues to evolve, the potential for gamification in education remains limitless, opening doors to new and exciting possibilities for the future of learning. Nevertheless, to maximize its effectiveness, gamification should be tailored to the specific needs and preferences of the learners and the educational context [121].

5.2 Game-Based

The concept of game-based learning entails incorporating educational content and objectives within the framework of games or game-like activities [122]. In contrast to conventional teaching approaches characterized by passive information transmission, game-based learning fosters active engagement, problem-solving abilities, critical thinking skills, and collaborative interactions among learners [123][124]. Educational games encompass various genres, from digital video games and applications [125] to traditional board games, card games, and immersive role-playing activities [126]. Whatever the medium employed, the basic principles of game-based education exhibit a consistent nature, namely the cultivation of active engagement, the enhancement of educational achievements, and the stimulation of intrinsic drive [112].

Numerous factors contribute to the efficacy of game-based teaching. To begin with, it might be argued that games possess a natural element of motivation. A combination of challenges, rewards, and a sense of accomplishment can effectively maintain student engagement. Furthermore, games frequently include an interactive nature, enabling learners to actively engage in acquiring knowledge. This can facilitate enhanced comprehension and long-term retention of the acquired knowledge. Furthermore, games have the potential to replicate real-life scenarios, facilitating in this way the cultivation of problem-solving abilities and critical thinking among learners [127]. Game-based education has the potential to serve as an effective teaching tool for a diverse range of academic disciplines, encompassing Mathematics, Science, History, Language, and the Arts [128].

The traditional methods of coding education can occasionally present challenges that result in decreased motivation and a sense of discouragement, particularly for novices in the field [118]. Game-based education presents a compelling and effective alternative for teaching coding, combining gaming principles with programming concepts [129]. Games can attract and motivate individuals, hence facilitating the maintenance of concentration and motivation among learners. Furthermore, games provide an interactive feature that enables learners to engage in coding exercises inside a secure and encouraging setting. Numerous studies have demonstrated that the integration of game-based approaches in educational settings may produce positive results in the field of coding education. The aforementioned outcomes include cognitive, behavioural, and emotional aspects. Moreover, the effectiveness of game-based teaching in the field of coding can be influenced by various aspects, including the particular game or platform employed, the teaching approach implemented, and the individual characteristics of the learners [130]. Game-based coding platforms provide a secure and controlled setting for individuals to explore and experiment with coding principles. Coding education has incorporated a range of game-based platforms and tools. These platforms offer learners prompt feedback and reward their progress through gamified prizes [112].

Chapter 6. The EL Greco Platform

6.1 Introduction

The El Greco Platform allows potential learners to remotely operate El Greco, a social humanoid robot that is affordable, easily constructed, and suitable for educational purposes. El Greco can execute several tasks, including combined movements, engaging in social interactions across multiple languages, and online information retrieval. The robot's capabilities can be programmed by the direct input of Python code or by utilising the Blockly library, which incorporates an editor within an application that visually portrays coding principles as interconnected blocks.

6.2 Related Work

Similar to Blockly, Scratch is a block-based programming language that enables users to generate animations, video games, and digital storytelling [90]. Although the Scratch coding environment has widespread success, the El Greco platform presents a number of distinct advantages due to its use of a real robot for teaching the Python programming language.

Blockly has been employed in various computer programming educational platforms. One example is BlockPy [131], a platform specifically designed for Python that offers a variety of compelling features. These features include guided feedback on the code output, CORGIS integration [132], and the ability to generate line graphs, scatter plots, and histograms. Additionally, a Python to Blocks translation option [133] can be used to turn written Python code into Blockly blocks.

While our work is closely connected to BlockPy, one of our objectives was to leverage the advantages of robotics in the educational context. Reeborg's World [134] affected the development of the El Greco Platform. Reeborg's World offers an offline version that is also an internet version and successor of RUR-PLE [135], a desktop application. Its construction aimed to facilitate the acquisition of programming skills among novices while providing an enjoyable experience. For teachers, this tool proves to be beneficial in computer programming teaching. It focuses primarily on Python, but JavaScript is supported as well. Reeborg is a virtual robot that can be controlled using

Python or JavaScript. It can transform into different robotic shapes and navigate inside a virtual environment that includes a variety of objects capable of interacting with both the robot and each other. As an example, it is possible to generate a tile containing water, which has the potential to submerge Reeborg. Reeborg possesses the capability to recognize the presence of certain tiles and effectively employ a bridge tile in order to bypass them. In this manner, individuals can generate complex labyrinths or even Sokoban puzzles [136] with predetermined objectives. One of the notable features of Reeborg's World is its step-by-step execution mode for Python code, which demonstrates a high level of precision.

Our approach to Python was inspired and presented similarities to Reeborg's World and other projects of a similar nature [137], [138]. However, the key feature of the El Greco platform is that users may operate a real robot from a distance, making education more imaginative, engaging, and enjoyable.

Several robotic kits permit the programming of an educational robot using Python and Blockly [139]–[143]. USB, Bluetooth, or a wireless network are typically utilised to connect the user interface to the robot. The previously mentioned robotic tools contribute significantly to learning, but the El Greco platform employs a social humanoid robot. Compared to other types of educational robotics, this significantly enhances the learner's creativity, interest, and enjoyment. In addition, the El Greco platform employs the Internet and, more specifically, a web page to facilitate the user's interaction with the robot. Consequently, the El Greco platform enables anyone interested in learning Python to do so while operating a robot. In most instances, purchasing an educational robot with the same capabilities as El Greco is prohibitively expensive.

The basis of our work is rooted in a pre-existing educational platform [144] developed by the Aegean Robotics Team. In order to enhance the platform, a brief survey was conducted. Two questionnaires were administered, one prior to the platform's utilisation and another after its utilisation. A total of 15 children responded to both questionnaires, with one child from the first grade, five from the second grade, and nine from the third grade of the lyceum. The primary objective of the initial study was to assess the participants' existing computer programming abilities and their level of interest in utilising robots within an educational setting. The results indicated that most participants, precisely 65.5%, reported lacking prior programming knowledge. Furthermore, a significant proportion, 79.3%, expressed confidence in the potential of robotics to enhance their learning experience. Additionally, a substantial percentage, 75.9%, believed that robotics could be a valuable tool for educational purposes. Moreover, a significant majority of individuals, precisely 69%, express a desire to have unrestricted access to the platform regardless of time or location (Figure 7).

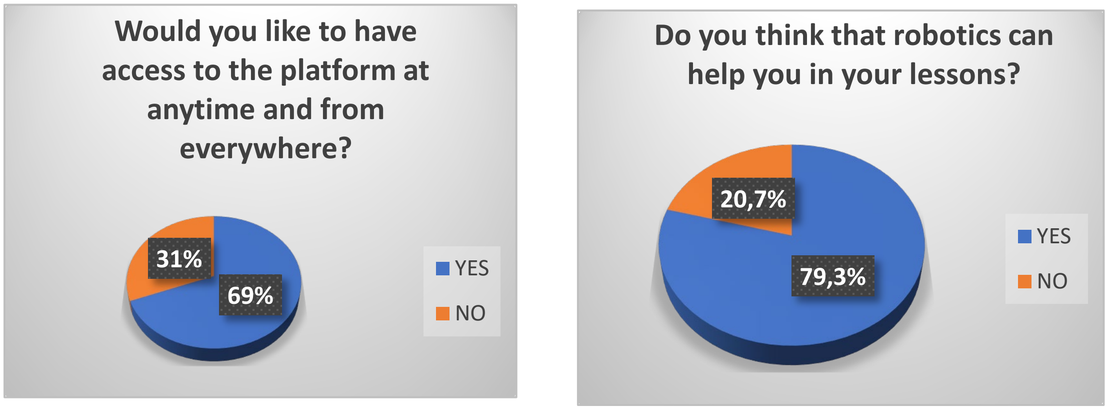


Figure 7: Findings of the first questionnaire

The second questionnaire focused on utilising El Greco and the student's interactions with the robot and its programming interface. The survey findings indicate that the platform had the capacity to facilitate the integration of STEM education. The level of acceptance demonstrated by the pupils towards the platform was exceedingly positive. Most participants, precisely 86.2%, had a positive sentiment towards the event, while 65.5% indicated a strong interest in utilising it again (Figure 8) [144].

The El Greco platform can be considered an enhanced version of the previously described platform. The enhancements to the platform above are:

- The direct execution of Python code is facilitated using Brython [16].
- Examining code for mistakes and providing correction messages to the user.
- The process of executing code in a step-by-step manner.

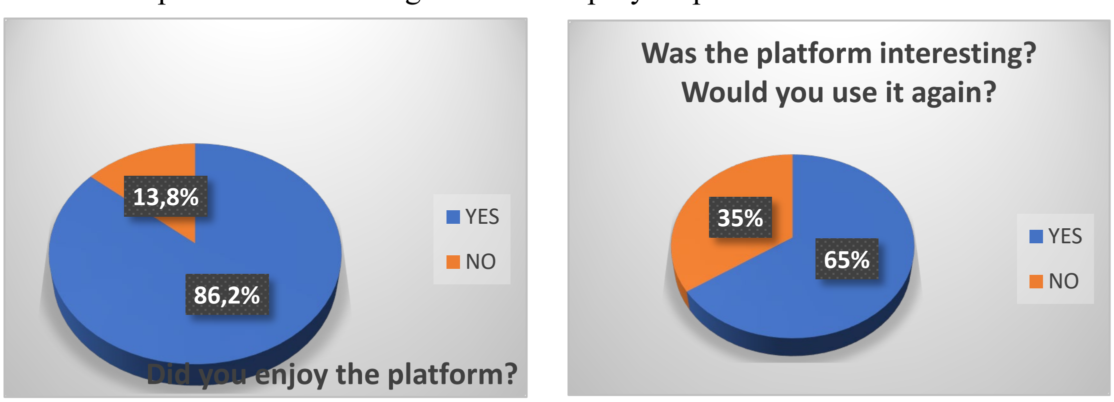


Figure 8: Findings of the second questionnaire

- Implementing a reservation system for the utilisation of the El Greco robot.
- The utilisation of an administrator application designed to facilitate remote management of reservations and other related tasks.
- The El Greco adventure mode.
- Improvement of the Website's user interface and visual aesthetics.
- Website fundamentals such as registration, verification, a Python tutorial, and profile construction.
- Automatic creation of log files that collect information anonymously about the user's performance and interaction with the webpage and the robot. The provided information has the potential to be utilised for enhancing the platform.

The subsequent sections will provide a more comprehensive analysis of these topics.

6.3 The humanoid robot EL Greco

The primary objectives behind the creation of the humanoid robot known as El Greco (Figure 9) encompassed the following aspects:

- The development of a cost-effective humanoid robot.
- A humanoid that utilises pre-existing knowledge while also integrating innovative components.
- Suitable for implementation in educational and tourism contexts.
- The incorporation of artificial intelligent systems that include extensibility and versatility, enabling their use across various applications [145].

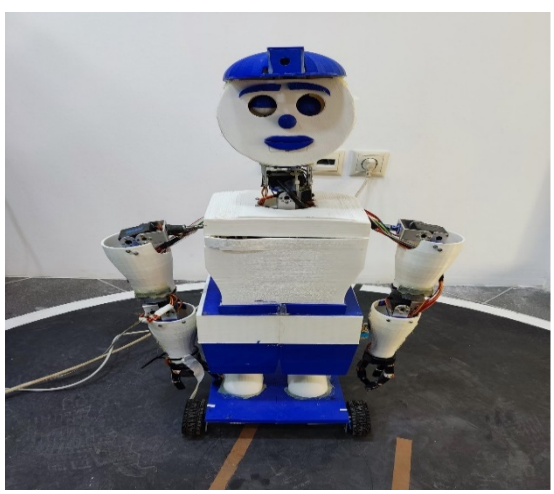


Figure 9: The robot named El Greco

The humanoid robot is often regarded as a highly desirable and effective choice in the field of education. The avoidance of the Uncanny Valley phenomenon is crucial in this context. The concept of the Uncanny Valley refers to the condition of exhibiting a high level of resemblance to humans. The presence of high similarity between humans and virtual humans has been seen to elicit feelings of anxiety and discomfort in human individuals during their interactions [146]. To address this concern, while designing a humanoid robot intended for children, a commonly employed strategy to mitigate the Uncanny Valley phenomenon involves imposing constraints on its range of motion and imposing substantial limitations on its hardware capabilities [147]. Moreover, the high expense of humanoid robots renders them unsuitable for widespread implementation in educational settings.

The design of El Greco incorporates all the aforementioned factors. El Greco exhibits anthropomorphic characteristics. The robot's appearance has been intentionally crafted to resemble a mechanical child between the ages of five and six. The robot can convey emotions through imitating facial expressions, including but not limited to happiness, sadness, agreement, disagreement, surprise, and drowsiness. Additionally, El Greco demonstrates proficiency in the execution of winks and blinks. The movements of the eyelids and eyebrows significantly influence nonverbal communication between individuals and serve a crucial function in the expression of emotions [148]. El Greco possesses the ability to engage in communication in Greek as well as 160 other languages. Additionally, El Greco can do tasks such as greeting individuals, identifying and comprehending spoken language, recognizing faces, and accurately determining the location of an individual within a group of people. El Greco can access information from the Internet relating to many subjects, such as weather conditions, news updates, and other relevant topics. The programming language used exclusively for the development of El Greco is Python. The code development process conformed to the component-based methodology, which is known for its ability to expedite development, enhance code maintenance and upgrades, and facilitate system extension. The El Greco movement is accomplished by utilising 25 servomotors of diverse specifications, resulting in a total of 25 degrees of freedom. The utilisation of complex servo motors was implemented in a comparable manner, mirroring the approach employed by Team Hector in their work on

Jonny 05 [149]. Furthermore, El Greco movement capabilities drew inspiration from the design principles employed in the development of the Robot CHIMP [150].

The key elements of El Greco's arrangement (Figure 10) that hold the most significant importance are identified as follows [145]

(i) The GpioZero library [22] was utilised to create low-level functions that facilitate the connection between the hardware and the high-level functions of the code. High-level functions facilitate the mobility of El Greco.

(ii) The functionality of image recognition relies on the OpenCV library [151].

(iii) El Greco possesses the capability of speech recognition through the utilisation of the Google Speech Recognition API [152] with the assistance of the Python speech recognition library [153].

(iv) Additionally, El Greco exhibits proficiency in multiple languages. This is achieved by utilising the Google Text-to-Speech API [154] via gTTS [155] and Google Translate [156].

(v) The robot can also access and retrieve useful data from the Internet. The extraction of this data is achieved by utilising Lxml [157] and Feedparser [158], which are employed for parsing Atom and RSS feeds.

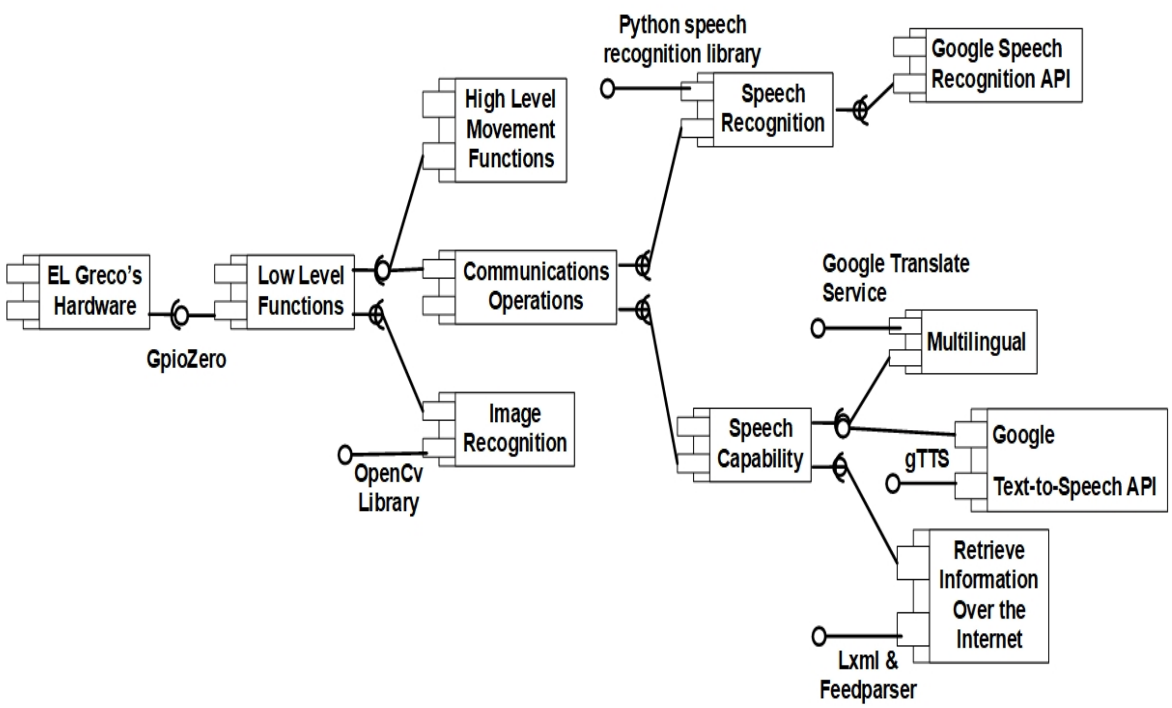


Figure 10: El Greco component diagram.

6.4 The El Greco Platform

6.4.1 The El Greco Platform Technical Set Up

Prospective learners can use the El Greco Platform to acquire proficiency in the Python programming language through the remote operation of the humanoid robot El Greco, which is facilitated by utilising the Blockly library or directly inputting Python code. The platform consists of four essential components: the server, the Website, the Robot El Greco, and the playroom.

The primary configuration of the server includes XAMPP [159], Nimble Streamer [160], and OBS Studio [161]. XAMPP is a complete software package that serves as a solution stack for web server functionality. Apache server, a part of the software package XAMPP, was chosen as the primary server for the Website, while MySql serves as the database for the El Greco platform. The database encompasses several components, including log-in credentials, user profiles, and a detailed account of El Greco's adventure necessary files, which will be further discussed in subsequent sections of this dissertation. The Platform utilises live-stream content that is recorded, encoded, and transmitted through OBS Studio in SLDP [162] format, a streaming protocol that relies on web sockets. This content includes auditory and visual elements and is delivered on the Website using Nimble Streamer, which functions as a supplementary server when required. The live stream is played using the HTML5 SLDP Player [163]. PHPMailer [164] automates Website email requirements. Phpseclib [165] establishes Secure Shell (SSH) connections between the Website, the server, and El Greco as part of the server configuration. WiFi technology facilitates connecting El Greco and the server (Figure 11).

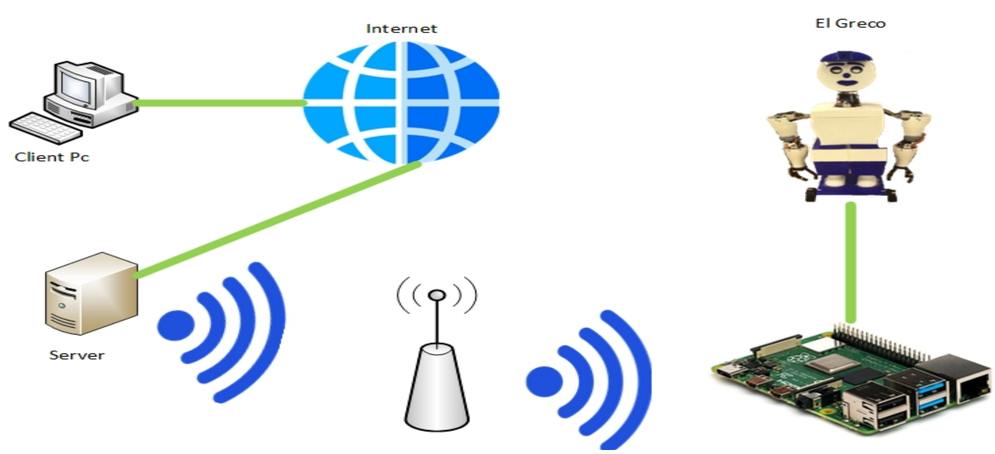


Figure 11: View of the Hardware employed for the needs of the El Greco platform.

The Website has been constructed utilising a combination of HTML, PHP, CSS, Javascript, Python, and Codemirror [166], a JavaScript-based text editor specifically designed for browser usage. In addition, Google's Javascript library Blockly [92] and Brython, a Python-to-Javascript compiler, were used. Brython was developed to substitute Javascript with Python as the scripting language for web browsers. The majority of the components and their corresponding dependencies of the El Greco platform are illustrated in Figure 12.

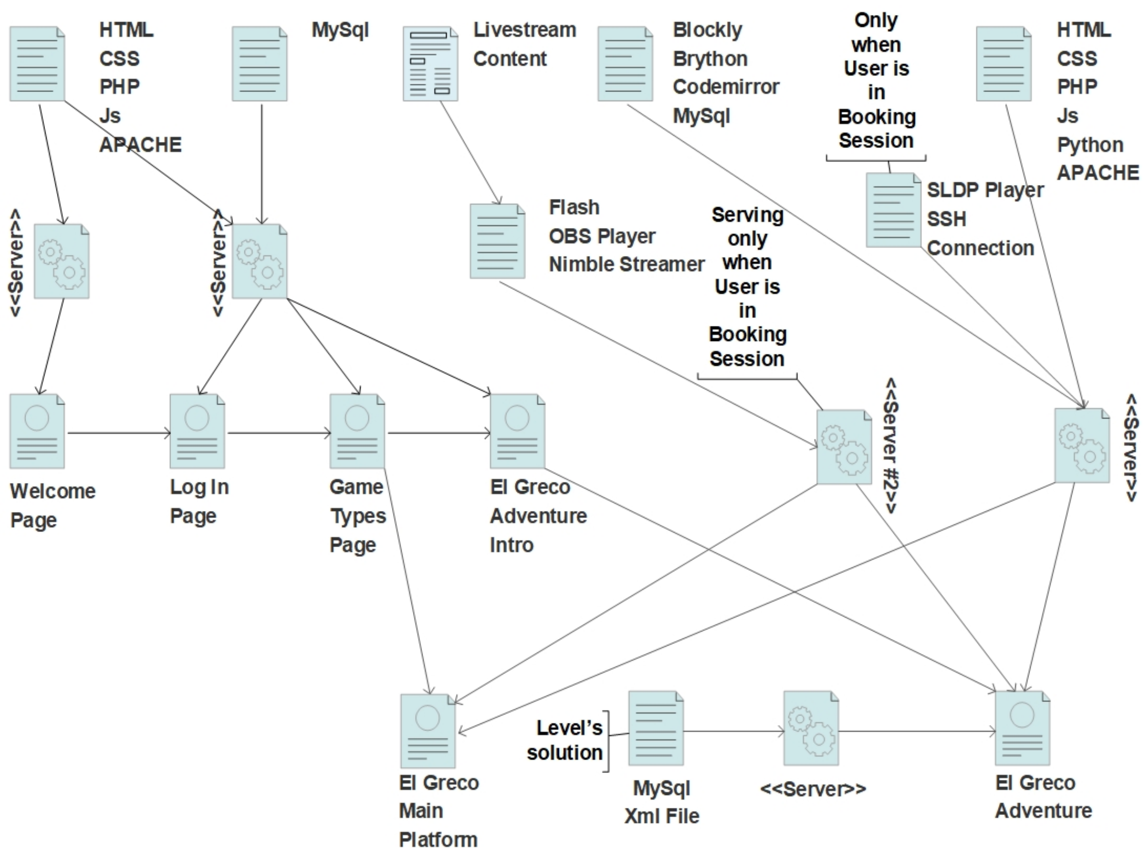


Figure 12: El Greco Platform component diagram.

The Playroom is a designated area intended to safeguard the robot from potential falls or collisions with the walls in the event of erroneous commands. The playground has a diameter of 1.60 metres and is characterized by a flat surface free of obstructions that may impede the robot's manoeuvrability. The Playroom and the robot are located in the Robotics Lab at Aegean University. The user can access the Playroom through live streaming (Figure 13).

Figure 13: Live streaming from El Greco playroom.

6.4.2 Website's main features

The following section will provide an overview and analysis of the primary functionalities of the Website, as illustrated in Figure 14. The initial webpage displayed to the user provides a range of options, including logging in, registering, recovering forgotten passwords, communicating with the site administrators, and accessing information about El Greco and the robotics team affiliated with the University of the Aegean. As part of the registration procedure, individuals are required to generate a profile. The mandatory data for establishing a profile includes a username, age, gender, and educational level. Users can provide their name, surname, photo, and a brief statement concerning why they like to use robots.

This data is collected to obtain anonymized statistical information regarding the Website's users to enhance its performance. Additionally, an automatic log file system was developed to document data related to user engagements with the platform. The existing body of literature indicates that in self-regulated e-learning settings, this information can be utilised to monitor pedagogical indicators, including the user's performance and motivation [24], [167]. By employing Web mining methodologies, it is possible to gather significant data related to the user's real-time performance, preferences, and behaviours.

This information can be utilised to enhance the learning process [168]. According to relative research [169], log files have been identified as a more appropriate and effective method for evaluating future user performance in an e-learning setting compared to data obtained from online questionnaires. Several factors are observed in El Greco Adventure, including the success rate, the time required to complete a level, the use of the tutorial, and the number of clicks the user performs during their attempt to finish a level. The combination of this information and a user's profile will serve as a helpful assessment of pedagogical indicators and the level of engagement. This assessment aims to evaluate the necessary improvements required for the Website to enhance its efficiency and integration within the educational process and overall learning experience.

An integral component of the registration procedure involves implementing an automated email response system to communicate with the user. The response includes a hyperlink leading to the profile creation page, where the password is encoded using the MD5 algorithm. The provided hyperlink possesses a single-use functionality.

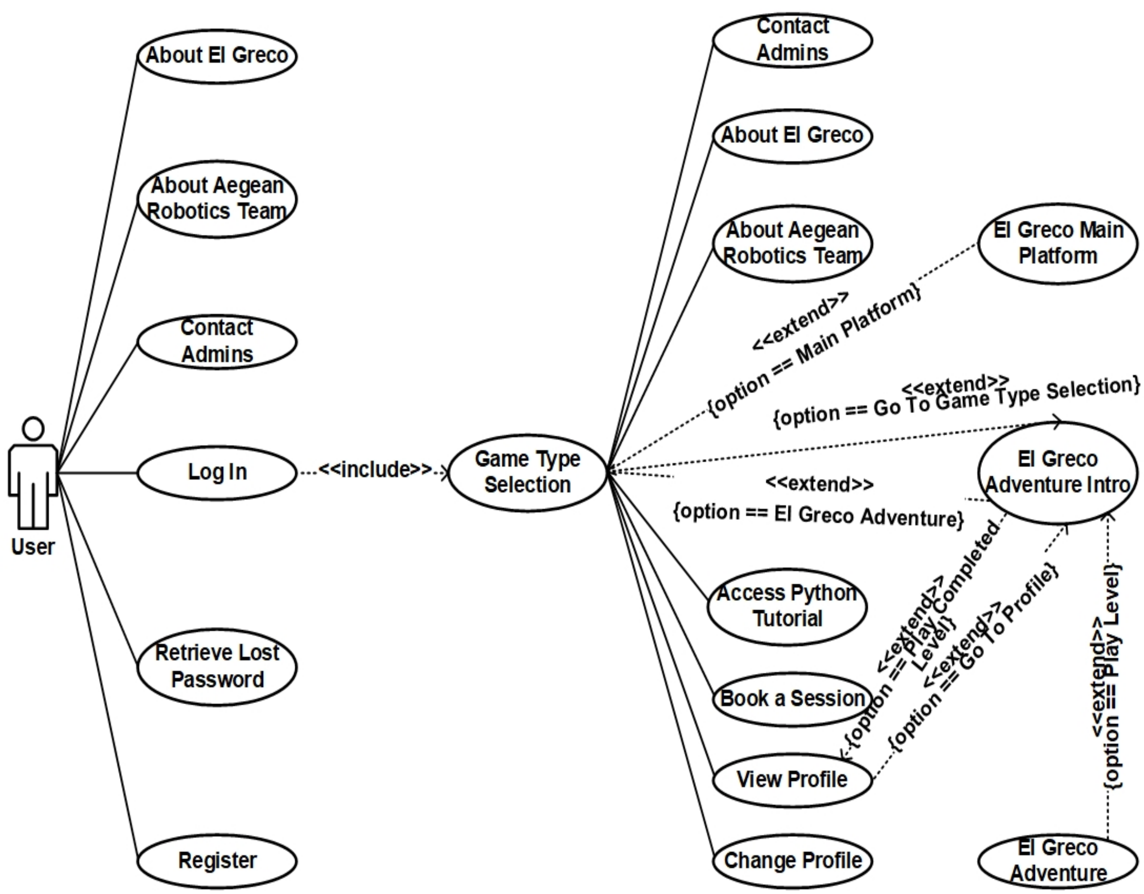


Figure 14: El Greco website use case diagram

Upon completing a successful registration process, the user can access the website by logging in. The primary webpage of the Website is the "Game Types" page, which features the El Greco Main Platform and El Greco Adventure options. These options will be further examined and analyzed in subsequent sections of this dissertation. This web page and its associated website pages offer various options accessible through the top menu bar. These options include information about El Greco, the university's robotics team, a Python tutorial, a contact form, and a user profile. The user profile allows individuals to modify optional information fields, the “level of education field”, and review completed levels of the El Greco adventure.

Upon logging in to the Website, users can access the booking service through the top menu on the Website. In order to remotely access and operate the robot inside its playground, the user must schedule a session. Due to the lack of total accuracy in the movements of El Greco, it is considered essential to have a supervisor present during sessions. Therefore, it was necessary to restrict the hours and dates available for booking. The standard session duration is set at thirty minutes; however, users can request additional time for the session using a comment area provided in the booking form. The presence of a booking session results in distinct modifications to the Website's interface to facilitate the booking process. One notable feature is the presence of a countdown timer positioned in the upper section of the webpage. This timer functions by counting down to the beginning of a session, provided that a session has been scheduled for the current day. Additionally, the timer displays the remaining session duration during an ongoing session. Furthermore, admission to the El Greco Playroom is only granted to users during specified session times. Consequently, the El Greco website can effectively oversee the environment and allocate resources to foster and enhance creativity.

A remote management application (Figure 15) was developed to manage the Website, focusing on the booking service. It should be acknowledged that fail-safes were implemented in order to mitigate the occurrence of double bookings. Regarding administrators, these safety measures are applicable at the beginning and during a session. However, when a user submits a session request, just the start time and date of the session are compared to the start time and duration of all sessions already scheduled for the requested date.

Part of the registration process is the automated email response to the user. A link with an md5-coded password to the profile creation page is included in the response. This link can be used only once. In this way, and with the limitation of one user registration per valid email account and the prohibition of using the same username, we can assure the safety of our users and the diversity and quality of the data gathered for the statistical analysis of the pedagogical performance of the platform.

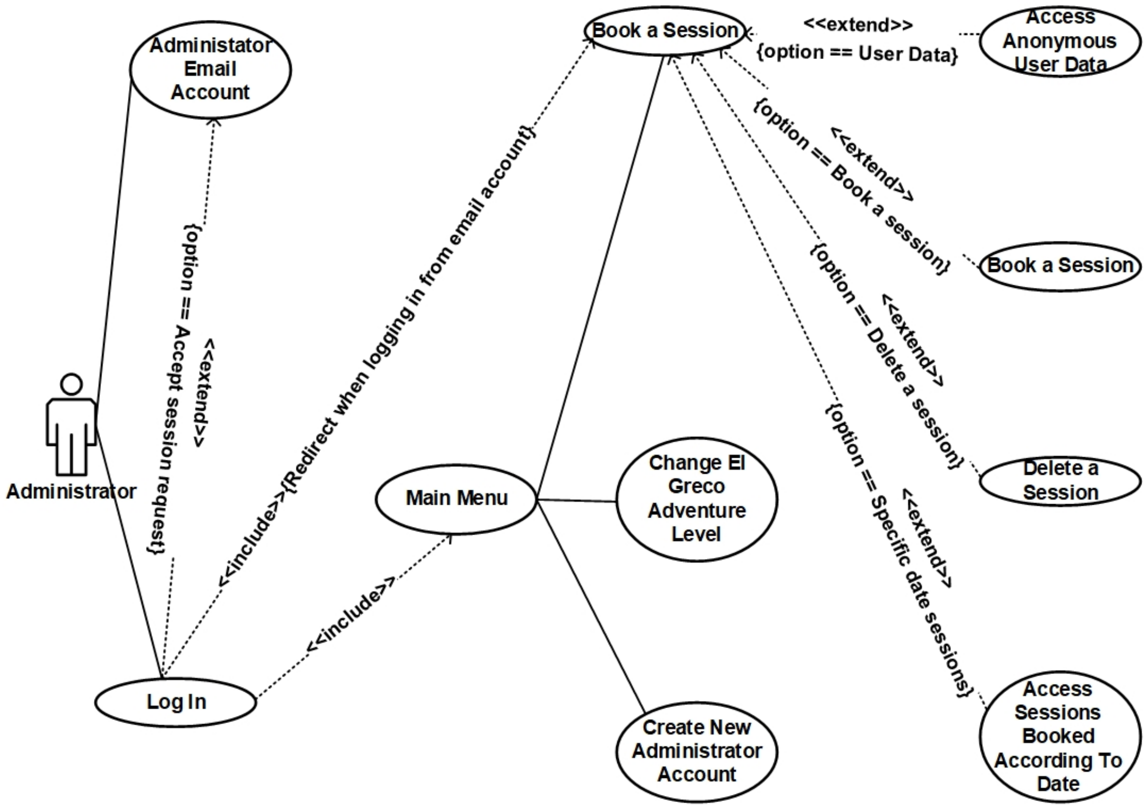


Figure 15: Administrator's application use case diagram.

6.4.3 Game Types

The El Greco Platform includes two distinct game categories: the El Greco main platform and the El Greco Adventure. Both platforms utilise Google's block programming framework Blockly, enabling users to employ a drag-and-drop interface for constructing programs that can subsequently be translated into several programming languages, including JavaScript, Python, PHP, Lua, and Dart (Figure 16). Currently, the El Greco Platform exclusively employs Blockly for Python. Blockly includes a set of pre-existing blocks but allows developers to create their own blocks. This option was employed to create the El Greco functions utilised to remotely control the Robot El Greco (Figure 17). Currently, the El Greco functions are as follows:

- Demo: El Greco demonstrates its capabilities.
- Salute: The user is greeted by El Greco.
- Dance: El Greco manoeuvres like it is dancing.
- Wait for: El Greco waits for a given number of seconds.
- Walk for: El Greco engages in the forward movement for a given number of seconds.
- Turn right: El Greco turns right for a given number of seconds.
- Turn left: El Greco turns left for a given number of seconds.

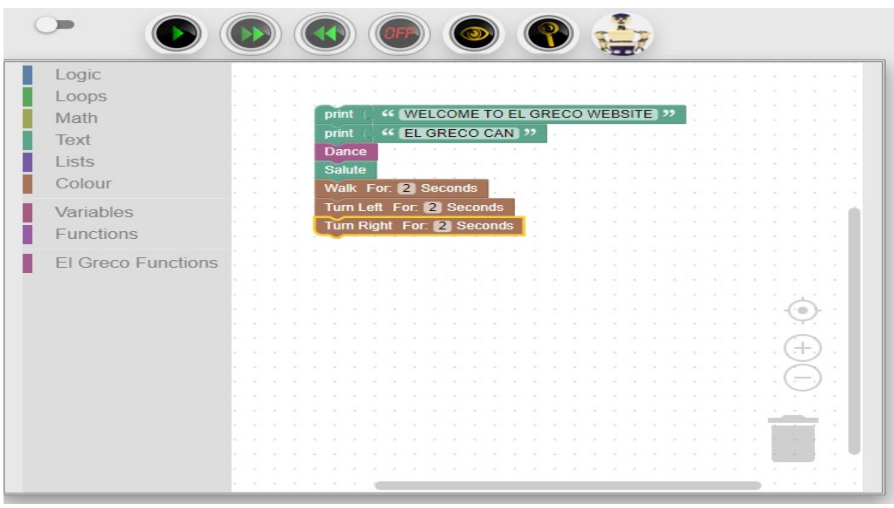


Figure 16: The Blockly interface.

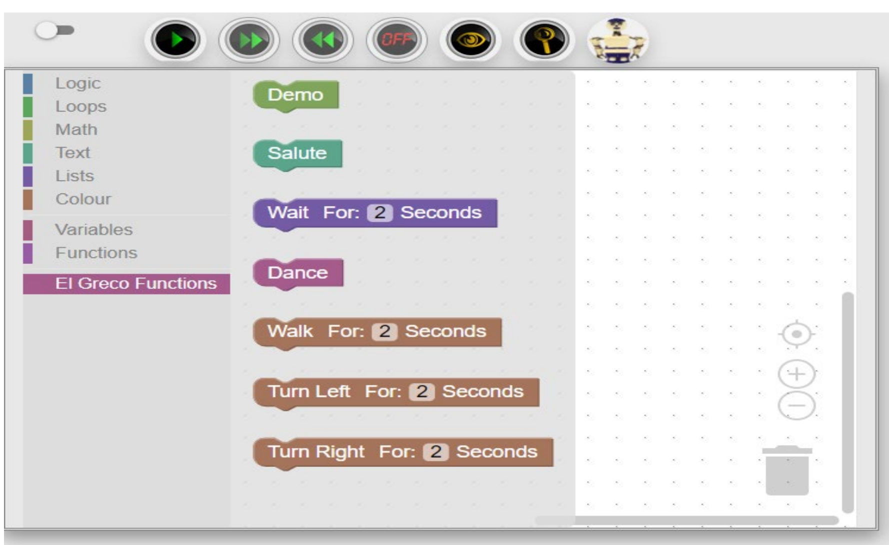


Figure 17: The El Greco functions.

The Website can accept programs developed using the Blockly visual programming language or by the direct input of Python code using the keyboard. The execution of the code differs depending on the program's method of creation and whether the user is in a booking session.

Blockly provides users with two options for executing code. One approach involves utilising the try-catch commands, while another approach involves employing the JavaScript interpreter built by Neil Fraser [170]. According to the information provided by Google, the second option effectively executes the code. This is the reason why it was chosen for implementation on the Website when Blockly is utilised for code integration. Blocky actively safeguards the process of code creation by actively preventing compile-time errors. The webpage experienced crashes due to runtime issues caused by unhandled exceptions thrown by the JavaScript interpreter in response to such failures. Within the El Greco Platform, these exceptions are identified and utilised to generate guided feedback for the user, aiding them in rectifying errors present in the code. A Blockly trap was created using the recommended method outlined on the official Blockly website [171] for dealing with infinite loops. In the event that a line of code is executed in excess of 1000 iterations, the execution process is terminated, and a suitable notice is presented to the user.

The manual insertion of code involves the utilisation of Brython, a Python-to-Javascript compiler enabling the execution of Python code within a web browser without necessitating the use of plugins or server-side support. The choice of this strategy was motivated by its numerous advantages. The primary advantage is in the enhanced speed of code execution, as the round trip to the server would significantly reduce this efficiency. Brython does code inspection to identify faults in debug mode, presenting them precisely. The errors mentioned above are collected and transcribed into appropriate messages that are subsequently presented to the user to correct the code. When utilising Brython for execution, a straightforward mechanism is employed to handle infinite loops by limiting the number of lines of code-generated output. If that limit is passed, the execution is halted, preventing the Website's imminent crash that would otherwise occur when endless loops are conducted.

One crucial aspect of the Website's feedback feature is its ability to execute the code in a step-by-step manner (Figure 18). Additionally, it offers the functionality to navigate through the code both forwards and backwards. Accordingly, it can be assumed that the El Greco Platform can offer valuable feedback to users, thereby enhancing the efficacy of the educational process. The design of the step-by-step execution mode and the incorporation of an infinite loop trap in the manual Python insertion process were influenced by the website Reeborg's World, created by Andre Roberge. Modifications were made to the highlight.py module in Reeborg's World for Python execution and the highlight block function in Neil Frazer's Js interpreter for Blockly execution to accommodate the Platform's requirements.

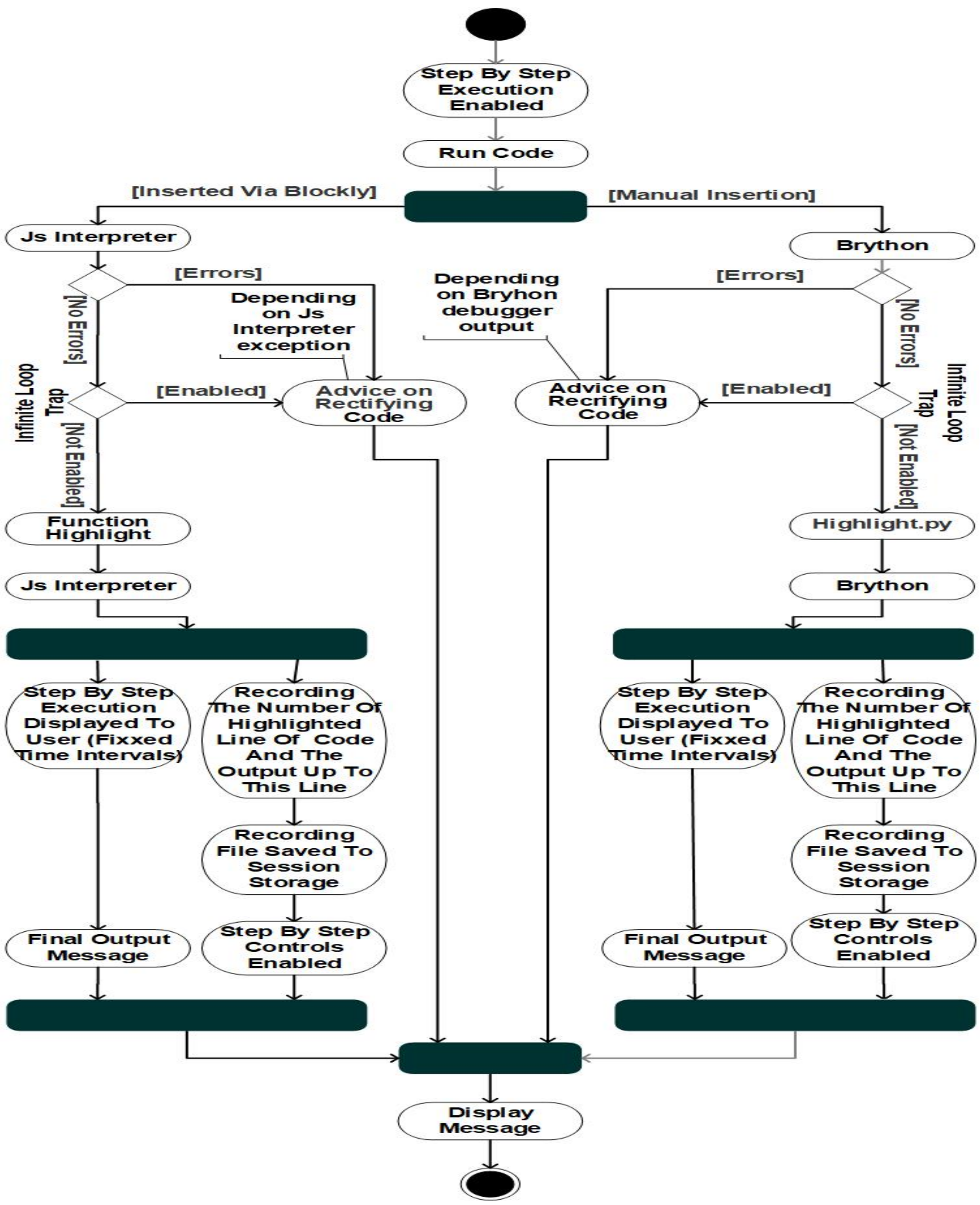


Figure 18: Step by Step execution activity diagram.

It is important to note that outside of a booking session, the El Greco functions are automatically declared and injected into the user's code as Python print commands. For instance, the El Greco dance function outputs the message "El Greco Dance!" During a session, the code written by the user can be submitted for execution to EL Greco after being examined for flaws and the inclusion of El Greco functions. A Python script is generated to serve as the program if an El Greco function is present. Subsequently, an SSH connection is established with the assistance of the Phpsec library, enabling El Greco to receive and run the aforementioned script. In the event that the generated code lacks an El Greco function, the user will be notified appropriately, and no further action will be taken.

6.4.4 Main Platform

El Greco main platform is a web page, which contains three primary sections. The interface consists of two main components: the Blockly and code areas (Figure 19). The Blockly area allows users to generate code by utilising Blockly blocks. On the other hand, the code area is a text area implemented with Codemirror, enabling users to input Python code either through keyboard input or by utilising a button that converts the code from Blockly blocks into Python code. The third and final section of the webpage displays live-streaming video footage captured from El Greco's playground, as depicted in Figure 13.

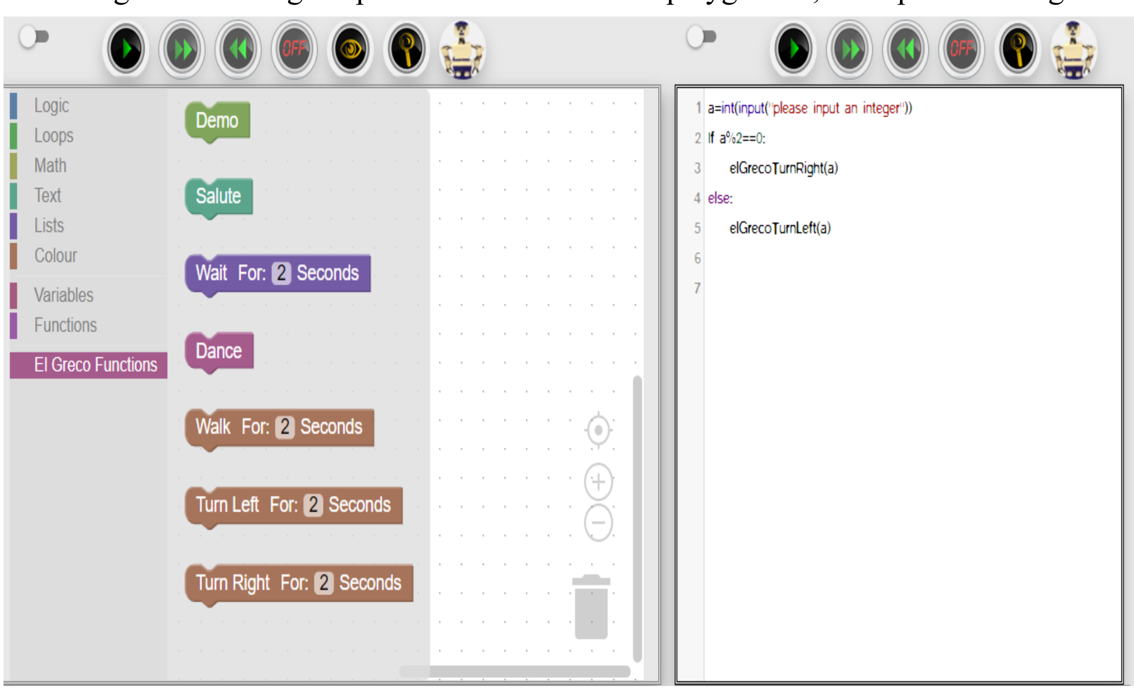


Figure 19: Blockly area and the Code area.

The visibility of this window is restricted to instances where the user is engaged in a booking session and can remotely manipulate the actions of El Greco.

Additionally, the interface provides a range of buttons that allow the user to perform several actions. These actions include executing code from the code area or Blockly blocks, converting Blockly blocks code into Python code, executing code generated using either aforementioned method utilising El Greco, and stepping through the code as it is being executed. The El Greco Main Platform use case diagram is shown in Figure 20.

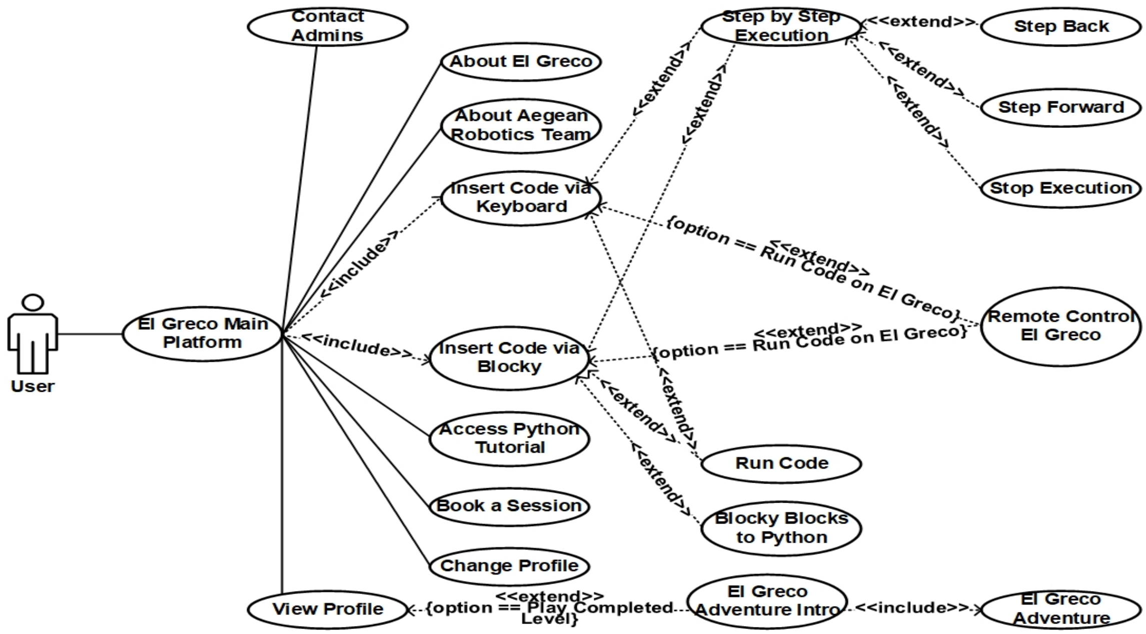


Figure 20: El Greco Main Platform use case diagram.

6.4.5 El Greco Adventure

The El Greco Adventure is an interactive game that challenges users to solve particular Python programming tasks that include the remote-control capabilities of the El Greco robot. The game consists of a total of eight levels. Each level is centered around a primary subject, such as using variables. The user is given objectives related to the main topic and is expected to carry them out; for instance, they may be requested to generate and compare two variables. Based on the analysis results, El Greco moves differently each time.

Moreover, the El Greco Adventure levels are designed in a flexible manner that minimizes the imposition of constraints. In an El Greco Adventure level, for example, the user is required to direct El Greco along a square route. This task can be accomplished by employing the El Greco function of executing a series of four consecutive turns, either to the right or left, for an equal duration of time. In this way, the Platform demonstrates the capacity to effectively engage learners by providing challenging opportunities that are not constrained by limits.

Ultimately, as demonstrated in the previous example, the design of the El Greco Adventure levels is intended to foster the expansion of one's understanding and proficiency in the fields of mathematics and geometry. Therefore, the Platform offers learners the chance to acquire knowledge from diverse fields of expertise. The use case diagram depicting the El Greco adventures mode is presented in Figure 21.

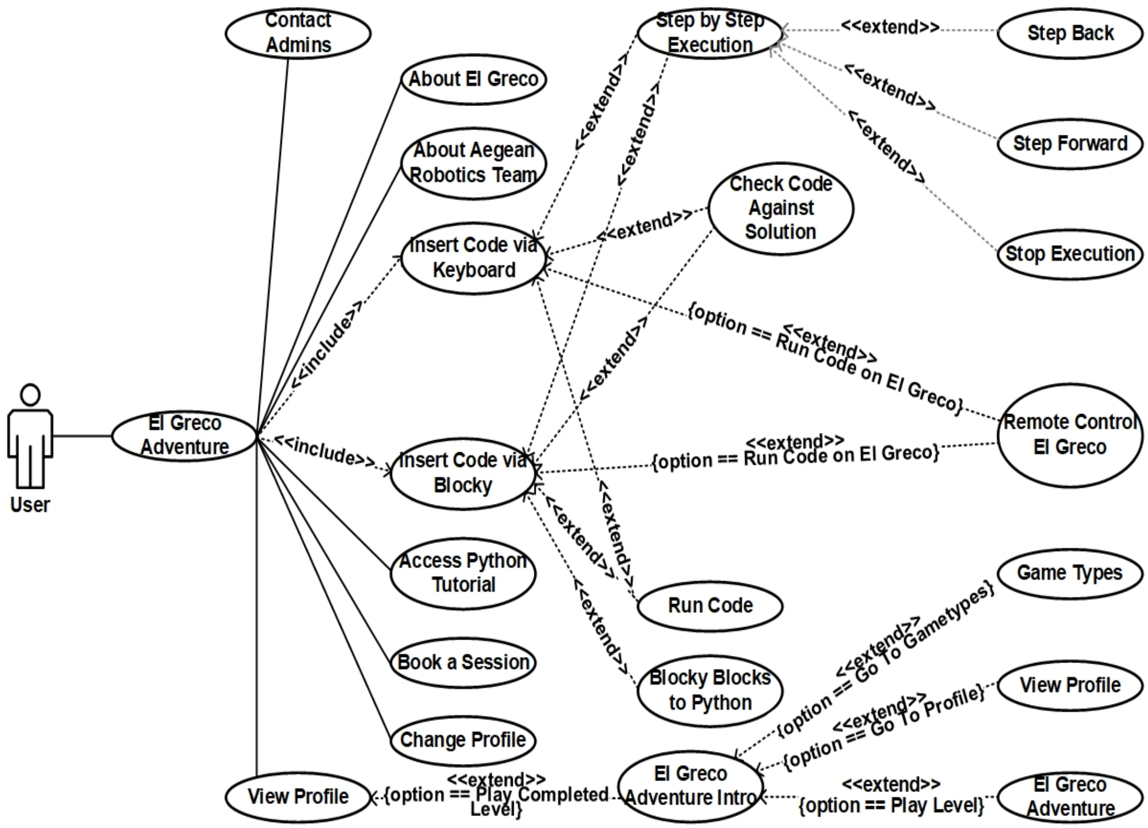


Figure 21: The El Greco Adventure use case diagram..

The term "toolbox" refers to the set of blocks that are accessible through the Blockly platform and can be customized to suit specific needs. While El Greco's main platform utilises the basic toolbox comprising of all the available Blockly blocks, El Greco Adventure employs configurable toolboxes tailored to the skill level needed to complete the level. By actively managing the resources involved in the educational process, newcomers can significantly benefit. When users choose either the El Greco Adventure option from the game types page or utilise the option provided on their Profile page to replay a previously completed level, they are subsequently sent to an initial animated scene (Figure 22). In this scene, El Greco is shown moving over the game map. Upon El Greco's arrival at the specified level, the user is offered a description outlining the level's objective. Subsequently, the user can either commence playing or navigate back to the game types or profile page. Upon completing all levels in El Greco Adventure, the introductory interface undergoes a modification, granting the player the ability to manipulate El Greco's movements through keyboard controls. Consequently, the user gains the freedom to select any desired level by utilising this newly acquired functionality.

Figure 22: El Greco Adventure introduction screen.

The "El Greco Adventure" webpage is accessed following the introductory screen. The design closely resembles the main platform of El Greco, with the inclusion of two additional buttons. The purpose of these buttons is to verify the accuracy of the code input by the user and determine its ability to effectively fulfil the required task to progress to the next level.

The code entered by the user undergoes a two-stage inspection process. During the initial phase, the code undergoes scrutiny based on the outcomes produced by the code using JavaScript pattern matching.

Following the completion of the initial phase of code evaluation performed by the Platform, a subsequent stage is undertaken, involving the comparison of syntax. Inspecting syntax errors at compile-time occurs earlier, as elaborated in section 6.2.3 of this dissertation. The syntactical comparison process involves comparing code against a regular expression representing the expected solution at a given level. This comparison is facilitated through the utilisation of JavaScript pattern matching. However, it is essential to note that, at this point, the comparison is performed on the code itself rather than its output. The website administrator and the requirements of the given level determine the amount of strictness or tolerance in this inspection.

Upon the completion of the two-stage code inspection process, the user is promptly notified through the utilisation of a pop-up window. The potential results of the inspection can be categorized as follows: (i) accurate, (ii) incorrect in terms of the output, (iii) incorrect solely in terms of the syntactical inspection, or (iv) incorrect in both aspects. This practice is implemented because many programs may yield identical outputs although utilising distinct or additional commands [172]. The administrator-designer of the level determines the degree of strictness or tolerance in the comparison. As a result, the El Greco website offers feedback and acknowledgement to foster motivation towards learning. Figure 23 displays the activity diagram illustrating the check against the solution process.

The El Greco platform, notably the El Greco Adventures mode, can be seen in a game-based learning and gamification scope. The essential idea underpinning gamification is that games' naturally enjoyable qualities can be harnessed in the educational process [173], whereas game-based learning requires actual games as pedagogical tools [122]. Gamification is distinguished from game-based learning by the fact that gamification does not involve the use of actual games as pedagogical instruments. Game-based learning and gamification approaches seek to excite and engage students in the learning process [112]. There is evidence that incorporating aspects of game design into engineering education can boost student motivation and encourage successful completion of classes [174]. Implementing this strategy can potentially increase the amount of satisfaction and active involvement in the learning process, ultimately resulting in a more robust understanding of the knowledge being taught. The field of engineering studies makes substantial use of gamification and game-based learning paradigms [175].

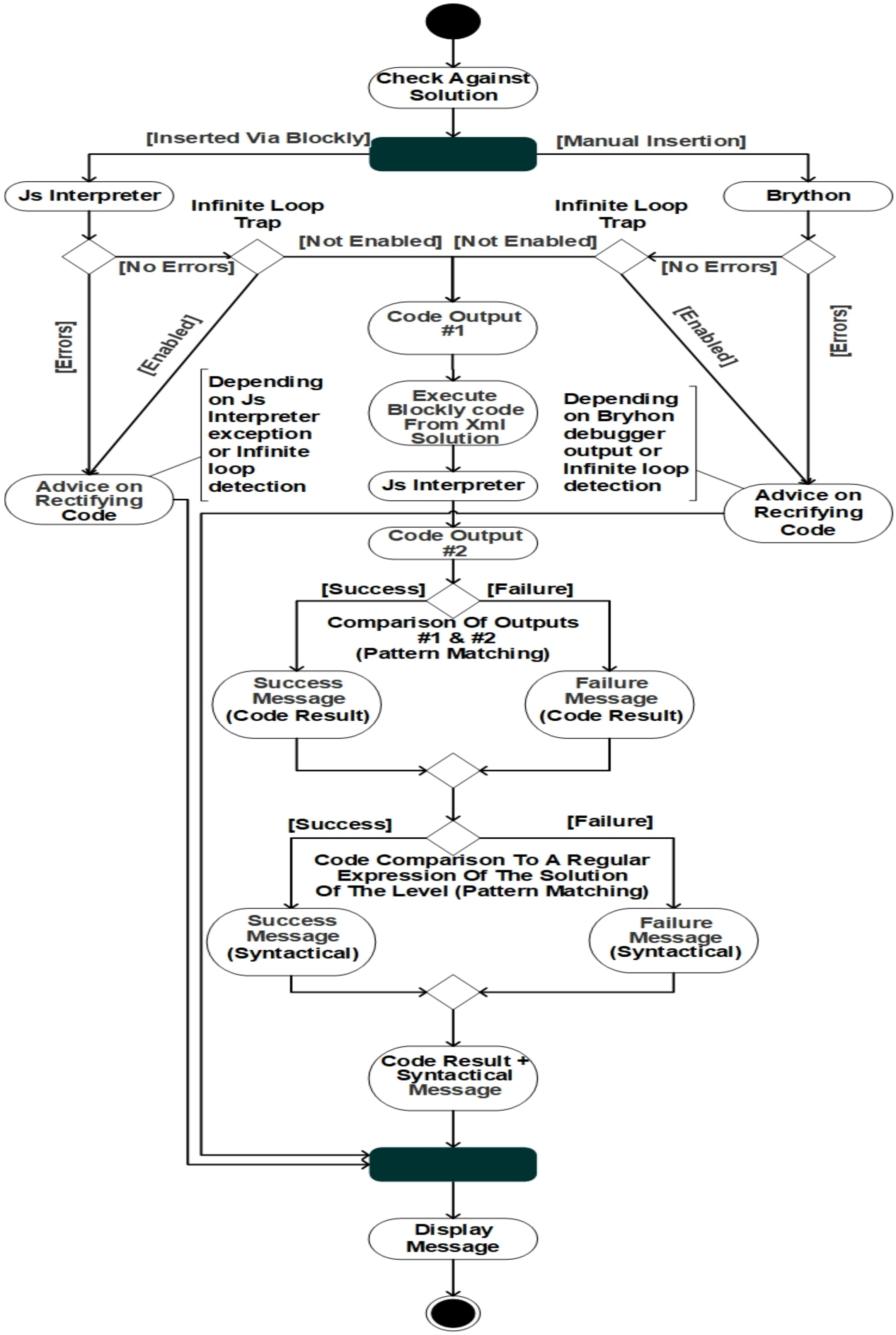


Figure 23: Check against solution activity diagram.

6.5 The El Greco survey

6.5.1 Study Design

The survey was carried out utilising two questionnaires. Prior to utilising the platform, the participants completed the first questionnaire. The initial section of the questionnaire utilised the Computer Attitude Questionnaire V5.22 [176], specifically focusing on three of its subscales. The first subscale, F1, assessed the participants' perception of the importance of computers. The second subscale, F2, measured the enjoyment the sample experienced when using computers. Lastly, the third subscale, F7, gauged the extent of anxiety induced by computer usage. The second section of the initial questionnaire incorporated inquiries derived from the SCAPA questionnaire [177] to assess the participants' attitudes and beliefs about programming. The third section of the first questionnaire evaluated the participants' attitudes towards robotics. This section included items derived from the 4-H Robotics and GPS/GIS Interest Questionnaire [178]. The final section of the initial questionnaire was designed to evaluate the participants' proficiency in Python and Blockly. Four Likert-type questions about code-related matters were utilised to accomplish this purpose. Two were relevant to the Blockly environment, while the remaining two applied to written Python code.

The second questionnaire was designed to evaluate the impact of the El Greco website on participants' views towards programming and robotics. It incorporated several questions from the initial questionnaire, and an additional section specifically focused on the Platform. The participants completed the questionnaire after they utilised the Platform. Before utilising the Website, users were given a brief tutorial on Blockly and the Python programming language. The individuals engaged in collaborative interactions with the Platform in groups of two. The decision to implement this technique is based on the benefits that are linked to the practice of small-group teaching [179]. Amongst them is better self-directed learning, a crucial part of the online learning method [180].

It must be noted that the findings of every Likert-type question were processed in such a way that a higher score indicated a more favourable outcome. The processes involved in the survey's design are depicted in Figure 24.

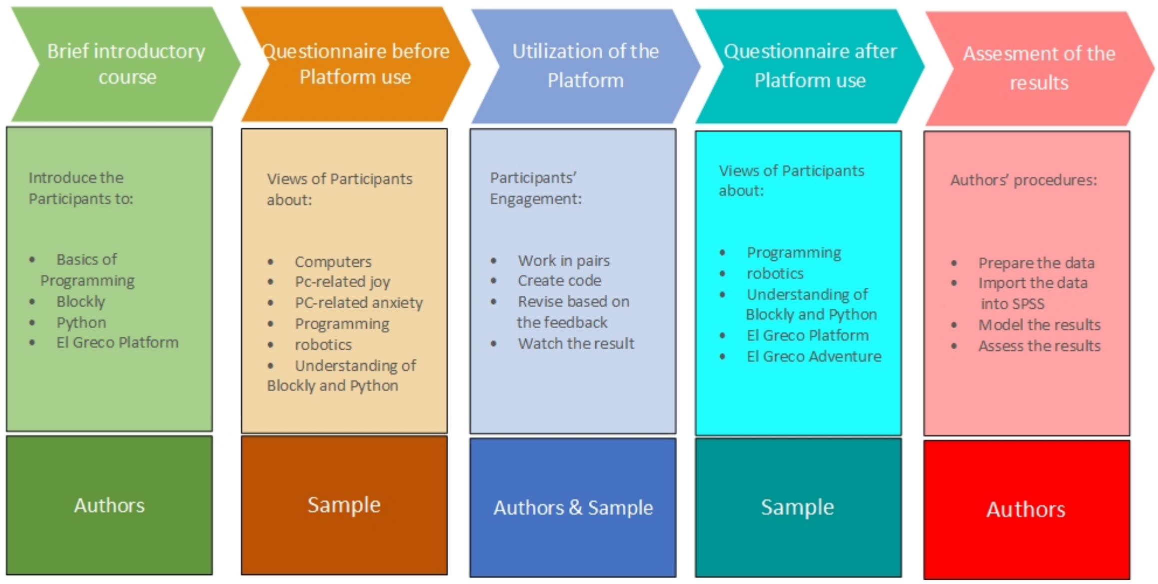


Figure 24: Processes involved in the El Greco platform survey's design.

6.5.2 Participants and Preliminary Findings

The survey was conducted among a total of 116 lyceum and gymnasium students in Greece. There were 52 male individuals and 64 female individuals within the age range of 13 to 18. The age and grade distribution of the participants are depicted in Figures 25 and 26, respectively.

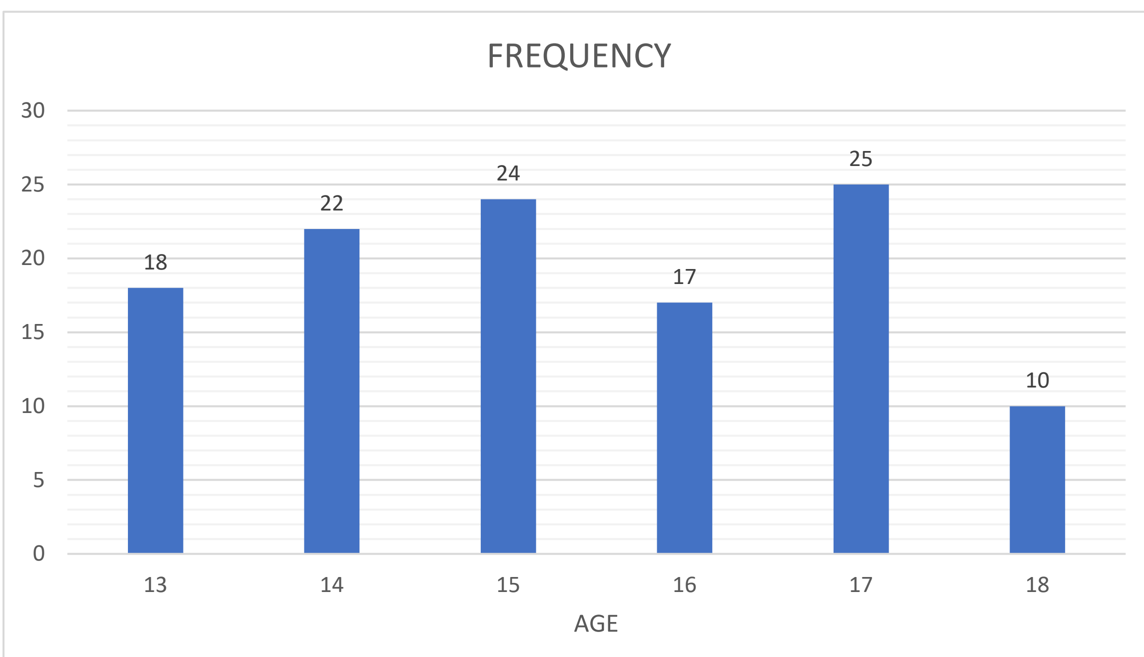


Figure 25: Age distribution of the sample.

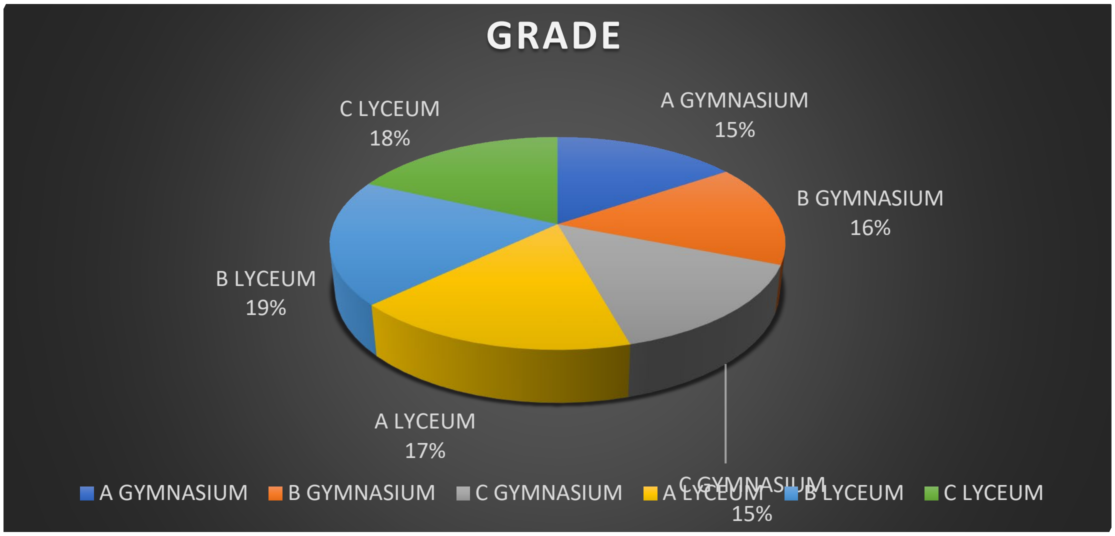


Figure 26: Grade distribution of the sample

The participants reported utilising a computer for a duration ranging from two to fifteen years, with an average duration of 5.92 years. The weekly use of computers ranged from 0 to 25 hours, averaging 11.68 hours. Based on these data, it can be inferred that the sample population utilises computers in their daily activities, regardless of the specific type of computer.

The participants' perception of the importance and enjoyment of computer usage was assessed using the F1, F2, and F7 subscales of the Computer Attitude Questionnaire V5.22. The data collected from the sample indicates that the average score for computer importance was 3.7931, the average score for computer enjoyment was 3.8549, and the average score for computer anxiety was 2.3017. The comprehensive score is presented in Table 1. Based on these data, it can be inferred that the participants hold the belief that computers play a significant role in their day-to-day activities. They derive pleasure from utilising them without feeling anxious.

Table 1:Descriptive statistics of the F1, F2 and F7 subscale of the Computer Attitude Questionnaire V5.22 for the El Greco platform survey.

	N	Min	Max	Mean	Std. Deviation
CAQF1	116	2.00	5.00	3.7931	0.80506
CAQF2	116	2.17	5.00	3.8549	0.62188
CAQF7	116	1.00	5.00	2.3017	0.89898

In order to assess the participants' attitudes towards programming, Likert-type questions were employed. The inquiries were derived from the SCAPA questionnaire and its corresponding subscales. The obtained mean score of 3.1401 in self-reported programming understanding supports the hypothesis that the population possesses limited familiarity with fundamental programming principles. Additionally, the data analysis revealed a mean value of 3.2155 for the beliefs regarding programming costs. This indicates that the sampled individuals perceive that programming does not entail significant cost. Moreover, the average score of 3.1293 on the programming intrinsic value belief scale suggests that the individuals do not exhibit strong positive or negative attitudes towards programming. The mean of 3.1466 in programming attainment value belief also led us to this conclusion.

However, it can be concluded from the data that the sample tends to be persistent in programming activities and leans to believe in the utility of programming, as evidenced by the slightly higher mean score in programming utility value belief (3.4368) and programming persistence (3.3405). These assumptions are further supported by the standard deviation observed in each subscale. Table 2 displays the detailed descriptive statistics relating to the subscales.

Table 2: Descriptive statistics of the SCARPA subscales used in the first questionnaire of the El Greco platform survey.

	N	Min	Max	Mean	Std. Deviation
Self-reported programming understanding	116	2.00	5.00	3.1401	0.60008
Programming cost belief	116	1.33	5.00	3.2155	0.72266
Programming intrinsic value belief	116	2.00	5.00	3.1293	0.72867
Programming attainment value belief	116	2.00	5.00	3.1466	0.76048
Programming utility value belief	116	2.00	5.00	3.4368	0.73084
Programming persistence	116	1.50	5.00	3.3405	0.84339

The first questionnaire also included two open-ended questions regarding the respondents' attitudes towards robotics. Based on the responses provided by the participants, it can be assumed that the sample population has not extensively engaged in the field of robotics. Nevertheless, a widely accepted belief among many participants is that robotics requires complex programming and construction procedures. Moreover, the findings indicate that the participants are intrigued by robots and their utilisation.

The subsequent portion of the initial questionnaire was designed to further explore the participants' attitudes towards robotics, drawing inspiration from the 4-H Robotics and GPS/GIS Interest Questionnaire. The average score in this questionnaire section was 3.1379, with a standard deviation of 0.54327. Based on the information at hand, it can be inferred that the surveyed individuals possess a restricted level of familiarity with robotics. However, it is noteworthy that the responses to the query "I like learning new technologies like Robotics" exhibited a mean score of 4.02, accompanied by a standard deviation of 0.844. This discovery suggests that the participants are interested in robots.

The following section of the first questionnaire assessed Blockly and Python expertise. Table 3 displays the findings. The results support the assumption that most participants are unfamiliar with Blockly or Python.

Table 3: Descriptive statistics of former experience in Blockly and Python of the El Greco platform survey.

	N	Min	Max	Mean	Std. Deviation
I have heard of Python Programming language	116	2	5	2.37	0.717
I can program in Python	116	2	4	2.33	0.615
I have heard of Blockly	116	2	5	2.30	0.675
I can program in Blockly	116	2	5	2.29	0.660

The assumption was further strengthened by the four control programming questions in the concluding section of the initial questionnaire. The sample had to interpret the results of the execution of four short and relatively basic programs. Two questions were implemented using the Blockly environment, while the remaining two were implemented through Python code. The subsequent section of this dissertation will present the responses provided by participants and their corresponding percentages compared to the results obtained after utilising the El Greco platform.

The initial questionnaire analysis indicated that most participants had a satisfactory level of computer proficiency; however, their exposure and proficiency in robotics and programming were rather limited. These aforementioned attributes render them suitable candidates for assessing the pedagogical advantages of the El Greco platform concerning Python learning and the level of joy experienced throughout the learning journey.

6.5.3 analysis of the second questionnaire.

A brief introduction to the platform and the Python programming language was made prior to the usage of the platform. Following that, the second questionnaire was answered. It was conceived in such a manner as to investigate the efficacy of the El Greco platform in the context of the learning procedure. The participants worked in pairs to use the Platform for approximately half an hour. The limitations of the platform, which were stated before, along with the size of the sample, prevented more extended testing periods.

This questionnaire repeated a few of the questions of the first questionnaire. In order to study how the platform affected the beliefs of the participants, a paired-sample t-test was conducted. The outcomes are presented in Tables 4 and 5, respectively.

According to the data in the Tables, every result demonstrates a statistically significant and beneficial effect. It is reasonable to conclude that the participants' perceptions of programming and robotics were improved due to their interaction with the El Greco Platform.

Table 4: The dependent t-test analysis of the beliefs of the sample about programming after and before the use of the El Greco platform.

	Paired Differences						
				95% Confidence Interval of the Difference			
	Mean	Std. Deviation	Std. Error Mean	Lower	Upper	t	Sig. (2-tailed)
Programming is tiresome	0.828	1.015	0.094	0.641	1.014	8.779	0.000
Programming scares me	0.733	0.963	0.089	0.556	0.910	8.193	0.000
I like programming	0.603	0.745	0.069	0.467	0.740	8.729	0.000
It's important to me to be good in programming	0.336	0.734	0.068	0.201	0.471	4.936	0.000
I want to make an effort to be good in programming	0.233	0.817	0.076	0.083	0.383	3.069	0.003

Table 5: The dependent t-test analysis of the beliefs of the sample about robotics after and before the use of the El Greco platform.

	Paired Differences						
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference		t	Sig. (2-tailed)
				Lower	Upper		
I like learning about new technologies like Robotics	0.284	0.630	0.058	0.169	0.400	4.865	0.000
It's important to me to learn Robotics	0.405	0.659	0.061	0.284	0.526	6.621	0.000
It's important to me to learn the basic principles of Robotics	0.440	0.636	0.059	0.323	0.557	7.440	0.000
It's important to me to learn to program a robot	0.414	0.605	0.056	0.302	0.525	7.362	0.000
I am certain that i can program a robot to move forward and stop	1.052	0.811	0.075	0.903	1.201	13.962	0.000
I am certain that i can fix the software of a robot that malfunctions	0.578	0.635	0.059	0.461	0.694	9.804	0.000
I think i can solve practical issues using Robotics and programming	0.586	0.723	0.067	0.453	0.719	8.731	0.000
I think robotics and programming will help to lead a successful life	0.336	0.709	0.066	0.206	0.467	5.104	0.000
I want to find a job in robotics and programming	0.595	0.722	0.067	0.462	0.728	8.873	0.000

The subsequent section of the second questionnaire had Likert-scale questions related to the attitudes held by the participants about the El Greco Platform. The questions were written to establish a positive correlation between a higher score and a more robust level of belief in the platform being evaluated. The mean of the overall score was 3.9178, with a standard deviation of 0.55505. Based on the obtained results, it can be inferred that the El Greco platform had a user-friendly interface and positively influenced the participants' attitudes towards programming and robotics. One of the questions asked related to the influence performed by El Greco Adventures on the sample. The calculated mean for this question was 4.27, while the standard deviation was 0.762. These findings provide evidence that the El Greco platform may be evaluated from a gamification and a game-based learning standpoint. This observation strengthens the notion that the El Greco Platform benefited the level of enthusiasm and engagement in the educational process while enhancing the comprehension of the taught subjects.

This questionnaire section also included two open-ended questions regarding the participants' preferences or criticisms of the Platform. Further examination reveals that most participants expressed a favourable disposition towards the Platform, particularly for the Robot element. Additional frequently mentioned replies indicated that the participants positively perceived the Website's aesthetic appeal and found it convenient to operate the robot remotely. On the other hand, many individuals believed that there ought to be an increased variety of robotic functions and additional levels within the El Greco Adventure game. It is imperative to acknowledge that a notable proportion of the sample, precisely 25.2%, had no concerns or issues with the Platform.

Tables 6 to 9 present the percentages of responses to the four control questions regarding programming before and after utilising the El Greco Platform.

Table 6: The difference in percentages to the answers for the first Blockly program.

	Percent (Before)	Percent (After)	Difference
Correct	29.3	81.9	52.6
Wrong	0	2.6	2.6
Do not know	70.7	15.5	-55.2

Table 7: The difference in percentages to the answers for the second Blockly program.

	Percent (Before)	Percent (After)	Difference
Correct	9.5	75.9	66.4
Wrong	11.2	8.6	-2.6
Do not know	79.3	15.5	-63.8

Table 8: The difference in percentages to the answers for the first Python program.

	Percent (Before)	Percent (After)	Difference
Correct	1.7	25	23.3
Wrong	16.4	31.9	17.5
Do not know	81.9	43.1	-38.8

Table 9: The difference in percentages to the answers for the second Python program.

	Percent (Before)	Percent (After)	Difference
Correct	23.3	70.7	47.4
Wrong	0.9	0.9	0
Do not know	75.9	28.4	-47.5

A significant rise in correct responses, particularly concerning the Blockly code, is evident. However, there was also a significant increase in the correct answers for the Python questions, particularly for the second Python program. Moreover, prior to utilising the platform, most participants preferred responding with "I do not know" rather than attempting to provide a solution to the given question. The occurrence of this event had a considerable decrease following the utilisation of the platform. This observation is further illustrated by the Python program inquiries, which may be inferred to have posed greater difficulty for the participants in the study. Additionally, there has been an observable rise in incorrect responses to the first Python question and an increase in accurate responses. Based on this observation, it may be inferred that the participants, subsequent to utilising the platform, exhibited increased confidence in their programming abilities and decided to attempt to answer the question rather than responding with "I do not know."

6.6 Discussion

The El Greco Platform serves as a valuable educational resource for individuals of varying programming proficiency, encompassing young pupils and adults. Basic programming principles are integrated into the curriculum of nearly all modern engineering educational programs. This learning tool has the potential to be utilised in several educational settings, including distant or blended learning environments, as well as traditional classroom settings. The platform is a cost-free system that operates on a novel Python programming framework, enabling users to manipulate a robot through the Internet remotely. The platform offers feedback and acknowledgement through an error feedback system for programming and an interface that is user-friendly and interesting. The El Greco adventures present a stimulating challenge to users, fostering knowledge acquisition from several areas of expertise.

The El Greco platform, particularly the El Greco Adventures mode, can be observed within game-based learning and gamification contexts. A fact that may potentially augment the level of satisfaction and active involvement in the educational process, hence solidifying the comprehension and retention of the material being taught.

Based on the survey results, it can be concluded that the El Greco Platform generated significant interest among the study participants and positively influenced their attitudes towards programming and robotics. Moreover, it had a significant influence on their comprehension of programming and served as an inspiration for their pursuit of additional avenues to enhance their expertise in robotics and programming.

Chapter 7. The EDUV Platform

7.1 Introduction

Despite the presence of several types of vehicles, such as those designed for ground, aerial, and maritime use, the current educational underwater vehicles are constrained in their capabilities and serve as learning tools for mechanics, control systems, and hydrodynamics [181]–[183]. Students can acquire programming skills by utilising the EDUV Platform, which enables them to operate an underwater vehicle remotely. The EDUV platform has been developed to use the EDUV (Educational Underwater Vehicle) underwater vehicle [184]. EDUV is illustrated in Figure 27.

Figure 27: The EDUV underwater robot.

This educational platform offers two distinct approaches to learning programming: first, students can utilise the Blockly library to assist them in managing the underwater vehicle; alternatively, they can directly input Python code into a coding environment. In this way, the process of acquiring knowledge is rendered more captivating and enjoyable.

7.2 Related Work

Autonomous and remotely operated underwater vehicles (AUVs and ROVs, respectively) have been recognized as valuable tools in the field of education [185]. The study in [186] involved the development of a functional underwater glider with the primary purpose of serving as a platform for oceanographic research and educational activities. The glider known as Bumblebee is capable of being operated by a limited crew without the need of heavy equipment. Bumblebees possess autonomous emergency systems and communication mechanisms. The utilisation of this underwater vehicle holds potential for conducting AI experimentation and developing navigation algorithms, even in the absence of specialized hardware knowledge. In the context of education, the utilisation of a cost-effective and functional AUV is highly ideal [187]. The system comprises of two distinct types of AUV testbeds, a very accurate ultrasonic range system, a Long Baseline and Long-Range Frequency (LBLF) coordinate detection system, and a Two-Frequency Shift Keying (2FSK) ultrasound communication system. The involvement of undergraduate students in the construction of this experimental system resulted in personal satisfaction and academic advancement. Similarly, a team of students successfully developed an AUV platform named "Lucky fin" by effectively integrating theoretical knowledge gained in the classroom to their projects [188]. Students have the opportunity to construct and apply diverse control algorithms, as well as carry out experiments aimed at determining and analyzing hydrodynamic parameters of underwater vehicles. The platform is equipped with a pair of testing tanks and a single underwater vehicle. The control card, user control program interface, and manipulator's arm have been specifically built to facilitate depth perception, directional control, target tracking, and object capture.

The pedagogical approach employed in the Eduv Platform draws inspiration from literature and shares similarities with Reeborg's World and similar endeavors [134], [135] that employed virtual robots. However, a notable differentiation lies in the EDUV Platform, which enables users to remotely control an actual robot, thereby fostering a more imaginative, captivating, and enjoyable learning experience.

7.3 The Underwater Robot EDUV

A significant proportion of underwater vehicles have a high cost and limited capacity for customization. In contrast to conventional underwater vehicles, the design of EDUV distinguishes itself by virtue of its adaptability, portability, and cost-effectiveness. The fundamental objective of the EDUV is to achieve affordability and flexibility. According to this perspective, the robot is constructed using easily replaced materials. Due to this rationale, using PLA filament in 3D printing, which possesses water and humidity resistance properties, emerges as the most optimal approach for addressing the constraints.

The sketch was created using the Autodesk Fusion360 CAD Software. The designed components exhibit high interchangeability, facilitating rapid construction, replacement, and modification of the 3D-printed body. The vehicle's tiny size facilitates convenient transportation and manipulation by a single person. The dimensions in Figure 28 are presented as 376x300x87 mm (width x length x height), and it weighs 1.5 kg when its ballast tanks are empty. Furthermore, as depicted in Figure 28, the structure consists of three primary components: the central cylinder, referred to as "P2," and the ballast tanks, denoted as "P1" and "P3." Furthermore, the system is outfitted with a total of six thrusters, with four of them being grouped in a vertical configuration denoted as Motors M1-4, forming a square shape. The remaining two thrusters, known as Motors M5-6, facilitate surge. In order to improve the stability of the horizontal axis and prevent roll motion, two fins are affixed to the front-facing horizontal motors (Figure 28, denoted as F1-F2).

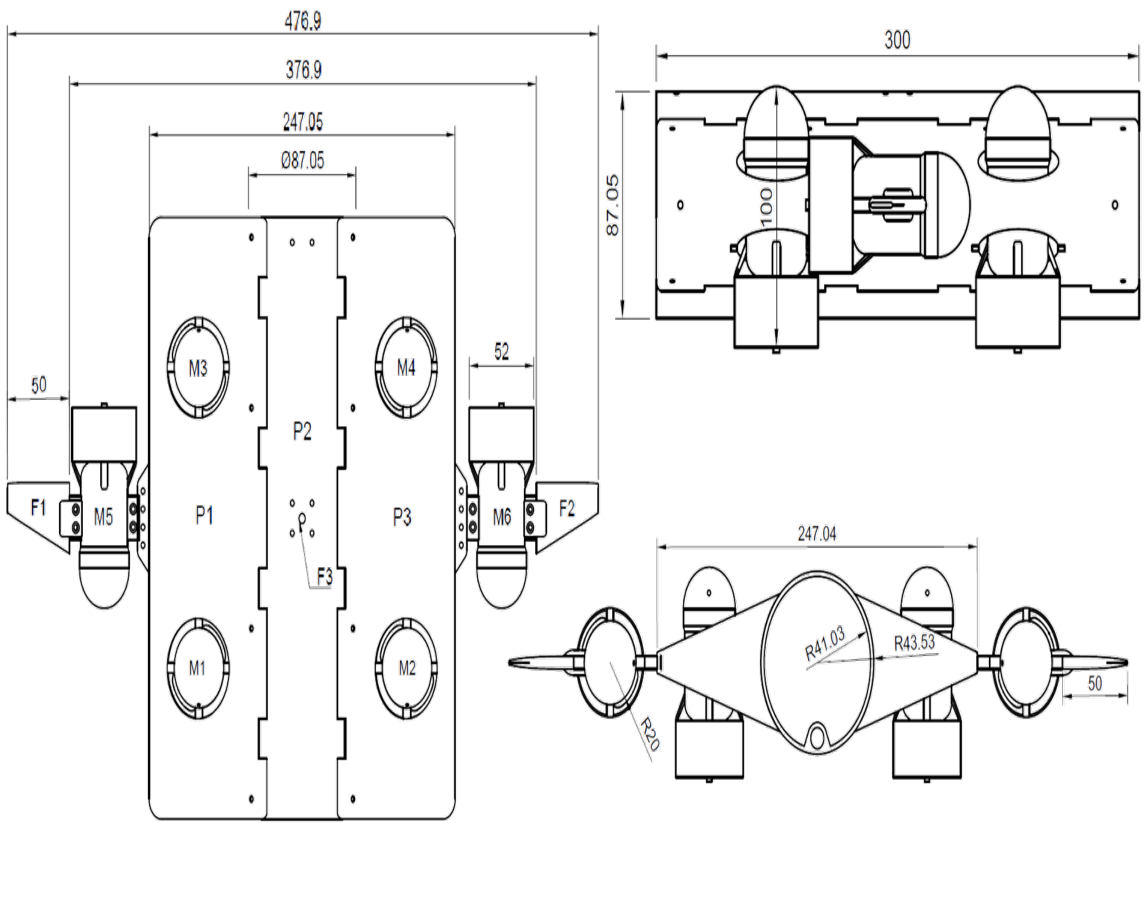


Figure 28: EDUV schematics

The term "mobility" relates to the ability of a robot to navigate its environment without constraints, while mobility analysis involves examining how this movement is decomposed into its constituent components. It is important to fully comprehend the vehicle's capabilities, encompassing its strengths and weaknesses. The vehicle's thruster arrangement enables it to possess five Degrees of Freedom and execute the following movements, as depicted in Figure 29:

- The translational-surge motion is achieved through the utilisation of two thrusters that are horizontally aligned. These thrusters enable linear longitudinal movement, allowing for forward and backward motion.
- The translational-heave capability is achieved using four vertically oriented thrusters, enabling a linear vertical upward or downward motion.
- Translational sway refers to a linear transverse motion along the Y axis. While the vehicle cannot execute this movement, it can be approximated by combining translational and rotational motions. For instance, a combination of movement in the roll axis and vertical motion.
- The rotational manoeuvre, known as a roll, involves the lateral tilting of a vehicle along its longitudinal (X) axis, utilising the four vertical thrusters at its disposal.
- The robot can perform rotational-pitch movement by utilising its four vertical thrusters. This enables the robot to spin along its Y-axis upward or downward.
- The rotational-yaw motion of the vehicle involves the rotation around its vertical (Z) axis, which is achieved through its two horizontal motors.

Furthermore, it has been observed that fins actively contribute to the stability and movement of the robot. Two fins are affixed to the horizontal motors, positioned in a forward-facing orientation. Essentially, the presence of forward-facing horizontal fins reduces the ease of roll and enhances stability. The application environment necessitates stability as a crucial factor, mainly due to the involvement of students in programming activities.

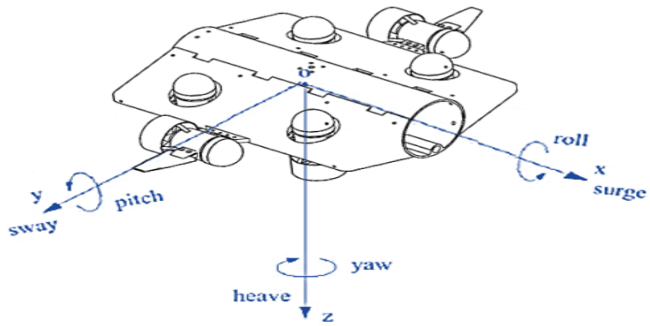


Figure 29. The movement of EDUV in its 3-dimensional enviroment

The vehicle was subjected to testing in a marine environment within a depth range of 0 to 5 meters. It was paramount to evaluate the vehicle's stability and its rotational and two translational motions. The experimental evaluation of EDUV encompassed various rotating motions and surge at a depth of approximately 0.5 meters. The heave motion was also examined within a 0 to 1 meter depth range. The vehicle was equipped with the IMU MPU-6050 to conduct these measurements. The observed results were consistent with expectations, as the robotic system demonstrated high accuracy and swiftness in its response to roll, pitch, yaw, surge, and heave inputs. To be more precise, the vehicle exhibits a velocity of 0.6 m/s when undergoing surge translational motion and 0.35 m/s during heave translational motion. The vehicle can rotate at angular velocities of 59 degrees per second for roll, 53.7 degrees per second for pitch, and 68.6 degrees per second for yaw.

The mobility of the robot is dependent on six direct current (DC) motors, which are regulated by twelve miniature relays. The motors exhibit a characteristic of 800 revolutions per minute per volt, with a maximum efficiency current of 0.74 amperes when powered by a 3-cell battery. The battery cells in question are lithium-polymer (Li-Poly) cells with a rated capacity of 2700mAh and an output voltage of 11.1 volts. The motors are supplied with power from the battery via the relays, operating within a voltage range of 9.6 to 12.6 Volts, with a nominal rating of 11.1 Volts. On the other hand, the single-board computer and camera are powered by a direct current step-down converter, which is connected to the battery and produces a stable output of five volts. The instructions for controlling the vehicle are derived from a cost-effective microcontroller, which was essential for overseeing the movement of the motors. Specifically, the device in question is a Raspberry Pi 4 Model B equipped with 2GB of RAM. The Raspberry Pi single-board computer is utilised to establish a connection between a submersible camera and a surface device, enabling the transmission of visual feedback. One crucial aspect of the suggested underwater vehicle is its affordability. The overall cost of the robot exhibits fluctuation around 160 Euros. Table 10 contains the components of the robot along with their respective pricing.

Table 10: EDUV's Components Prices

Part	Qty	Total Cost
DC Motors	6	36,00 €
Mini relays	6	6,00 €
Raspberry Pi 4 Model B 2GB	1	50,00 €
LiPo Battery 2700mAh 11.1V	1	21,00 €
DC Step down to 5v 3A	1	2,00 €
Camera	1	11,00 €
Propellers	6	4,00 €
PLA 3D filament (in Kg)	1	19,00 €
Cables etc. (approximately)	-	10,00 €
Total		159,00 €

7.4 The EDUV Platform

7.4.1 The EDUV Platform Set Up

The Platform's set up mimics the set up used for the El Greco Platform presented in section 6.3.1 of the dissertation. One notable distinction is in the utilisation of the inbuilt camera of the EDUV Platform (Figure 30), whereas El Greco platform relies on an external camera that captures a panoramic view of the robot (Figure 13). One more distinction lies in the connectivity method employed by the server in the El Greco Platform, which utilises an ethernet connection to access the Internet. Conversely, in the EDUV platform, the server is established on a laptop that connects to the Internet via a mobile phone, utilising cellular data. The commands are compiled in the Raspberry Pi, which serves as the central control

Figure 30: Live streaming from EDUV's onboard camera

unit for controlling the movement of the underwater vehicle. The Raspberry Pi and the laptop are wirelessly connected to the access point through Wi-Fi. A client computer connected to the internet lets platform users browse the website (Figure 31).

Figure 31:View of the Hardware employed for the needs of the EDUV platform.

7.4.2 The EDUV Platform's Features

The EDUV Platform only employs the main platform of the El Greco Platform's website. The primary factor influencing this choice was the constraints imposed by the deployment environment of EDUV (Figure 32). Due to the absence of visual cues in an open sea setting, a game such as El Greco Adventures may lack the essential visual information required for comprehending the robot's activities. The required modifications were made to assist the implementation of EDUV. However, it should be noted that the

Figure 32: EDUV's Deployment

EDUV platform is identical to the El Greco main platform, as described in section 6.3.2 of this dissertation.

The most significant modification to the original platform involved incorporating EDUV movements. The EDUV platform provides support for the following moving commands (Figure 33):

- Forward for n seconds: The vehicle exhibits surge translational motion and undergoes forward movement for n seconds.
- Backwards for n seconds: The vehicle executes a surge translational motion, resulting in a backward movement for a specified period of time.
- Emerge for n seconds: During a specified duration of n seconds, the vehicle executes a heave translation motion, resulting in an upward movement.
- Dive for n seconds: The vehicle does a heave translational action, descending in a downward direction.
- Turn Right for n seconds: The vehicle executes a yaw rotational motion and undergoes a right turn for a duration of n seconds.
- Turn Left for n seconds: The vehicle engages in yaw rotational motion and executes a left turn for a duration of n seconds.
- Turn Up for n seconds: The vehicle executes a pitch rotational motion and undergoes an upward turn for a duration of n seconds.
- Turn Down for n seconds: The vehicle executes a pitch rotational motion and undergoes a downward turn for a duration of n seconds.
- Roll Right for n seconds: The vehicle engages in a roll rotational action, executing a rightward roll for a duration of n seconds.
- Roll Left for n seconds: The vehicle executes a roll rotational motion by rolling to the left for a duration of n seconds.

The commands mentioned above have been implemented using the Python programming language, with each command representing a distinct function within the underwater vehicle. These functions are executed by the Raspberry Pi single-board computer, situated on the surface and connected to the motors through a waterproof cable for power transmission. Furthermore, the operator must reset the vehicle to its initial location following each test.

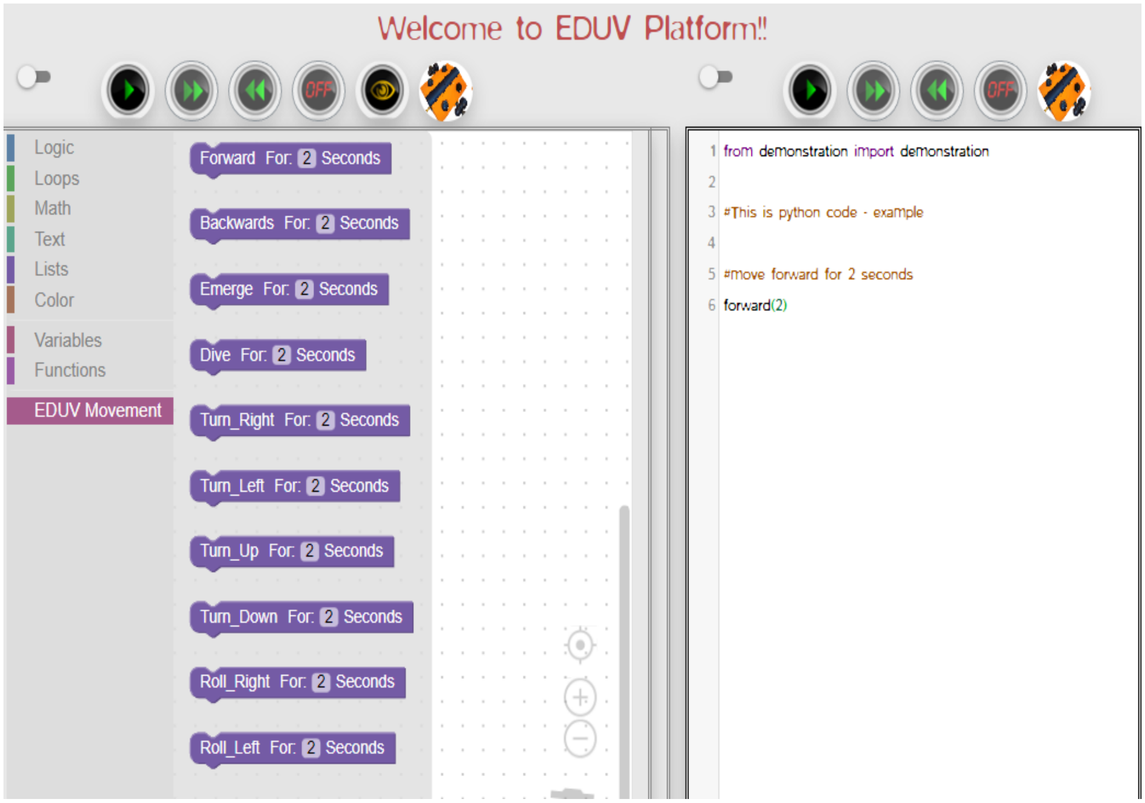


Figure 33: The EDUV Platform's code area and Blockly area illustrating the EDUV's Movement Functions.

7.5 THE EDUV Survey

7.5.1 Study Design

The study design adhered to the study's logic on the El Greco platform, as outlined in section 7.4.1 of this dissertation. Specific questions were modified or omitted, while others were included. This was done in response to criticisms about the El Greco Platform survey's lengthy questionnaire. The primary distinction is in utilising three Blockly-based programming questions and one Python-based programming question, in contrast to the arrangement seen in the El Greco platform survey, which consisted of two Blockly-based programming questions and two Python-based programming questions. This approach was preferred because the platform's main target audience consists of inexperienced users. It is important to note that the results of each Likert-type question were analyzed in a manner where a higher score denoted a more positive outcome. The methodology employed is the same as the El Greco Platform, illustrated in Figure 34.

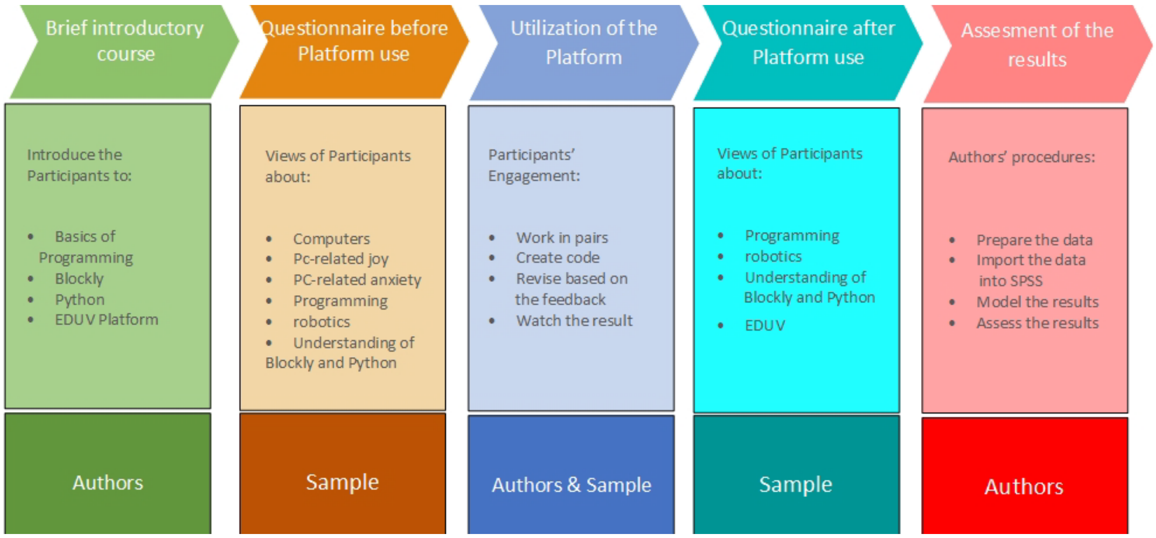


Figure 34: Processes involved in the EDUV platform survey's design.

7.5.2 Participants and Preliminary Findings

The survey contained a sample of 112 individuals from Greece, consisting of sixty-four males and forty-eight females, aged between 14 and 18 years. Figures 35 and 36 display the distribution of age and grade among the student population. It must be noted that the sample differentiates in the El Greco Platform survey.

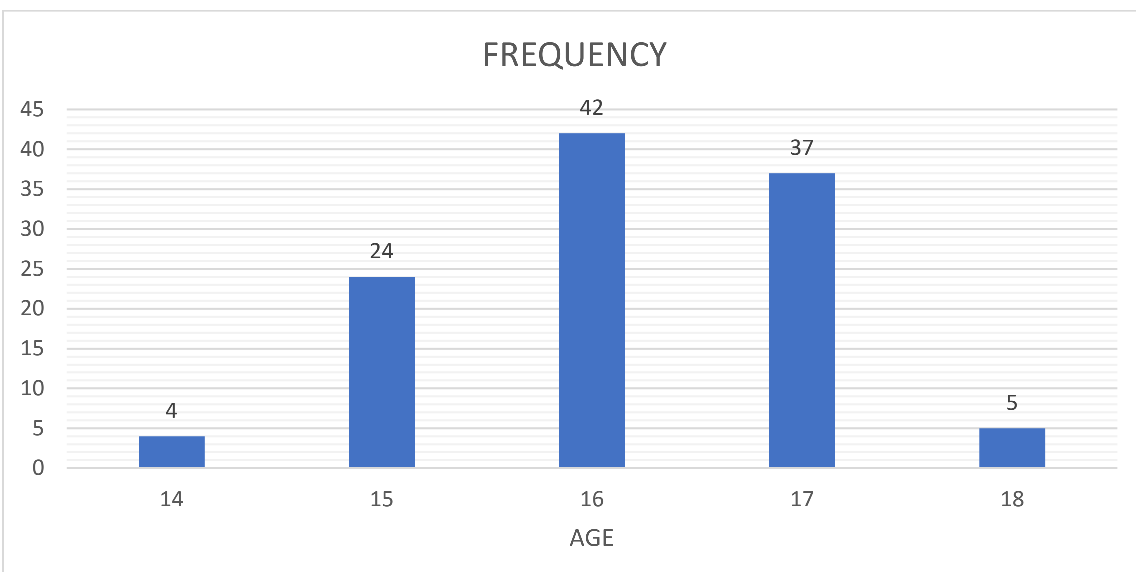


Figure 35: Age distribution of the EDUV Platform survey.

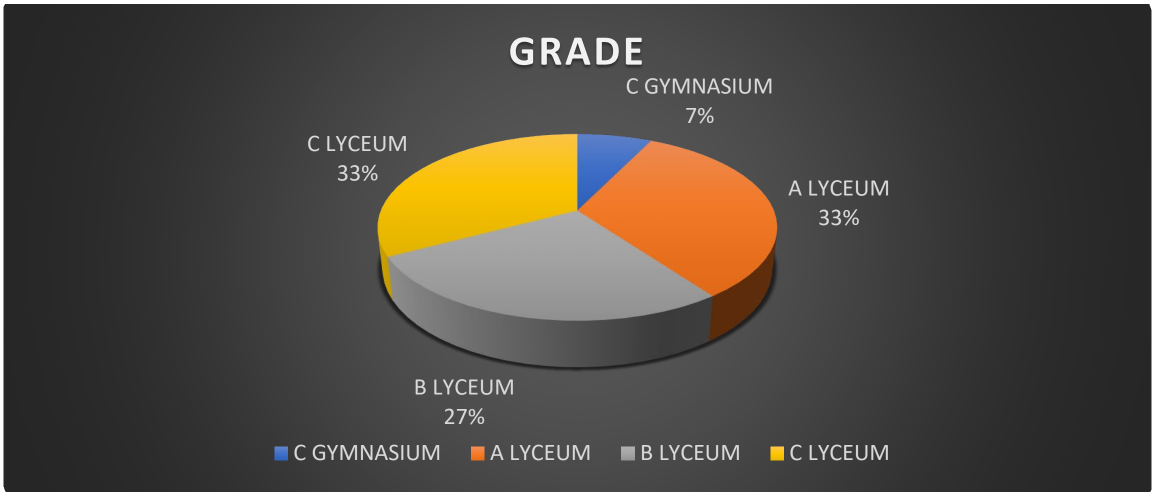


Figure 36: Grade distribution of the EDUV Platform survey.

Participants reported a range of computer usage experience, ranging from a minimum of one year to a maximum of ten years, with a mean value of 4.43. Moreover, the participants' average weekly computer usage, regardless of the type of computer, amounts to 13.56 hours. The majority of the individuals utilise the computer for a duration of 5 to 16 hours a week. Based on the findings, it can be inferred that the students incorporate the usage of computers into their everyday routines.

The results obtained from questions associated with the Computer Attitude Questionnaire V5.22 indicate that the sample had a mean score of 3.5938 in terms of the perceived importance of computers, 3.9125 in terms of enjoyment derived from using computers, and 2.1696 in terms of anxiety experienced when using computers. Based on the findings, it is justifiable to infer that the participants saw computers as a significant component of their daily routines, using them with enjoyment and minimal stress. The descriptive statistics of the corresponding subscales of the Computer Attitude Questionnaire are illustrated in Table 11.

Table 11: Descriptive statistics of the F1, F2 and F7 subscale of the Computer Attitude Questionnaire V5.22 for the EDUV Platform survey.

	N	Min	Max	Mean	Std. Deviation
CAQF1	112	1.50	5.00	3.5938	1.08123
CAQF2	112	2.00	5.00	3.9125	0.83214
CAQF7	112	1.00	4.50	2.1696	1.09986

The analysis of the SCAPA questionnaire related questions reveals that the sample population had an average score of 3.0537 on self-reported programming understanding. This finding suggests that most study participants are unfamiliar with computer programming. Furthermore, it is noteworthy that the sample exhibited a mean score of 2.4286 concerning their belief regarding programming cost. This finding suggests a general consensus among the participants that programming does not entail a substantial cost. Furthermore, based on the obtained mean score of 2.9464 for the belief in the intrinsic value of programming, it may be concluded that the students' attitudes towards programming are not particularly strong in either a positive or negative direction. In accordance with this assumption, we were also directed by the mean of 2.5268 in programming utility value belief and by the mean of 2.8929 in programming compliance. Most importantly, the observed data suggests that the group has a tendency for persistence in programming assignments, as evidenced by the marginally elevated average score of 3.6964. The standard deviation of each scale provides empirical evidence in support of the theories that have been previously expressed. These standard deviations are presented in Table 12.

Table 12: Descriptive statistics of the SCARPA subscales related to the EDUV Platform.

	Min	Max	Mean	STD
Self-reported programming understanding	1.00	5.00	3.0357	0.91709
Programming intrinsic value belief	1.00	5.00	2.9464	1.18997
Programming utility value belief	1.00	5.00	2.5268	1.09429
Programming cost belief	1.00	5.00	2.4286	1.03727
Programming persistence	3.00	5.00	3.6964	0.76922
Programming compliance	1.00	5.00	2.8929	1.13389

The subsequent section of the first questionnaire was based on the 4-H Robotics and GPS/GIS Interest Questionnaire to gain further insight into the participants' beliefs regarding robotics. The observed mean score for this survey section was 2.5306, with a corresponding standard deviation of 0.89624. Based on these findings, it can be inferred that the sampled individuals possess limited prior knowledge of robotics and do not hold strong opinions on the subject matter. Furthermore, the respondents indicated that they had prior lessons in programming, with an average rating of 3.54 on a scale of 1 to 5 and a standard deviation of 1.138. The initial questionnaire included a set of questions aimed at assessing participants' prior knowledge of Python and Blockly. These questions are presented in Table 13.

Table 13: Descriptive statistics of former experience in Blockly and Python related to the EDUV Platform survey.

	Min	Max	Mean	STD
I can program in Python	1.00	5.00	1.96	1.150
I have heard of Blockly	1.00	5.00	3.09	1.418
I can program in Blockly	1.00	5.00	2.64	1.361

These findings provide evidence that most participants had a restricted understanding of Python and Blockly. This notion was reinforced by the four programming questions in the last section of the initial questionnaire. The participants were instructed to analyze the outcomes of running four concise and comparatively uncomplicated programs. Specifically, three programs were coded using the Blockly platform, while the other one was coded in Python. The following section of this dissertation will present the replies submitted by the participants and their associated percentages compared to the results obtained after utilising the EDUV platform.

In summary, according to the findings of the initial questionnaire, most of the participants demonstrate a level of familiarity with computers, while their understanding of robotics and computer programming is deemed insufficient. Due to these features, they serve as optimal candidates for assessing the pedagogical effectiveness of the EDUV platform, specifically concerning the acquisition of programming skills and overall satisfaction with the learning experience.

7.5.3 Analysis of the second questionnaire.

Following a brief introduction to the platform, the Python programming language, and the utilisation of the platform, the second questionnaire was filled out. The purpose of its creation was to examine the benefits of utilising the EDUV Platform within the realm of education. The participants spent around 30 minutes engaging with the website in pairs.

A paired-sample t-test was used to examine the Platform's impact on the participants' beliefs. The statistical significance of all the data is evident in Table 14, suggesting that the participants' initial beliefs about programming were positively influenced following their utilisation of the EDUV platform. Based on the calculated mean and T values, it can be inferred that the platform exerted a statistically significant favourable influence on the participants. This is further corroborated by the Sig. Values, which are observed to be at a significantly low level.

Table 14: The dependent t-test analysis of the beliefs of the sample about programming, based on the SCAPA and 4-H Robotics and GPS/GIS Interest questionnaires prior to and after the use of the EDUV Platform.

	Paired Differences					t	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference			
				Lower	Upper		
Self-reported programming understanding (SCAPA)	1.30804	0.76301	0.07210	1.16517	1.45090	18.143	$5.44 \cdot 10^{-35}$
Programming intrinsic value belief (SCAPA)	1.02679	0.76181	0.07198	0.88414	1.16943	14.264	$7.43 \cdot 10^{-27}$
Programming utility value belief (SCAPA)	0.70982	0.67663	0.06394	0.58313	0.83651	11.102	10^{-19}
Programming cost belief (SCAPA)	0.64286	0.78105	0.07380	0.78910	0.49661	8.711	$3.25 \cdot 10^{-14}$
Programming persistence (SCAPA)	0.43750	0.54988	0.05196	0.33454	0.54046	8.420	$1.49 \cdot 10^{-13}$
Programming compliance (SCAPA)	0.87500	0.85028	0.08034	0.71579	1.03421	10.891	$3.15 \cdot 10^{-19}$
Pair 1 4H – 4H after	1.33163	0.58517	0.05529	1.44120	1.22207	24.083	$6.91 \cdot 10^{-46}$

Additionally, the paired sample T-test findings indicated that the EDUV Platform positively impacted participants' perceptions of robotics, enhancing their motivation to actively seek opportunities for advancing their knowledge in this field. The t score for the questionnaire relating to 4-H has the highest value; however, the Sig. value is the lowest. This discovery allows us to infer that the EDUV platform significantly influenced the participants' attitudes regarding robotics.

Tables 15 through 18 illustrate the percentages of participants' responses to the four control questions related to programming before and after using the EDUV Platform.

Table 15: The difference in percentages to the answers for the first Blockly program related to the EDUV Platform. survey.

	Percent (Before)	Percent (After)	Difference
Correct	42.0	92.0	50.0
Don't know	39.3	1.8	-37.5
Wrong	18.7	6.2	-12.5

Table 16: The difference in percentages to the answers for the second Blockly program related to the EDUV Platform. survey.

	Percent (Before)	Percent (After)	Difference
Correct	37.5	92.9	55.4
Don't know	42.9	1.8	-41.1
Wrong	19.6	5.3	-14.3

Table 17: The difference in percentages to the answers for the second Blockly program related to the EDUV Platform survey.

	Percent (Before)	Percent (After)	Difference
Correct	36.6	80.4	43.8
Don't know	50.9	3.6	-47.3
Wrong	12.5	16.0	3.5

Table 18: The difference in percentages to the answers for the Python program related to the EDUV Platform survey.

	Percent (Before)	Percent (After)	Difference
Correct	13.4	67.0	53.6
Don't know	73.2	13.4	-59.8
Wrong	13.4	19.6	6.2

According to the data presented in Tables 15 - 18, it is evident that there has been a notable increase in the frequency of accurate responses observed in both the Python and Blockly environments. Moreover, subsequent to the utilisation of the EDUV platform, there was a significant decrease in the frequency of "Do not know" responses. The Python programming question, which was presumably more difficult for the participants, serves as more evidence to support this observation. To clarify further, the observed percentages of accurate responses exhibit an upward trend, precisely measuring 50.0%, 55.4%, 43.8%, and 53.6% across the four code questions, respectively. Nevertheless, the Python question exhibited fewer correct answers, amounting to 67%. In all instances of programming examples, the participants demonstrated improved performance compared to their initial exposure to this programming framework.

7.6 Discussion

The task of fostering student motivation and dedication towards their education might pose difficulties, particularly when the educational delivery is conducted through online platforms. Robotics enables the manifestation of students' creative and innovative abilities. Using educational robotics transforms the traditional pedagogy framework, redirecting attention towards the student and emphasizing collaborative learning. This section of the dissertation presented an underwater vehicle designed for use in education through an online platform without limiting its possible utilisation in blended or traditional classroom environments. The primary objective of this vehicle is to foster creativity and enhance student involvement. Additionally, the underwater vehicle is very adaptable and cost-effective.

Based on the data mentioned earlier, it can be concluded that the EDUV platform successfully captivated the survey participants and positively influenced their attitudes towards robotics and programming. Participants made full use of the platform to enhance their understanding of programming. Furthermore, this fact served as a catalyst for them to actively seek out further opportunities to enhance their proficiency in programming and robotics. The utilisation of both Blockly and Python offers the advantage of enabling users to familiarize themselves with and acquire coding skills in two distinct coding environments.

Chapter 8. Conclusions and Future Work

8.1 Comparative Analysis of the Surveys Contacted

The collected survey samples exhibit variations in age and grade, as depicted in Figures 37 and 38.

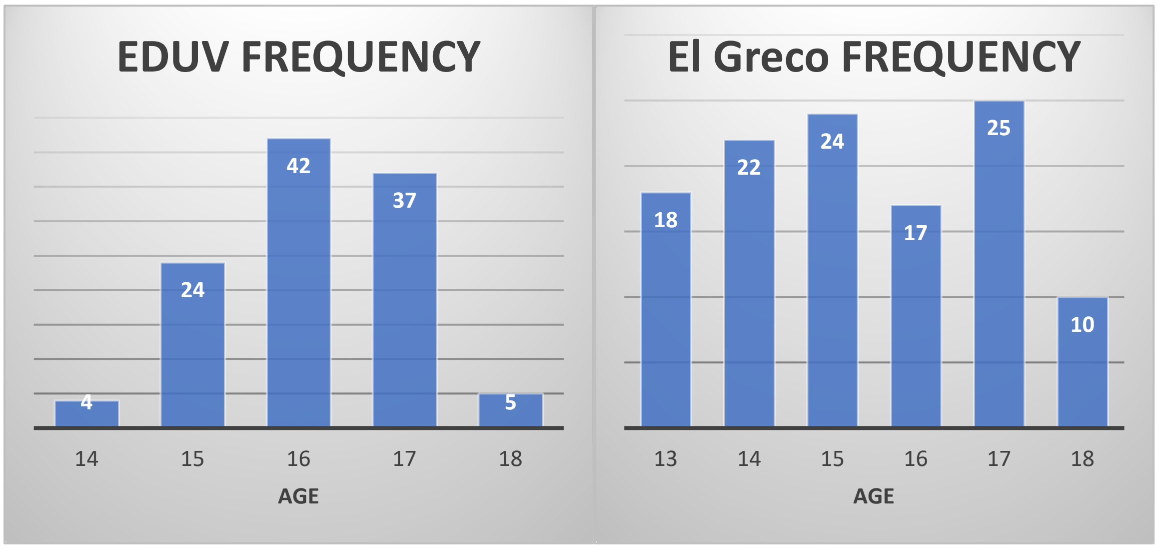


Figure 37: Age distribution of the surveys.

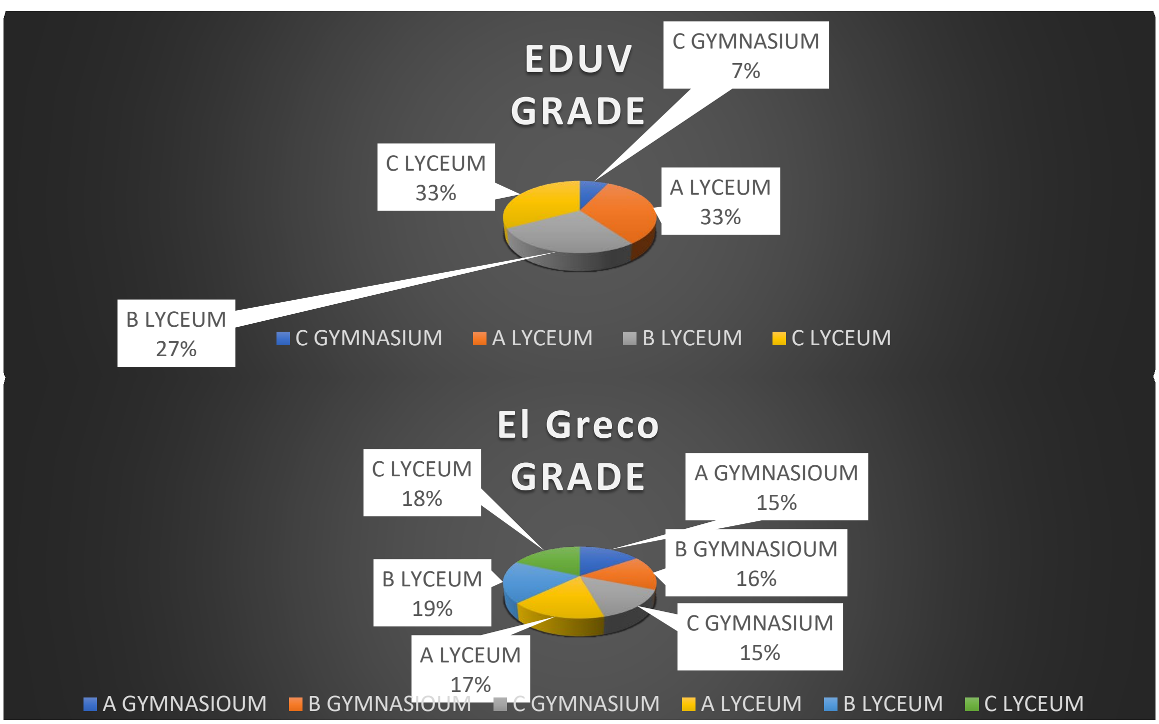


Figure 38: Grade distribution of the surveys.

An independent t-test was performed to examine variations in computer usage over previous years and the number of hours spent using a computer weekly. The results of this analysis are presented in Table 19. Based on these findings, it may be inferred that the samples exhibited a significant differentiation in weakly computer usage.

Table 19: Independent t-test scores for computer usage.

	Mean Difference	t	SIG
Years (average)	1.49	4.503	0.000
Weekly (average)	-1.880	-1,992	0.48

Additionally, an independent t-test was conducted to examine whether the sample had similar attitudes towards programming before utilising the platforms, as measured by the Computer Attitude Questionnaire V5.22 subscales. The results are presented in Table 20. These results suggest that the samples demonstrate comparable attitudes and perspectives about programming.

Table 20: Independent t-test scores on the Computer Attitude Questionnaire V5.22 subscales before using the platforms.

	Mean Difference	t	SIG
CAQF1	0.197	1.558	0.199
CAQF2	-0.058	-0.590	0.556
CAQF7	0.132	0.991	0.323

Additionally, a comparison of attitudes prior to using the platforms, measured by the SCAPA questionnaire subscales and the 4-H Robotics and GPS/GIS Interest questionnaire, was carried out employing an independent t-test. The findings are presented in Table 21. On the basis of these data, we can assume that the samples hold similar beliefs regarding their understanding of programming and programming's intrinsic value. On the other hand, participants of the El GRECO believed more strongly that programming is useful and does not entail significant costs. Additionally, they hold positive views on robotics. Nevertheless, EDUV's survey participants demonstrate a higher level of persistence in programming activities.

Table 21: Independent t-test scores on the SCAPA and 4H questionnaires before using the platforms.

	Mean Difference	t	SIG
Self-reported programming understanding (SCAPA)	0.104	1.013	0.312
Programming cost belief (SCAPA)	0.787	6.625	0.000
Programming intrinsic value belief (SCAPA)	0.183	1.394	0.161
Programming utility value belief (SCAPA)	0.91	7.358	0.000
Programming persistence (SCAPA)	-0.356	-3.331	0.001
4H	0.607	6.158	0.000

In order to assess the similarity of prior knowledge in Python and Blockly among the samples, an independent t-test was conducted. The findings are presented in Table 22. Based on the aforementioned data, it can be argued that the El Greco survey participants had a higher proficiency level in Python and Blockly. However, this assumption is inconsistent with the data collected prior to using the platforms, as EDUV's survey sample exhibited a higher score on accurate responses. These findings are based on the four programming questions included at the end of the first questionnaire and are presented in Tables 22 and 23.

Table 22: Independent t-test scores of former experience to Blockly and Python

	Mean Difference	t	SIG
I can program in Python	0.370	3.014	0.003
I have heard of Blockly	-0.790	-5.341	0.000
I can program in Blockly	0.142	-2.457	0.015

Table 23: The percentages of the answers to the Blockly programming questions.

	El Greco	**EDUV**	
	Percent	**Percent**	**Difference**
Correct	19.4	38.7	-19.3
Wrong	5.6	16,9	-11.3
Do not know	75	44.4	30.6

Table 24: The percentages of the answers to the Python programming questions

	El Greco	EDUV	
	Percent	Percent	Difference
Correct	12.5	13.4	-0.9
Wrong	8.6	13.4	-4.8
Do not know	78.9	73.2	5.7

The findings presented above support the notion that the samples of the conducted surveys differ from one another to some degree; nonetheless, participants shared a common view of programming, and as it was argued in the previous chapters of this dissertation, both samples' exposure to robots and expertise in programming was very limited. Because of these characteristics, they are good candidates for a study into the pedagogical merits of the platforms concerning programming education and the degree of pleasure that can be attained throughout the education process.

Regarding evaluating the platforms' influence on the participants, an independent t-test was conducted to compare the means of the SCAPA questionnaire subscales and the 4-H Robotics and GPS/GIS Interest questionnaire before and after using the Platform. The results of this analysis are presented in Table 25. The data presented provide evidence that compared to the El Greco Platform, the EDUV Platform has a more significant positive effect on individuals' beliefs regarding the inherent worth of programming, their compliance with programming tasks, and their perception of Robotics. This discovery can be attributed to the nature of EDUV, which is more intriguing than El Greco due to the deployment environment. It is important to note that before utilising the Platform, the El Greco survey sample demonstrated a comparatively elevated score in the 4H section of the Questionnaire. This observation is supported by the fact that the 4H score exhibits the most significant difference in mean values.

On the other hand, it is evident that those who participated in the surveys demonstrated similar convictions regarding the belief in programming costs after utilising the platforms.

Table 25: Independent t-test scores of the beliefs of the samples about programming and robotics after and before the use of the platforms.

	Mean Difference	t	SIG
Programming cost belief (SCAPA)	0.137	1.158	0.248
Programming intrinsic value belief (SCAPA)	-0.294	-2.552	0.011
Programming compliance (SCAPA)	-0.642	-5.815	0.000
4H	-0.809	-9.633	0.000

An analysis of the data presented in Tables 26 and 27 allows for an assessment of the relative impact of the platforms on Blockly and Python programming skills. The findings indicate that in terms of Blockly skills, the El Greco platform had a more advantageous effect on participants, as evidenced by their higher number of correct answers and a notable decrease in the frequency of the "I do not know" response compared to the EDUV participants. In contrast, the participants in the EDUV sample demonstrated an enhanced comprehension of Python after they utilised the platform, which was also evidenced by a substantial rise in the number of accurate responses and a noteworthy decline in the frequency of selecting the "I do not know" option.

Table 26: Comparative analysis of the Blocly programming tasks related to the platforms.

	El Greco	EDUV	Platform Difference
	Difference (Percent)	Difference (Percent)	
Correct	59.5	49.73	9.77
Wrong	0	-7.77	7.77
Do not know	-59,5	-41.97	-17.56

Table 27: Comparative analysis of the Python programming tasks related to the platforms

	El Greco	EDUV	Platform Difference
	Difference (Percent)	Difference (Percent)	
Correct	35.35	53.6	-19.3
Wrong	8.75	6.2	2.55
Do not know	-43.15	-59.8	16.65

8.2 Discussion

As previously mentioned in the preceding section of this dissertation, the surveyed samples exhibit variations, although they are both appropriate choices for assessing the pedagogical advantages of the platforms in the context of programming education. Furthermore, as elaborated in sections 6.4 and 7.4 of this dissertation, both platforms stimulated substantial interest among the participants of the studies and had a beneficial impact on their attitudes towards programming and robotics. Furthermore, they substantially impacted their understanding of programming and acted as a catalyst for exploring further options to augment their proficiency in robotics and programming.

These surveys exhibit a shared susceptibility. Both surveys employ anonymous questionnaires, preventing the tracking of the effect of these platforms to every individual of the sample. Additionally, a post-intervention survey was not conducted to assess the long-term sustainability of the platforms' impact.

Regarding the research questions of this dissertation, we can conclude the following:

RQ1: Does the Platform facilitate enhancing the survey participants' creativity and motivation?

The examination of the collected data indicates that it is possible to deduce that the Platform intrigues the learners and promotes their creative thinking. Moreover, using the platforms encouraged the exploration of additional avenues to enhance their expertise in robotics and programming. Furthermore, the El Greco platform, particularly the El Greco Adventures mode, can be seen from the game-based learning and gamification perspective. This factor can potentially enhance the joy and active engagement in the educational process.

RQ2: How does the Platform influence survey participants' attitudes towards programming and robotics?

Based on the results obtained from the conducted surveys, it can be concluded that the participants' initial attitudes and preconceptions about programming and robots experienced positive transformations following their engagement with both platforms. The

EDUV platform, in particular, produced an extensive and favourable influence on the sample's attitudes towards robotics. One potential rationale is that the inherent characteristics of EDUV render it appealing for the individuals participating.

RQ3: What is the impact of the Platform on the programming comprehension and skills of the survey participants?

The findings suggest that the El Greco platform had a more favourable impact on participants' Blockly skills in comparison to the EDUV participants. On the other hand, the individuals involved in the EDUV study had an improved understanding of Python after using the platforms. However, both platforms positively impacted their understanding of programming and overall coding abilities.

RQ4: Are the Platform and framework utilised considered to be versatile?

The integration of El Greco and EDUV with the Platform was accomplished effortlessly. The Platform received minimal modifications to facilitate the respective functionalities of the robots. Based on this fact, it can be argued that the employed Platform and framework exhibit versatility. It is conceivable to argue that the utilisation of the framework is compatible with any robotic system capable of executing Python scripts.

8.3 Future Work

Future work involves achieving full autonomy in both the booking service and the deployment of the robots. In the context of the El Greco Platform, it is essential to implement modifications to the El Greco playground and enhance its movement precision, hence eliminating the need for a supervisor during sessions. This objective can be achieved by utilising a new, similar, yet enhanced robot. Concerning the EDUV, it is imperative to consider the robot's deployment at a pool to mitigate the impact of weather conditions and assess its capacity for unsupervised operation. This application may be beneficial for educational institutions with a pool.

Moreover, it is necessary that additional Adventure levels and novel functions be developed in order to enhance the educational experience by fostering creativity, increasing engagement, and promoting enjoyment. In addition, a potentially intriguing strategy for ground-based robots like El Greco would be the establishment of playrooms resembling the virtual environments found in Reeborg's World. Also, using many robots to enable simultaneous and multiple-user interaction with a robot is an appealing possibility.

Finally, with more widespread use of the Platform, data mining techniques can collect information from the files gathered from the automatic log file system. This information has the potential to be exploited for the improvement of the Platform.

References

[1] P. Tuomi, J. Multisilta, P. Saarikoski, and J. Suominen, “Coding skills as a success factor for a society,” *Educ. Inf. Technol.*, vol. 23, pp. 419–434, 2018.

[2] E. Loh, “Medicine and the rise of the robots: A qualitative review of recent advances of artificial intelligence in health,” *BMJ Leader*, vol. 2, no. 2. BMJ Publishing Group, pp. 59–63, Jun. 2018. doi: 10.1136/leader-2018-000071.

[3] C. Yang, J. Luo, Y. Pan, Z. Liu, and C. Y. Su, “Personalized Variable Gain Control with Tremor Attenuation for Robot Teleoperation,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 48, no. 10, pp. 1759–1770, Oct. 2018, doi: 10.1109/TSMC.2017.2694020.

[4] S. H. Ivanov, C. Webster, K. Berezina, and S. Ivanov, “Adoption of robots and service automation by tourism and hospitality companies SPECIAL INTEREST TOURISM View project HOSPITALITY MARKETING AND MANAGEMENT View project Adoption of robots and service automation by tourism and hospitality companies,” 2017.

[5] M. L. Lupetti, “Designing playful HRI. Acceptability of robots in everyday life through play,” in *ACM/IEEE International Conference on Human-Robot Interaction*, Apr. 2016, vol. 2016-April, pp. 631–632. doi: 10.1109/HRI.2016.7451891.

[6] C. M. Kandemir, F. Kalelioğlu, and Y. Gülbahar, “Pedagogy of teaching introductory text-based programming in terms of computational thinking concepts and practices,” *Comput. Appl. Eng. Educ.*, vol. 29, no. 1, pp. 29–45, Jan. 2021, doi: 10.1002/cae.22374.

[7] A. M. Müller, C. Goh, L. Z. Lim, and X. Gao, “COVID-19 Emergency eLearning and Beyond: Experiences and Perspectives of University Educators,” *Educ. Sci.*, vol. 11, no. 1, 2021, doi: 10.3390/educsci11010019.

[8] M. N. O. Sadiku, P. O. Adebo, and S. M. Musa, “Online teaching and learning,” *Int. Journals Adv. Res. Comput. Sci. Softw. Eng.*, vol. 8, no. 2, pp. 73–75, 2018.

[9] V.-M. Cojocariu, I. Lazar, V. Nedeff, and G. Lazar, “SWOT anlysis of e-learning educational services from the perspective of their beneficiaries,” *Procedia-Social Behav. Sci.*, vol. 116, pp. 1999–2003, 2014.

[10] V. Singh and A. Thurman, “How many ways can we define online learning? A systematic literature review of definitions of online learning (1988-2018),” *Am. J. Distance Educ.*, vol. 33, no. 4, pp. 289–306, 2019.

[11] G. O’Regan and G. O’Regan, “George Devol,” *Giants Comput. A Compend. Sel. Pivotal Pioneers*, pp. 99–101, 2013.

[12] S. Papert, “What is Logo? And Who Needs It?,” 1999, Accessed: Jan. 05, 2023. [Online]. Available: http://www.microworlds.com/company/philosophy.pdf

[13] T. Sapounidis and D. Alimisis, “Educational robotics for STEM: A review of technologies and some educational considerations Edu4AI View project Robotics-based learning interventions for preventing school failure and Early School Leaving View project,” 2020, Accessed: Jan. 05, 2023. [Online]. Available: https://www.researchgate.net/publication/346588762

[14] D. H. Clements and J. S. Meredith, “Research on Logo: Effects and Efficacy,” 1992, Accessed: Jan. 05, 2023. [Online]. Available: www.logofoundation.org

[15] S. Papert, “Teaching children to be mathematicians us. teaching about mathematics,” 1980.

[16] L. Armstrong and A. Tawfik, “The History of Robotics and Implications for K-12 STEM Education,” *TechTrends*, vol. 67, no. 1, pp. 14–16, 2023.

[17] “Media Lab gets serious with toys | MIT News | Massachusetts Institute of Technology,” 1997. https://news.mit.edu/1997/lego-1126 (accessed Jul. 11, 2023).

[18] S. Opiyo, J. Zhou, E. Mwangi, W. Kai, and I. Sunusi, “A Review on Teleoperation of Mobile Ground Robots: Architecture and Situation Awareness,” *Int. J. Control. Autom. Syst.*, vol. 19, no. 3, pp. 1384–1407, Mar. 2021, doi: 10.1007/S12555-019-0999-Z/METRICS.

[19] C. Breazeal, K. Dautenhahn, and T. Kanda, “Social Robotics,” *Springer Handbooks*, pp. 1935–1972, 2016, doi: 10.1007/978-3-319-32552-1_72/COVER.

[20] J. Troccaz, G. Dagnino, and G. Z. Yang, “Frontiers of Medical Robotics: From Concept to Systems to Clinical Translation,” *https://doi.org/10.1146/annurev-bioeng-060418-052502*, vol. 21, pp. 193–218, Jun. 2019, doi: 10.1146/ANNUREV-BIOENG-060418-052502.

[21] J. Broekens, M. Heerink, and H. Rosendal, “Assistive social robots in elderly care:

a review," *Gerontechnology*, vol. 8, no. 2, 2009, doi: 10.4017/gt.2009.08.02.002.00.

[22] B. Zhong and L. Xia, "A Systematic Review on Exploring the Potential of Educational Robotics in Mathematics Education," *Int. J. Sci. Math. Educ.*, vol. 18, no. 1, pp. 79–101, Jan. 2020, doi: 10.1007/S10763-018-09939-Y/TABLES/3.

[23] D. Alimisis alimisis, "Educational robotics: Open questions and new challenges," *Themes Sci. Technol. Educ.*, vol. 6, no. 1, pp. 63–71, Jun. 2013, Accessed: Jan. 05, 2023. [Online]. Available: http://earthlab.uoi.gr/theste/index.php/theste/article/view/119

[24] O. Mubin, C. J. Stevens, S. Shahid, A. Al Mahmud, and J.-J. Dong, "A REVIEW OF THE APPLICABILITY OF ROBOTS IN EDUCATION," *Technol. Educ. Learn.*, 2013, doi: 10.2316/Journal.209.2013.1.209-0015.

[25] I. Werry, K. Dautenhahn, B. Ogden, and W. Harwin, "Can social interaction skills be taught by a social agent? the role of a robotic mediator in autism therapy," in *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 2001, vol. 2117, pp. 57–74. doi: 10.1007/3-540-44617-6_6.

[26] T. Belpaeme, J. Kennedy, A. Ramachandran, B. Scassellati, and F. Tanaka, "Social robots for education: A review," *Science Robotics*, vol. 3, no. 21. American Association for the Advancement of Science, p. 5954, Aug. 2018. doi: 10.1126/scirobotics.aat5954.

[27] D. Scaradozzi, L. Screpanti, and L. Cesaretti, "Towards a Definition of Educational Robotics: A Classification of Tools, Experiences and Assessments," *Smart Learn. with Educ. Robot.*, pp. 63–92, 2019, doi: 10.1007/978-3-030-19913-5_3.

[28] V. Yrjönsuuri, K. Kangas, K. Hakkarainen, and P. Seitamaa-Hakkarainen, "The roles of material prototyping in collaborative design process at an elementary school," *Des. Technol. Educ.*, 2019.

[29] I. Gorakhnath and J. Padmanabhan, "Educational Robotics in Teaching Learning Process," 2017.

[30] S. Kucuk and B. Sisman, "Behavioral patterns of elementary students and teachers in one-to-one robotics instruction," *Comput. \& Educ.*, vol. 111, pp. 31–43, 2017.

[31] C. Kálózi-Szabó, K. Mohai, and M. Cottini, "Employing Robotics in Education to Enhance Cognitive Development—A Pilot Study," *Sustainability*, vol. 14, no. 23,

p. 15951, 2022.

[32] N. Eteokleous and E. Nisiforou, *Designing, Constructing, and Programming Robots for Learning*. IGI Global, 2021. [Online]. Available: https://books.google.gr/books?id=FqlQEAAAQBAJ

[33] A. Buss and R. Gamboa, “Teacher transformations in developing computational thinking: Gaming and robotics use in after-school settings,” *Emerg. Res. Pract. policy Comput. Think.*, pp. 189–203, 2017.

[34] A. Eguchi and L. Uribe, “Educational robotics meets inquiry-based learning: integrating inquiry-based learning into educational robotics,” in *Cases on inquiry through instructional technology in math and science*, IGI Global, 2012, pp. 327–366.

[35] A. Eguchi and L. Uribe, “Robotics to promote STEM learning: Educational robotics unit for 4th grade science,” in *2017 IEEE Integrated STEM Education Conference (ISEC)*, 2017, pp. 186–194. doi: 10.1109/ISECon.2017.7910240.

[36] F. Á. Bravo Sánchez, A. Forero Guzmán, and others, “La robótica como un recurso para facilitar el aprendizaje y desarrollo de competencias generales,” 2012.

[37] E. Ültanir, “An epistemological glance at the constructivist approach: Constructivist learning in Dewey, Piaget, and Montessori,” *Int. J. Instr.*, vol. 5, no. 2, 2012.

[38] L. S. forme avant 2007 Vygotskij and V. John-Steiner, *Mind in society: The development of higher psychological processes*. Harvard University Press, 1979.

[39] P. A. Ertmer and T. J. Newby, “Behaviorism, cognitivism, constructivism: Comparing critical features from an instructional design perspective,” *Perform. Improv. Q.*, vol. 6, no. 4, pp. 50–72, 1993.

[40] A. Eguchi, “Bringing robotics in classrooms,” in *Robotics in STEM Education: Redesigning the Learning Experience*, Springer International Publishing, 2017, pp. 3–31. doi: 10.1007/978-3-319-57786-9_1.

[41] J. G. Brooks and M. G. Brooks, *In search of understanding: The case for constructivist classrooms*. Ascd, 1999.

[42] S. Papert, *Mindstroms: Children , Computers , and Powerful Ideas and Powerful Ideas*. 1980. Accessed: Feb. 15, 2023. [Online]. Available: http://kvantti.kapsi.fi/Documents/LCL/mindstorms-chap1.pdf

[43] A. Rumjaun and F. Narod, "Social Learning Theory—Albert Bandura," *Sci. Educ. theory Pract. An Introd. Guid. to Learn. theory*, pp. 85–99, 2020.

[44] M. A. Weegar and D. Pacis, "A comparison of two theories of learning--behaviorism and constructivism as applied to face-to-face and online learning," 2012.

[45] B. F. Skinner, *Beyond freedom and dignity*. Hackett Publishing, 2002.

[46] M. J. Sutton, "Problem representation, understanding, and learning transfer implications for technology education," 2003.

[47] G. Owens, Y. Granader, A. Humphrey, and S. Baron-Cohen, "LEGO® therapy and the social use of language programme: An evaluation of two social skills interventions for children with high functioning autism and Asperger syndrome," *J. Autism Dev. Disord.*, vol. 38, no. 10, pp. 1944–1957, Nov. 2008, doi: 10.1007/s10803-008-0590-6.

[48] G. Nugent, B. Barker, N. Grandgenett, and V. Adamchuk, "The use of digital manipulatives in K-12: Robotics, GPS/GIS and programming," 2009. doi: 10.1109/FIE.2009.5350828.

[49] J. Sweller, "Cognitive load during problem solving: Effects on learning," *Cogn. Sci.*, vol. 12, no. 2, pp. 257–285, 1988.

[50] J. Sweller, J. J. G. Van Merrienboer, and F. G. W. C. Paas, "Cognitive architecture and instructional design," *Educ. Psychol. Rev.*, vol. 10, pp. 251–296, 1998.

[51] A. Otaran, O. Tokatli, and V. Patoglu, "Hands-on learning with a series elastic educational robot," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9775, pp. 3–16. doi: 10.1007/978-3-319-42324-1_1.

[52] C. Budiyanto, R. N. Fitriyaningsih, F. Kamal, R. Ariyuana, and A. Efendi, "Hands-on Learning in STEM: Revisiting Educational Robotics as a Learning Style Precursor," *Open Eng.*, vol. 10, no. 1, pp. 649–657, Jan. 2020, doi: 10.1515/ENG-2020-0071/MACHINEREADABLECITATION/RIS.

[53] Z. Dogmus, E. Erdem, and V. Patoglu, "ReAct!: An interactive educational tool for AI planning for robotics," *IEEE Trans. Educ.*, vol. 58, no. 1, pp. 15–24, 2015, doi: 10.1109/TE.2014.2318678.

[54] S. Atmatzidou, S. Demetriadis, and P. Nika, "How Does the Degree of Guidance

Support Students' Metacognitive and Problem Solving Skills in Educational Robotics?," *J. Sci. Educ. Technol.*, vol. 27, no. 1, pp. 70–85, Feb. 2018, doi: 10.1007/s10956-017-9709-x.

[55] F. Gratani, L. Giannandrea, A. Renieri, and M. Annessi, "Fostering Students' Problem-Solving Skills Through Educational Robotics in Primary School," in *Studies in Computational Intelligence*, 2021, vol. 982, pp. 3–14. doi: 10.1007/978-3-030-77022-8_1.

[56] A. Eguchi, "Educational Robotics for Promoting 21st Century Skills," *J. Autom. Mob. Robot. Intell. Syst.*, vol. 8, no. 1, pp. 5–11, 2014, doi: 10.14313/jamris_1-2014/1.

[57] R. Avello-Martínez, J. Lavonem, and M. Zapata-Ros, "Coding and educational robotics and their relationship with computational and creative thinking. A compressive review," *Revista de Educación a Distancia*, vol. 20, no. 63. 2020. doi: 10.6018/RED.413021.

[58] T. a Mikropoulos and I. Bellou, "Educational Robotics as Mindtools," *Themes Sci. Technol. Educ.*, vol. 6, no. 1, pp. 5–14, 2013, Accessed: May 07, 2023. [Online]. Available: http://earthlab.uoi.gr/ojs/theste/index.php/theste/article/view/114

[59] S. Anwar, N. A. Bascou, M. Menekse, and A. Kardgar, "A Systematic Review of Studies on Educational Robotics," *J. Pre-College Eng. Educ. Res.*, vol. 9, no. 2, p. 2, Jul. 2019, doi: 10.7771/2157-9288.1223.

[60] T. Ball and B. Zorn, "Teach foundational language principles," *Commun. ACM*, vol. 58, no. 5, pp. 30–31, 2015.

[61] M. Chevalier, C. Giang, A. Piatti, and F. Mondada, "Fostering computational thinking through educational robotics: a model for creative computational problem solving," *Int. J. STEM Educ.*, vol. 7, no. 1, pp. 1–18, Dec. 2020, doi: 10.1186/S40594-020-00238-Z/FIGURES/9.

[62] A. Khanlari, "Effects of robotics on 21st century skills," *Eur. Sci. J.*, vol. 9, no. 27, 2013.

[63] E. Van Laar, A. J. A. M. Van Deursen, J. A. G. M. Van Dijk, and J. de Haan, "Determinants of 21st-century skills and 21st-century digital skills for workers: A systematic literature review," *Sage Open*, vol. 10, no. 1, p. 2158244019900176,

2020.

[64] S. He, J. Zubarriain, and N. Kumia, "Integrating robotics education in pre-college engineering program," in *2015 IEEE Integrated STEM Education Conference*, 2015, pp. 183–188.

[65] C. Breazeal, P. L. Harris, D. Desteno, J. M. Kory Westlund, L. Dickens, and S. Jeong, "Young Children Treat Robots as Informants," *Top. Cogn. Sci.*, vol. 8, no. 2, pp. 481–491, Apr. 2016, doi: 10.1111/TOPS.12192.

[66] A. Khanlari, "Effects of educational robots on learning STEM and on students' attitude toward STEM," in *2013 IEEE 5th conference on engineering education (ICEED)*, 2013, pp. 62–66.

[67] L. Daniela and M. D. Lytras, "Educational robotics for inclusive education," *Technology, Knowledge and Learning*, vol. 24. Springer, pp. 219–225, 2019.

[68] E. Castro *et al.*, "Design and impact of a teacher training course, and attitude change concerning educational robotics," *Int. J. Soc. Robot.*, vol. 10, pp. 669–685, 2018.

[69] D. S. Fussy, H. Iddy, J. Amani, and S. T. Mkimbili, "Girls' participation in science education: structural limitations and sustainable alternatives," *Int. J. Sci. Educ.*, pp. 1–21, Mar. 2023, doi: 10.1080/09500693.2023.2188571.

[70] K. Wang, G. Y. Sang, L. Z. Huang, S. H. Li, and J. W. Guo, "The Effectiveness of Educational Robots in Improving Learning Outcomes: A Meta-Analysis," *Sustainability (Switzerland)*, vol. 15, no. 5. Multidisciplinary Digital Publishing Institute, p. 4637, Mar. 05, 2023. doi: 10.3390/su15054637.

[71] L. Nasi *et al.*, "Pomelo, a Collaborative Education Technology Interaction Robot," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2019, vol. 2019-March, pp. 757–758. doi: 10.1109/HRI.2019.8673160.

[72] A. Edwards, C. Edwards, P. R. Spence, C. Harris, and A. Gambino, "Robots in the classroom: Differences in students' perceptions of credibility and learning between 'teacher as robot' and 'robot as teacher,'" *Comput. Human Behav.*, vol. 65, pp. 627–634, 2016, doi: 10.1016/j.chb.2016.06.005.

[73] A. H. William, "Gender differences in learning style specific to science, technology, engineering and math--stem," 2007.

[74] J. J. Todd and E. Himburg, "Bringing robotics to life," *Classr. Robot. Case stories*

21st century Instr. Millenn. students, pp. 115–132, 2007.

[75] E. A. Pepitone, “Children in Cooperation and Competition,” *Learn. to Coop. Coop. to Learn*, pp. 17–65, 1985, doi: 10.1007/978-1-4899-3650-9_2.

[76] D. Topalli and N. E. Cagiltay, “Improving programming skills in engineering education through problem-based game projects with Scratch,” *Comput. Educ.*, vol. 120, pp. 64–74, 2018, doi: 10.1016/j.compedu.2018.01.011.

[77] M. Zizyte and T. Tabor, “Should Robotics Engineering Education Include Software Engineering Education?,” in *2022 IEEE/ACM 4th International Workshop on Robotics Software Engineering (RoSE)*, 2022, pp. 39–42.

[78] D. Siegle, “Technology: Encouraging creativity and problem solving through coding,” *Gift. Child Today*, vol. 40, no. 2, pp. 117–123, 2017.

[79] P. Griffin and E. Care, *Assessment and teaching of 21st century skills: Methods and approach*. Springer, 2014.

[80] R. Xia *et al.*, “The Beneficial Effect of Growth Mindset Intervention for Adolescents in Economically Disadvantaged Areas of China,” *Journal of Pacific Rim Psychology*. 2022. doi: 10.1177/18344909221142368.

[81] J. Thompson and G. Childers, “The impact of learning to code on elementary students’ writing skills,” *Comput. \& Educ.*, vol. 175, p. 104336, 2021.

[82] M. A. Bernardo and J. D. Morris, “Transfer effects of a high school computer programming course on mathematical modeling, procedural comprehension, and verbal problem solution,” *J. Res. Comput. Educ.*, vol. 26, no. 4, pp. 523–536, 1994.

[83] G. M. Kayg\is\iz, Ö. Üzümcü, and F. Melike Uçar, “The case of prospective teachers’ integration of coding-robotics practices into science teaching with STEM approach,” 2020.

[84] D. R. Mullet, T. Kettler, and A. Sabatini, “Gifted Students’ Conceptions of Their High School STEM Education,” *J. Educ. Gift.*, vol. 41, no. 1, pp. 60–92, 2018, doi: 10.1177/0162353217745156.

[85] D. R. Dounas-Frazer, L. R\'\ios, and H. J. Lewandowski, “Preliminary model for student ownership of projects,” *arXiv Prepr. arXiv1907.01690*, 2019.

[86] C. C. Kuhlthau, L. K. Maniotes, and A. K. Caspari, *Guided inquiry: Learning in the 21st century: Learning in the 21st century*. Abc-Clio, 2015.

[87] G. Falloon, "An Analysis of Young Students' Thinking When Completing Basic Coding Tasks Using Scratch Jnr. On the iPad," *Journal of Computer Assisted Learning*. 2016. doi: 10.1111/jcal.12155.

[88] "Turtle Academy." https://turtleacademy.com/

[89] D. Weintrop, "Block-based programming in computer science education," *Commun. ACM*, vol. 62, no. 8, pp. 22–25, 2019.

[90] D. Fields, V. Vasudevan, and Y. B. Kafai, "The programmers' collective: Connecting collaboration and computation in a high school scratch mashup coding workshop," in *Proceedings of International Conference of the Learning Sciences, ICLS*, 2014, vol. 2, no. January, pp. 855–862.

[91] S. Krouse, "Scratch Has a Marketing Problem," 2016. https://www.freecodecamp.org/news/scratch-has-a-marketing-problem-f84626bd18ef/ (accessed Jul. 22, 2023).

[92] Google Inc., "Blockly | Google Developers," *https://developers.google.com/blockly*, 2017. https://developers.google.com/blockly

[93] P. D. P. Adi and A. Kitagawa, "A review of the Blockly programming on M5Stack board and MQTT based for programming education," in *2019 IEEE 11th International Conference on Engineering Education (ICEED)*, 2019, pp. 102–107.

[94] S. Peta, "Python- An Appetite for the Software Industry," *International Journal of Programming Languages and Applications*. 2022. doi: 10.5121/ijpla.2022.12401.

[95] D. Chicco and R. Shiradkar, "Ten Quick Tips for Computational Analysis of Medical Images," *Plos Computational Biology*. 2023. doi: 10.1371/journal.pcbi.1010778.

[96] F. Perez, B. E. Granger, and J. D. Hunter, "Python: an ecosystem for scientific computing," *Comput. Sci. \& Eng.*, vol. 13, no. 2, pp. 13–21, 2010.

[97] R. Paffenroth and X. Kong, "Python in data science research and education," in *PROC. OF THE 14th PYTHON IN SCIENCE CONF*, 2015, pp. 164–170.

[98] P. F. Dubois, "Guest Editor's Introduction: Python: Batteries Included," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 7–9, 2007, doi: 10.1109/MCSE.2007.51.

[99] J. Helminen and L. Malmi, "Jype-a program visualization and programming exercise tool for Python," in *Proceedings of the 5th international symposium on*

Software visualization, 2010, pp. 153–162.

[100] S. Wang, D.-Y. Liu, N. Wang, and Y.-X. Yuan, "Design and Implementation of an Online Python Teaching Case Library for the Training of Application-Oriented Talents," *International Journal of Emerging Technologies in Learning (Ijet)*. 2020. doi: 10.3991/ijet.v15i21.18191.

[101] Y. Govilkar, "A Gesture Based Home Automation System," *International Journal for Research in Applied Science and Engineering Technology*. 2019. doi: 10.22214/ijraset.2019.11081.

[102] E. Mészárosová, "Is python an appropriate programming language for teaching programming in secondary schools," *Int. J. Inf. Commun. Technol. Educ.*, vol. 4, no. 2, pp. 5–14, 2015.

[103] M. B. J. Fagan and B. R. Payne, "Learning to Program in Python – By Teaching It!," *Proceedings of the Interdisciplinary Stem Teaching and Learning Conference*. 2017. doi: 10.20429/stem.2017.010109.

[104] P. J. Guo, "Online Python Tutor: Embeddable Web-Based Program Visualization for Cs Education," in *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 2013, pp. 579–584. doi: 10.1145/2445196.2445368.

[105] L. Liu, L. Wanwu, H. Zhang, and J. Li, "Key technologies and case implementation of online course," *Comput. Appl. Eng. Educ.*, vol. 31, no. 1, pp. 200–215, 2023.

[106] D. Castillo-Merino and E. Serradell-López, "An analysis of the determinants of students' performance in e-learning," *Comput. Human Behav.*, vol. 30, pp. 476–484, 2014.

[107] E. Marlina, B. Tjahjadi, and S. Ningsih, "Factors affecting student performance in e-learning: A case study of higher educational institutions in Indonesia," *J. Asian Financ. Econ. Bus.*, vol. 8, no. 4, pp. 993–1001, 2021.

[108] H. M. W. Rasheed, Y. He, J. Khalid, H. M. U. Khizar, and S. Sharif, "The relationship between e-learning and academic performance of students," *J. Public Aff.*, vol. 22, no. 3, p. e2492, 2022.

[109] O. Deperlioglu and U. Kose, "The effectiveness and experiences of blended learning approaches to computer programming education," *Comput. Appl. Eng. Educ.*, vol. 21, no. 2, pp. 328–342, 2013.

[110] X. Zeng *et al.*, “The construction and online/offline blended learning of small private online courses of Principles of Chemical Engineering,” *Comput. Appl. Eng. Educ.*, vol. 26, no. 5, pp. 1519–1526, 2018.

[111] O. V. Berestok, “Technological and Methodological Support of Blended Learning in Higher Education in The Covid-19 Pandemic,” 2022.

[112] O. Dallas and A. Gogoulou, “Learning Programming Using Python: The Case of the DigiWorld Educational Game,” *Eur. J. Eng. Technol. Res.*, no. CIE, pp. 1–8, 2022, doi: 10.24018/ejeng.2021.0.CIE.2750.

[113] S. Deterding, R. Khaled, L. E. Nacke, D. Dixon, and others, “Gamification: Toward a definition,” in *CHI 2011 gamification workshop proceedings*, 2011, vol. 12, pp. 1–79.

[114] E. Uaidullakyzy, R. Gulnara, S. Khalima, B. Zeinep, R. Turmanov, and G. Rysbayeva, “Creating Integration Situations of Students’ Computer Lesson and Learning With Gamification,” *International Journal of Emerging Technologies in Learning (Ijet)*. 2022. doi: 10.3991/ijet.v17i19.32177.

[115] S. Bigdeli *et al.*, “Underpinning Learning Theories of Medical Educational Games: A Scoping Review,” *Med. J. Islam. Repub. Iran*, vol. 37, 2023.

[116] F. F.-H. Nah, Q. Zeng, V. R. Telaprolu, A. P. Ayyappa, and B. Eschenbrenner, “Gamification of education: a review of literature,” in *HCI in Business: First International Conference, HCIB 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings 1*, 2014, pp. 401–409.

[117] D. Sudana, A. W. R. Emanuel, Suyoto, and A. S. Fornia, “Applying Social-Gamification for Interactive Learning in Tuberculosis Education,” *International Journal of Advanced Computer Science and Applications*. 2020. doi: 10.14569/ijacsa.2020.0110342.

[118] S. Schefer-Wenzl and I. Miladinovic, “Teaching software engineering with gamification elements,” *Int. J. Adv. Corp. Learn.*, vol. 11, no. 1, p. 48, 2018.

[119] A. H. Nabizadeh, J. Jorge, S. Gama, and D. Gonçalves, “How Do Students Behave in a Gamified Course?—A Ten-Year Study,” *IEEE Access*, vol. 9, pp. 81008–81031, 2021.

[120] D. Dicheva, C. Dichev, G. Agre, and G. Angelova, “Gamification in education: A

systematic mapping study," *J. Educ. Technol. \& Soc.*, vol. 18, no. 3, pp. 75–88, 2015.

[121] C. Dichev, D. Dicheva, and K. Irwin, "Gamifying learning for learners," *Int. J. Educ. Technol. High. Educ.*, vol. 17, pp. 1–14, 2020.

[122] M. Prensky, "Digital game-based learning," *Comput. Entertain.*, vol. 1, no. 1, p. 21, 2003.

[123] H.-Y. Sung and G.-J. Hwang, "A collaborative game-based learning approach to improving students' learning performance in science courses," *Comput. Educ.*, vol. 63, pp. 43–51, 2013, doi: https://doi.org/10.1016/j.compedu.2012.11.019.

[124] M. H. Hussein, S. H. Ow, L. S. Cheong, and M.-K. Thong, "A Digital Game-Based Learning Method to Improve Students' Critical Thinking Skills in Elementary Science," *IEEE Access*, vol. 7, pp. 96309–96318, 2019, doi: 10.1109/ACCESS.2019.2929089.

[125] S. De Freitas, "Learning in immersive worlds: A review of game-based learning," 2006.

[126] J. Lean, J. Moizer, C. Derham, L. Strachan, and Z. Bhuiyan, "Real world learning: simulation and gaming," *Appl. Pedagog. High. Educ. Real World Learn. Innov. across Curric.*, pp. 187–214, 2021.

[127] J. L. Plass, B. D. Homer, and C. K. Kinzer, "Foundations of Game-Based Learning," *Educ. Psychol.*, vol. 50, no. 4, pp. 258–283, 2015, doi: 10.1080/00461520.2015.1122533.

[128] F. J. Agbo, S. S. Oyelere, J. Suhonen, and T. H. Laine, "Co-design of mini games for learning computational thinking in an online environment," *Educ. Inf. Technol.*, vol. 26, no. 5, pp. 5815–5849, 2021, doi: 10.1007/s10639-021-10515-1.

[129] O. Shabalina, N. Sadovnikova, and A. Kravets, "Methodology of teaching software engineering: Game-based learning cycle," in *2013 3rd Eastern European Regional Conference on the Engineering of Computer Based Systems*, 2013, pp. 113–119.

[130] D. Vlachopoulos and A. Makri, "The effect of games and simulations on higher education: a systematic literature review," *Int. J. Educ. Technol. High. Educ.*, vol. 14, no. 1, pp. 1–33, 2017.

[131] D. Bart, A.C., Gusukuma, L., Kafur, "BlockPy Editor," 2019.

[132] A. C. Bart, C. A. Shaffer, D. Kafura, R. Whitcomb, and E. Tilevich, "Computing with CORGIS: Diverse, real-world datasets for introductory computing," in *ACM Inroads*, Jun. 2017, vol. 8, no. 2, pp. 66–72. doi: 10.1145/3095781.3017708.

[133] A. C. Bart, J. Tibau, D. Kafura, C. A. Shaffer, and E. Tilevich, "Design and Evaluation of a Block-based Environment with a Data Science Context," *IEEE Trans. Emerg. Top. Comput.*, vol. 8, no. 1, pp. 182–192, Jan. 2020, doi: 10.1109/TETC.2017.2729585.

[134] A. Roberge, "Reeborg's world," *https://reeborg.ca/index_en.html*, 2020.

[135] A. Roberge, "RUR-PLE," *http://rur-ple.sourceforge.net/*, 2005.

[136] H. Imabayashi, "Sokoban Official Site," *https://www.sokoban.jp/*, 1982. https://www.sokoban.jp/

[137] CodeCombat Inc, "CodeCombat - Coding games to learn Python and JavaScript | CodeCombat," *https://codecombat.com/*, 2019. https://codecombat.com/

[138] Amazon Web Services, "Code.org - Learn Computer Science," *https://studio.code.org/courses*, 2020. https://studio.code.org/courses

[139] The LEGO group, "MINDSTORMS EV3 Support | Everything You Need | LEGO® Education," *https://education.lego.com/en-us/product-resources/mindstorms-ev3/teacher-resources/python-for-ev3*, 2019. https://education.lego.com/en-us/product-resources/mindstorms-ev3/teacher-resources/python-for-ev3

[140] L. Benotti, M. J. Gómez, and C. Martínez, "UNC++Duino: A kit for learning to program robots in python and C++ starting from blocks," in *Advances in Intelligent Systems and Computing*, 2017, vol. 457, pp. 181–192. doi: 10.1007/978-3-319-42975-5_17.

[141] E. Susilo *et al.*, "STORMLab for STEM Education: An Affordable Modular Robotic Kit for Integrated Science, Technology, Engineering, and Math Education," *IEEE Robot. Autom. Mag.*, vol. 23, no. 2, pp. 47–55, Jun. 2016, doi: 10.1109/MRA.2016.2546703.

[142] M. Khamphroo, N. Kwankeo, K. Kaemarungsi, and K. Fukawa, "MicroPython-based educational mobile robot for computer coding learning," Jun. 2017. doi: 10.1109/ICTEmSys.2017.7958781.

[143] M. Khamphroo, N. Kwankeo, K. Kaemarungsi, and K. Fukawa, "Integrating

MicroPython-based educational mobile robot with wireless network," in *2017 9th International Conference on Information Technology and Electrical Engineering, ICITEE 2017*, Jul. 2017, vol. 2018-Janua, pp. 1–6. doi: 10.1109/ICITEED.2017.8250449.

[144] I. Bakali *et al.*, "Control a robot via internet using a block programming platform for educational purposes," in *ACM International Conference Proceeding Series*, Jul. 2018, pp. 1–2. doi: 10.1145/3200947.3201063.

[145] P. Skoupras *et al.*, "El greco: A 3d-printed humanoid that anybody can afford," in *ACM International Conference Proceeding Series*, Jul. 2018, pp. 1–2. doi: 10.1145/3200947.3201062.

[146] H. Brenton, M. Gillies, D. Ballin, and D. Chatting, "The Uncanny Valley: does it exist and is it related to presence," *Presence Connect (www. presence- …*, 2005, [Online]. Available: http://scholar.google.co.uk/scholar?hl=en&q=author%3Achatting&btnG=Search&as_sdt=2000&as_ylo=&as_vis=0#4

[147] J. Taufatofua *et al.*, "Designing for Robust Movement in a Child-Friendly Robot," in *IEEE International Conference on Intelligent Robots and Systems*, Dec. 2018, pp. 7667–7674. doi: 10.1109/IROS.2018.8593414.

[148] I. Doroftei, F. Adascalitei, D. Lefeber, B. Vanderborght, and I. A. Doroftei, "Facial expressions recognition with an emotion expressive robotic head," in *IOP Conference Series: Materials Science and Engineering*, Sep. 2016, vol. 147, no. 1, p. 012086. doi: 10.1088/1757-899X/147/1/012086.

[149] Team Hector, "Johnny - Team Hector," *https://www.teamhector.de/robot/johnny*, 2014.

[150] A. Stentz *et al.*, "CHIMP, the CMU Highly Intelligent Mobile Platform," *J. F. Robot.*, vol. 32, no. 2, pp. 209–228, Mar. 2015, doi: 10.1002/rob.21569.

[151] B. Gary, "The OpenCV Library," *Dr. Dobb's J. Softw. Tools*, vol. 25, no. 2236121, pp. 120–123, 2008, [Online]. Available: https://www.elibrary.ru/item.asp?id=4934581

[152] Google inc., "Speech-to-Text: Automatic Speech Recognition | Google Cloud," *https://cloud.google.com/speech-to-text*, 2021. https://cloud.google.com/speech-to-

text

[153] Python Speech Recognition, “SpeechRecognition · PyPI,” *Pypi.org https://pypi.org/project/SpeechRecognition/*, 2019.

[154] Google Cloud, “Text-to-Speech: Lifelike Speech Synthesis,” *https://cloud.google.com/text-to-speech*, 2013. https://cloud.google.com/text-to-speech

[155] Pierre Nicolas Durette, “gTTS · PyPI,” *Pypi.Org https://pypi.org/project/gTTS/*. 2020.

[156] S. Vollmer, “Google translate,” in *Figures of Interpretation*, 2021, pp. 72–76. doi: 10.1007/s12490-017-0013-7.

[157] S. Behnel, “lxml - Processing XML and HTML with Python,” 2016.

[158] M. Pilgrim and K. McKee, “Feedparser: Parse Atom and RSS feeds in Python,” *https://pypi.org/project/feedparser/*, 2015. https://pypi.org/project/feedparser/

[159] I. G. Triandini, Evi; Suardika, “Installing, configuring, and developing with Xampp,” *D. Dvorski Dalibor*, no. March, pp. 1–10, 2007, Accessed: Nov. 09, 2022. [Online]. Available: http://www.apachefriends.org/.

[160] Softvelum LLC, “Softvelum Nimble Streamer: freeware media server for live and VOD streaming,” *https://wmspanel.com/nimble*. https://wmspanel.com/nimble

[161] Jim and O. S. Contributors, “Open Broadcaster Software | OBS,” *Open Broadcaster Software https://obsproject.com/*, 2020. https://obsproject.com/

[162] Softvelum LLC, “Softvelum Low Delay Protocol - low latency streaming protocols - Softvelum,” *https://softvelum.com/sldp/*. https://softvelum.com/sldp/

[163] S. LLC, “HTML5 player for SLDP - Softvelum,” *https://softvelum.com/player/web/*. https://softvelum.com/player/web/

[164] PHPMailer Contributors, “GitHub - PHPMailer/PHPMailer: The classic email sending library for PHP,” *https://github.com/PHPMailer/PHPMailer*. https://github.com/PHPMailer/PHPMailer

[165] J. Wigginton, P. Monnerat, A. Fischer, H. J. Petrich, and C. Campbell, “phpseclib: pure PHP implementations of SSH, SFTP and RSA.”

[166] M. Haverbeke, “CodeMirror,” *https://codemirror.net/*.

[167] J. Naidoo and S. Hajaree, “Exploring the perceptions of Grade 5 learners about the

use of videos and PowerPoint presentations when learning fractions in mathematics," *South African J. Child. Educ.*, vol. 11, no. 1, p. 846, 2021.

[168] A. Hershkovitz, G. Ben-Zadok, R. Mintz, and R. Nachmias, "Examining online learning processes based on log files analysis: A case study Need to cite this paper? Want more papers like this? Examining online learning processes based on log files analysis: A case study," 2009.

[169] M. H. Cho and J. S. Yoo, "Exploring online students' self-regulated learning with self-reported surveys and log files: a data mining approach," *Interact. Learn. Environ.*, vol. 25, no. 8, pp. 970–982, Nov. 2017, doi: 10.1080/10494820.2016.1232278.

[170] N. Fraser, "JS-Interpreter Documentation," *https://neil.fraser.name/software/JS-Interpreter/docs.html*.

[171] G. Developers, "Generating and Running JavaScript | Blockly | Google Developers," *https://developers.google.com/blockly/guides/app-integration/running-javascript*. https://developers.google.com/blockly/guides/app-integration/running-javascript (accessed Apr. 27, 2022).

[172] L. Jiang and Z. Su, "Automatic mining of functionally equivalent code fragments via random testing," in *Proceedings of the 18th International Symposium on Software Testing and Analysis, ISSTA 2009*, Jul. 2009, pp. 81–91. doi: 10.1145/1572272.1572283.

[173] N. Zeybek and E. Saygı, "Gamification in Education: Why, Where, When, and How?—A Systematic Review," *Games Cult.*, p. 155541202311586, Mar. 2023, doi: 10.1177/15554120231158625.

[174] G. Barata, S. Gama, J. Jorge, and D. Gonçalves, "Engaging engineering students with gamification," in *2013 5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, 2013, pp. 1–8.

[175] C. Paciarotti, G. Bertozzi, and M. Sillaots, "A new approach to Gamification in engineering education: the Learner-Designer Approach to Serious Games," *Eur. J. Eng. Educ.*, vol. 46, no. 6, pp. 1092–1116, 2021, doi: 10.1080/03043797.2021.1997922.

[176] "Institute for the Integration of Technology into Teaching and Learning," *University*

of North Texas, 2014. https://iittl.unt.edu/content/computer-attitude-questionnaire-caq (accessed Aug. 07, 2022).

[177] L. Leifheit *et al.*, “SCAPA: Development of a Questionnaire Assessing Self-Concept and Attitudes Toward Programming,” *Annu. Conf. Innov. Technol. Comput. Sci. Educ. ITiCSE*, pp. 138–144, Jun. 2020, doi: 10.1145/3341525.3387415.

[178] “Robotics and GPS/GIS Interest Questionnaire (4-H) | Assessment Tools in Informal STEM (ATIS).” http://www.pearweb.org/atis/tools/31 (accessed Aug. 08, 2022).

[179] R. W. Jones, “Learning and teaching in small groups: Characteristics, benefits, problems and approaches,” *Anaesth. Intensive Care*, vol. 35, no. 4, pp. 587–592, Jan. 2007, doi: 10.1177/0310057x0703500420.

[180] J. C.-C. G. Journal and undefined 2010, “Self-Direction: A Critical Tool in Distance Learning.,” *commongroundjournal.org*, 2010, Accessed: Aug. 08, 2022. [Online]. Available: http://www.commongroundjournal.org/volnum/v07n02.pdf#page=89

[181] A. El-Fakdi and X. Cuf\'\i, “An Innovative Low Cost Educational Underwater Robotics Platform for Promoting Engineering Interest among Secondary School Students,” *Electronics*, vol. 11, no. 3, p. 373, 2022.

[182] G. Bampasidis, D. Piperidis, V. C. Papakonstantinou, D. Stathopoulos, C. Troumpetari, and P. Poutos, “Hydrobots, an Underwater Robotics STEM Project: Introduction of Engineering Design Process in Secondary Education.,” *Adv. Eng. Educ.*, 2021.

[183] S. Ziaeefard, G. A. Ribeiro, and N. Mahmoudian, “GUPPIE, underwater 3D printed robot a game changer in control design education,” *Proc. Am. Control Conf.*, vol. 2015-July, pp. 2789–2794, Jul. 2015, doi: 10.1109/ACC.2015.7171157.

[184] M. Vasileiou, N. Manos, and E. Kavallieratou, “A low-cost 3D printed mini underwater vehicle: Design and Fabrication,” in *2021 20th International Conference on Advanced Robotics (ICAR)*, 2021, pp. 390–395.

[185] M. Vasileiou, N. Manos, N. Vasilopoulos, A. Douma, and E. Kavallieratou, “kalypso Autonomous Underwater Vehicle: A 3D-Printed Underwater Vehicle for Inspection at Fisheries,” *J. Mech. Robot.*, vol. 16, no. 4, May 2023, doi:

10.1115/1.4062355.

[186] A. K. Gottschall, “Design of an underwater glider for education and research,” in *2012 IEEE/OES Autonomous Underwater Vehicles (AUV)*, 2012, pp. 1–4.

[187] K. Watanabe, “An AUV based experimental system for the underwater technology education,” in *OCEANS 2006-Asia Pacific*, 2006, pp. 1–7.

[188] Y. Serhat, “Development stages of a semi-autonomous underwater vehicle experiment platform,” *Int. J. Adv. Robot. Syst.*, vol. 19, no. 3, p. 17298806221103710, 2022.

Apprendix

1. User Profile Questionnaire for the El Greco Platform survey

User Profile Questionnaire

This questionnaire seeks to evaluate views on issues concerning:
1. The use and utility of the computer,
2. Computer Programming,
3. Robotics.

The questionnaire consists of a total of three (3) parts. Filling in the questionnaire is done anonymously. The answers you will give are confidential and will not be used as a criterion for your evaluation.

1. Demographics

Sex *

- ○ Man
- ○ Woman

Age *

Grade *

- ○ A Gymnasium
- ○ B Gymnasium
- ○ C Gymnasium
- ○ A Lyceum
- ○ B Lyceum
- ○ C Lyceum

2. Views on computer use

How many years have you been using a computer? *

How many hours a week do you use a computer? *

For each of the adjacent statements circle the number that represents your response according to the scale: 1= "Strongly Disagree", 2= "Disagree", 3= "Neither Agree nor Disagree", 4= "Agree", 5= " Strongly Agree". *

	Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
I like using a computer.	○	○	○	○	○
I get tired of using a computer.	○	○	○	○	○
I feel comfortable using a computer.	○	○	○	○	○
Using a computer presents great difficulties.	○	○	○	○	○
I really like playing computer games	○	○	○	○	○
I can learn using a computer.	○	○	○	○	○
I like computer lessons more than the classroom.	○	○	○	○	○

3. Views on Programming

For each of the adjacent statements circle the number that represents your response according to the scale: 1= “Strongly Disagree”, 2= “Disagree”, 3= “Neither Agree nor Disagree”, 4= “Agree”, 5= “ Strongly Agree”. *

	Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
I can explain what computer programming means.	○	○	○	○	○
I have been taught programming.	○	○	○	○	○
I can program a computer.	○	○	○	○	○
Programming is tiresome	○	○	○	○	○
Programming scares me	○	○	○	○	○
I like programming	○	○	○	○	○
Programming lessons are enjoyable.	○	○	○	○	○
It's important to me to be good in programming	○	○	○	○	○
Computer programming will help me in my daily life.	○	○	○	○	○
Computer programming will help me find a job after I graduate from school/university.	○	○	○	○	○
Computer programming will be useful in my school/university performance.	○	○	○	○	○

In order to get good at programming I will need to be less involved in other activities.	○	○	○	○	○
I want to make an effort to be good in programming	○	○	○	○	○
When programming gets difficult I give up.	○	○	○	○	○

4. Views on Robotics

What do you like in robotics?

What you do not like in robotics?

For each of the adjacent statements circle the number that represents your response according to the scale: 1= "Strongly Disagree", 2= "Disagree", 3= "Neither Agree nor Disagree", 4= "Agree", 5= " Strongly Agree". *

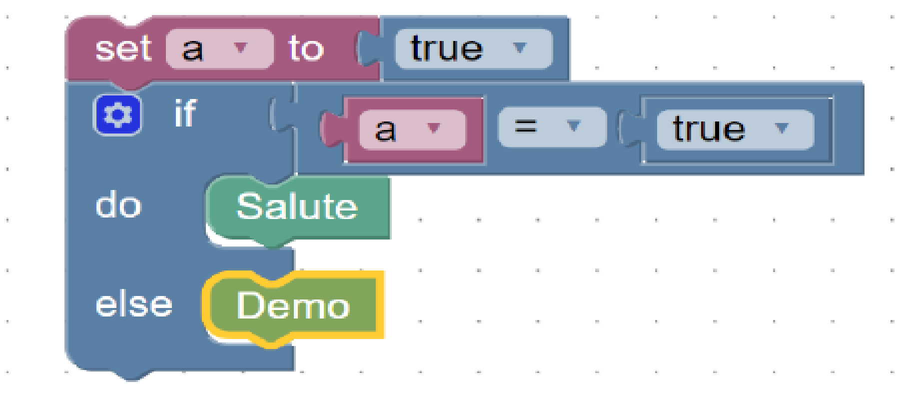
	ΔιαφωνώΑπόλυτα	Διαφωνώ	Ούτε Συμφωνώ Ούτε Διαφωνώ	Συμφωνώ	Συμφωνώ Απόλυτα
I like learning about new technologies like Robotics	○	○	○	○	○
It's important to me to learn Robotics	○	○	○	○	○
It's important to me to learn to program a robot	○	○	○	○	○

It's important to me to learn the basic principles of Robotics	○	○	○	○	○
I am certain that i can program a robot to move forward and stop	○	○	○	○	○
I am certain that i can fix the software of a robot that malfunctions	○	○	○	○	○
I think i can solve practical issues using Robotics and programming	○	○	○	○	○
I think robotics and programming will help to lead a successful life	○	○	○	○	○
I want to find a job in robotics and programming	○	○	○	○	○
I have heard of Python Programming language	○	○	○	○	○
I can program in Python I have heard of Blockly	○	○	○	○	○
I can program in Blockly	○	○	○	○	○
I can program in Blockly	○	○	○	○	○

If the Blockly code in the photo below is executed. What will El Greco do? *

- It will move forward.
- It will turn left.
- it will turn right.
- I do not know

If the Blockly code in the photo below is executed. What will El Greco do? *



- It will move forward.
- It will salute.
- It will demonstrate it skills (Demo).
- I do not know.

*

If the Python code in the picture below is executed. What will El Greco do

```
a = 1
b = 0
if a < b:
    from go_forward_seconds import go_forward_seconds
    go_forward_seconds(-50,50,2)
else:
    from go_forward_seconds import go_forward_seconds
    go_forward_seconds(50,-50,4)
```

- ◯ It will move forward.
- ◯ It will turn left.
- ◯ It will turn right.
- ◯ I do not know.

*

If the Python code in the picture below is executed. What will El Greco do

```
a = True
if a == True:
    from salute import salute
    salute()
else:
    from demo import demo
    demo()
```

- ◯ It will move forward
- ◯ It will salute.
- ◯ It will demonstrate it skills (Demo).
- ◯ I do not know.

2. Questionnaire after using the El Greco Platform

Questionnaire about the El Greco programming learning platform

This questionnaire seek to evaluate views on the El Greco programming learning platform and programming in robotics. The questionnaire consists of 3 sections. Filling in the questionnaire is done anonymously. The answers you will give are confidential and will not be used as a criterion for your evaluation.

1. Views on Robotics and Programming

Here it is evaluated if after using the platform you changed your views on topics related to robotics and programming

For each of the adjacent statements circle the number that represents your response according to the scale: 1= "Strongly Disagree", 2= "Disagree", 3= "Neither Agree nor Disagree", 4="Agree", 5= " Strongly Agree" *

	Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
I like learning about new technologies like Robotics	◯	◯	◯	◯	◯
It's important to me to learn Robotics	◯	◯	◯	◯	◯
It's important to me to learn the basic principles of Robotics	◯	◯	◯	◯	◯
It's important to me to learn to program a robot	◯	◯	◯	◯	◯
I am certain that i can program a robot to move forward and stop	◯	◯	◯	◯	◯
I am certain that i can fix the software of a robot that malfunctions	◯	◯	◯	◯	◯

I think i can solve practical issues using Robotics and programming	○	○	○	○	○
I think robotics and programming will help to lead a successful life	○	○	○	○	○
I want to find a job in robotics and programming	○	○	○	○	○
Programming is tiresome	○	○	○	○	○
Programming scares me	○	○	○	○	○
I like programming	○	○	○	○	○
It's important to me to be good in programming	○	○	○	○	○
I want to make an effort to be good in programming	○	○	○	○	○

2. View on El Greco Platform

For each of the adjacent statements circle the number that represents your response according to the scale: 1= "Strongly Disagree", 2= "Disagree", 3= "Neither Agree nor Disagree", 4="Agree", 5= " Strongly Agree" *

	Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
I think the handling of the platform is easy.	○	○	○	○	○

I can easily explain to others how the platform works.	○	○	○	○	○
I liked El Greco Adventures.	○	○	○	○	○
I would like to complete El Greco Adventures.	○	○	○	○	○
It helped me understand the science of robotics better.	○	○	○	○	○
It helped me to understand the concept of programming better.	○	○	○	○	○
It made me think about what I will do in terms of work after I graduate from school/university.	○	○	○	○	○
It made me feel like I can perform better in programming and robotics.	○	○	○	○	○
It made me want to use platforms similar to EL Greco's.	○	○	○	○	○
Using the platform encourages me to learn programming and robotics.	○	○	○	○	○
Using the platform helps me learn robotics and programming more efficiently.	○	○	○	○	○

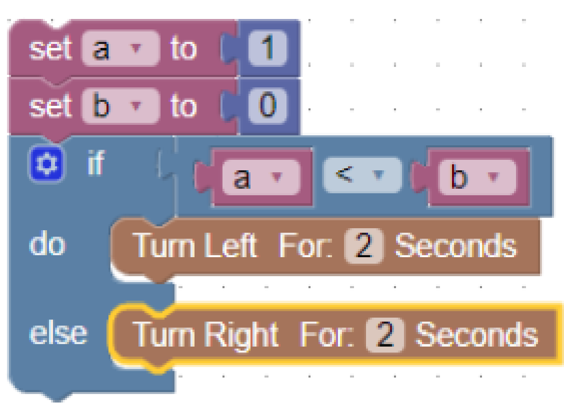
I believe I can gain knowledge in robotics and programming.

I want to continue learning about robotics and programming.

What did you like about the El Greco's platform?

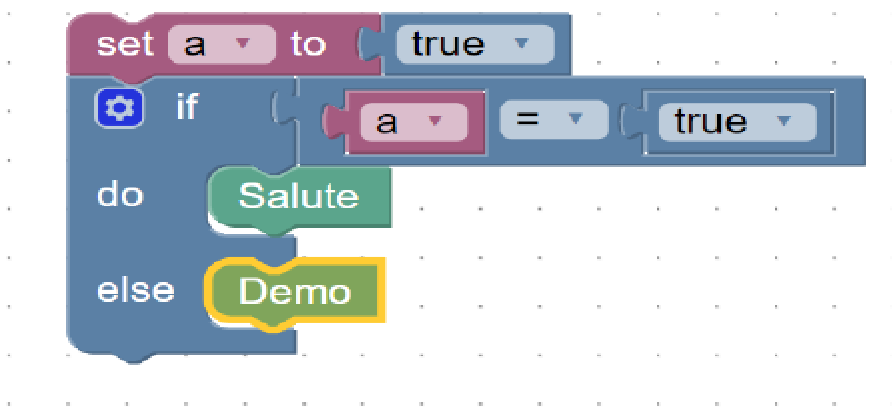
What didn't you like about the El Greco's platform?

If the Blockly code in the photo below is executed. What will El Greco do? *



- It will move forward.
- It will turn left.
- It will turn right.
- I do not know.

If the Blockly code in the photo below is executed. What will El Greco do? *



- ◯ It will move forward.
- ◯ It will salute.
- ◯ It will demonstrate it skills (Demo).
- ◯ I do not know.

If the Python code in the picture below is executed. What will El Greco do? *

```
a = 1
b = 0
if a < b:
    from go_forward_seconds import go_forward_seconds
    go_forward_seconds(-50,50,2)
else:
    from go_forward_seconds import go_forward_seconds
    go_forward_seconds(50,-50,4)
```

- ◯ It will move forward.
- ◯ It will turn left.
- ◯ It will turn right.
- ◯ I do not know.

*

If the Python code in the picture below is executed. What will El Greco do?

```python

a = True
if a == True:
    from salute import salute
    salute()
else:
    from demo import demo
    demo()

```

- ◯ It will move forward.
- ◯ It will salute.
- ◯ It will demonstrate it skills (Demo).
- ◯ I do not know.

3. User profile Questionnaire for the EDUV Platform survey

User Profile Questionnaire

This questionnaire seeks to evaluate views on issues concerning:
1. The use and utility of the computer,
2. Computer Programming,
3. Robotics.

The questionnaire consists of a total of three (3) parts. Filling in the questionnaire is done anonymously. The answers you will give are confidential and will not be used as a criterion for your evaluation.

1. Demographics

Sex *

- ◯ Man
- ◯ Woman

Age *

Grade *

- ◯ A Gymnasium
- ◯ B Gymnasium
- ◯ C Gymnasium
- ◯ A Lyceum
- ◯ B Lyceum
- ◯ C Lyceum

2. Computer knowledge/views

How many years have you been using a computer? *

How many hours a week do you use a computer? *

For each of the adjacent statements circle the number that represents your answer according to the scale: 1 = "Strongly Disagree", 2 = "Disagree", 3 = "Neither Agree nor Disagree", 4 = "Agree", 5 = "Strongly Agree". *

	1	2	3	4	5
1. I like using a computer.	○	○	○	○	○
2. I get tired of using a computer.	○	○	○	○	○
3. I find it difficult to use a compute	○	○	○	○	○
4. I like playing computer games	○	○	○	○	○
5. I learn using a computer	○	○	○	○	○
6. I prefer online classes over in-person classes	○	○	○	○	○

3. Knowledge related to programming

For each of the adjacent statements circle the number that represents your answer according to the scale: 1 = “Strongly Disagree”, 2 = “Disagree”, 3 = “Neither Agree nor Disagree”, 4 = “Agree”, 5 = “Strongly Agree”. *

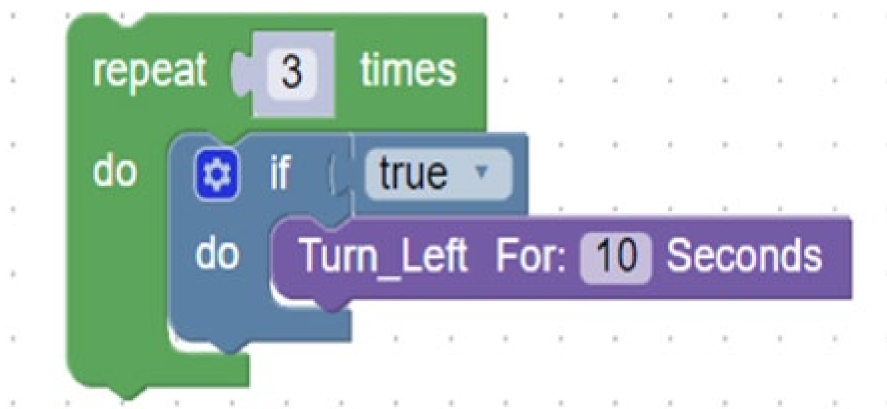
	1	2	3	4	5
1. I can explain what computer programming means.	○	○	○	○	○
2. I have been taught programming.	○	○	○	○	○
3. I can program in some programming language	○	○	○	○	○
4. I like programming.	○	○	○	○	○
5. Programming lessons are enjoyable.	○	○	○	○	○
6. Computer programming will help me in my daily life.	○	○	○	○	○
7. Computer programming will help me find a job after I graduate from school/university.	○	○	○	○	○
8. To get good at programming I will need to do less of other activities.	○	○	○	○	○
9. I want to make an effort to learn programming.	○	○	○	○	○
10. When programming becomes difficult I give up. I give up.	○	○	○	○	○

4. Views on Robotics

For each of the adjacent statements circle the number that represents your answer according to the scale: 1 = “Strongly Disagree”, 2 = “Disagree”, 3 = “Neither Agree nor Disagree”, 4 = “Agree”, 5 = “Strongly Agree”. *

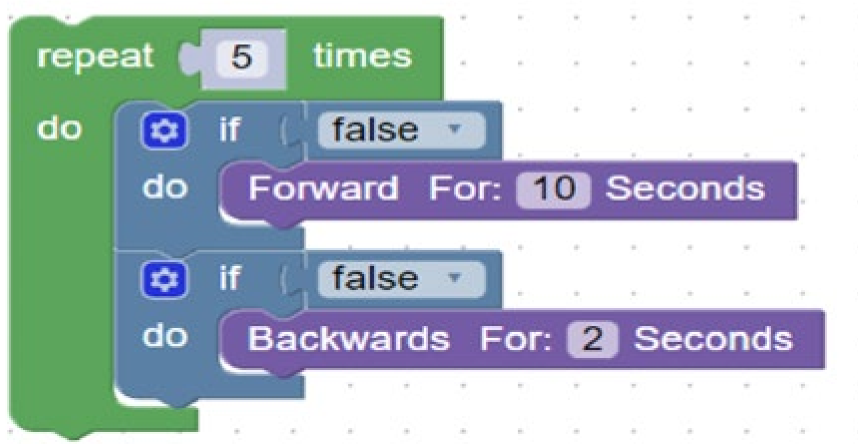
	1	2	3	4	5
1. I like to learn new technologies such as robotics.	○	○	○	○	○
2. It is important for me to understand basic principles of building and operating Robots	○	○	○	○	○
3. It is important for me to learn to program a robot.	○	○	○	○	○
4. I'm pretty sure I can program a robot to move forward and then stop.	○	○	○	○	○
5. I'm confident that I can fix a bot's code that isn't behaving as it should.	○	○	○	○	○
6. I think I can solve practical problems using robotics and programming.	○	○	○	○	○
7. I would like to find work related to robotics and programming.	○	○	○	○	○
9. I know the Blockly development environment.	○	○	○	○	○
10. I can program in the Blockly development environment.	○	○	○	○	○

If the Blockly code in the photo below is executed. What will EDUV do? 1: It will move forward. 2: It will perform a triangle move 3: It will perform a square move 4: it will make circles 5: I don't know *



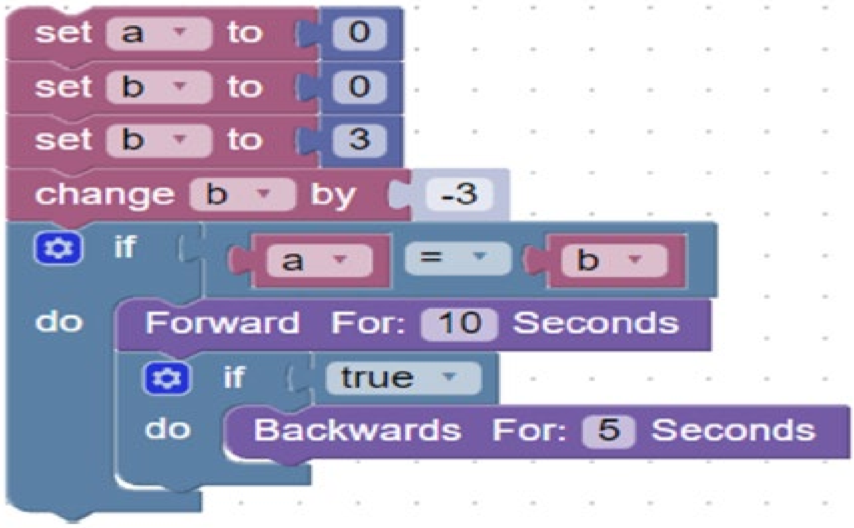
	1	2	3	4	5
ANSWER	○	○	○	○	○

If the Blockly code in the photo below is executed. What will EDUV do? 1: It will move forward for 10 seconds 2: It will move forward and then back 3: It won't do anything 4: I don't know *



	1	2	3	4
ANSWER	○	○	○	○

If the Blockly code in the photo below is executed. What will EDUV do? 1: It will move forward and then back 2: It will move forward 3: It won't do anything 4: I don't know *



	1	2	3	4
ANSWER	○	○	○	○

I If the Python code in the picture below is executed. What will EDUV do? 1: It will emerge and move forward 2: It will emerge 3: It will move forward 4: It won't do anything 5: I don't know *

```
from movement import*
x = 10
if x<=10:
    emerge(10)
if (x-10)>=0:
    forward(5)
```

	1	2	3	4	5
ANSWER	○	○	○	○	○

4. Questionnaire after using the EDUV Platform

Questionnaire after using the EDUV platform

Questionnaire of opinions on the EDUV programming learning platform and programming in robotics.
Filling in the questionnaire is done anonymously. The answers you will give are confidential and will not be used as a criterion for your evaluation.

1. Knowledge related to programming

For each of the adjacent statements circle the number that represents your answer according to the scale: 1 = "Strongly Disagree", 2 = "Disagree", 3 = "Neither Agree nor Disagree", 4 = "Agree", 5 = "Strongly Agree". *

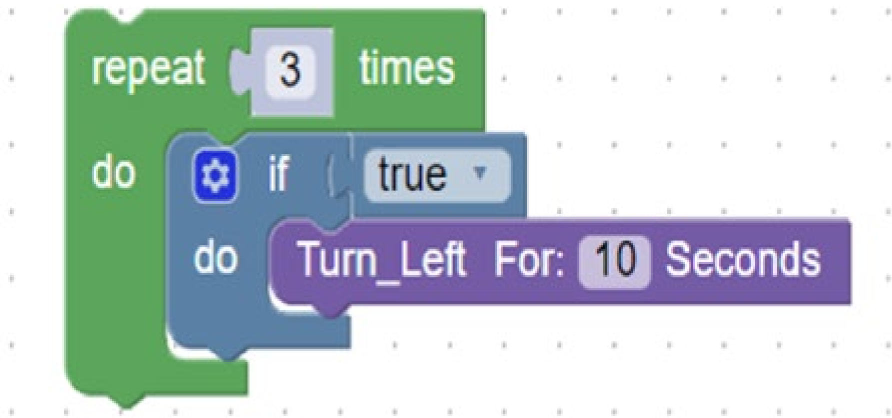
	1	2	3	4	5
1. I can explain what computer programming means.	○	○	○	○	○
2. I can program in some programming language	○	○	○	○	○
3. I like programming.	○	○	○	○	○
4. Programming lessons are enjoyable.	○	○	○	○	○
5. Computer programming will help me in my daily life.	○	○	○	○	○
6. Computer programming will help me find a job after I graduate from school/university.	○	○	○	○	○
7. To get good at programming I will need to do less of other activities.	○	○	○	○	○
8. I want to make an effort to learn programming.	○	○	○	○	○
9. When programming becomes difficult I give up.	○	○	○	○	○

2. Views on Robotics

For each of the adjacent statements circle the number that represents your answer according to the scale: 1 = "Strongly Disagree", 2 = "Disagree", 3 = "Neither Agree nor Disagree", 4 = "Agree", 5 = "Strongly Agree". *

	1	2	3	4	5
1. I like to learn new technologies such as robotics.	○	○	○	○	○
2. It is important for me to understand basic principles of building and operating Robots.	○	○	○	○	○
3. It is important for me to learn to program a robot.	○	○	○	○	○
4. I'm pretty sure I can program a robot to move forward and then stop.	○	○	○	○	○
5. I'm confident that I can fix a bot's code that isn't behaving as it should.	○	○	○	○	○
6. I think I can solve practical problems using robotics and programming.	○	○	○	○	○
7. I would like to find work related to robotics and programming.	○	○	○	○	○
8. I know how to program in Python.	○	○	○	○	○
9. I know the Blockly development environment.	○	○	○	○	○
10. I can program in the Blockly development environment.	○	○	○	○	○

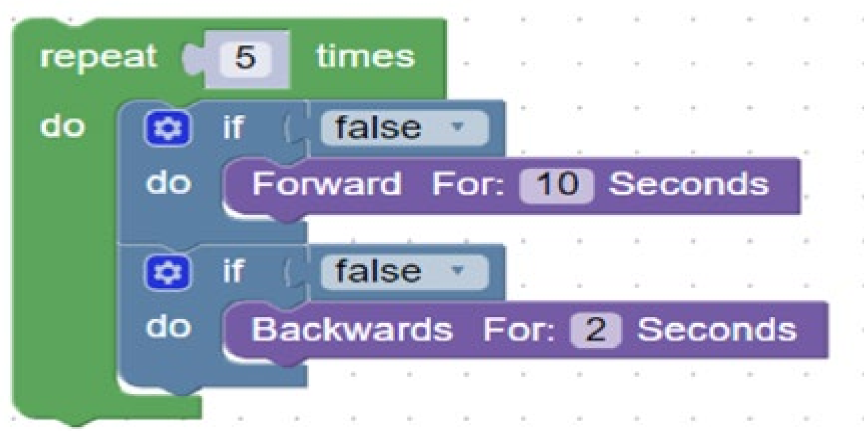
If the Blockly code in the photo below is executed. What will EDUV do? 1: It will move forward. 2: It will perform a triangle move 3: It will perform a square move 4: it will make circles 5: I don't know *



	1	2	3	4	5
ANSWER	○	○	○	○	○

*

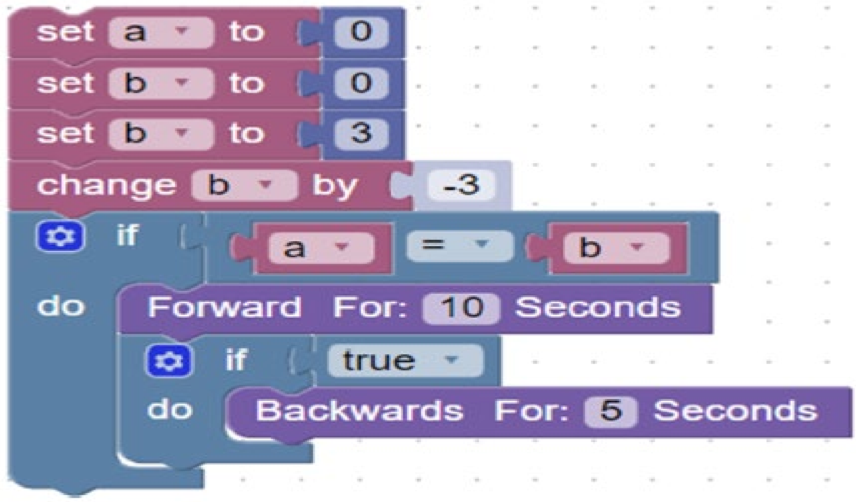
If the Blockly code in the photo below is executed. What will EDUV do? 1: It will move forward for 10 seconds 2: It will move forward and then back 3: It won't do anything 4: I don't know



	1	2	3	4
ANSWER	○	○	○	○

*

If the Blockly code in the photo below is executed. What will EDUV do? 1: It wiill move forward and then back 2: It will move forward 3: It won't do anything 4: I don't know



	1	2	3	4
ANSWER	○	○	○	○

*

If the Python code in the picture below is executed. What will EDUV do? 1: It will emerge and move forward 2: It will emerge 3: It will move forward 4: It won't do anything 5: I don't know

```
from movement import*
x = 10
if x<=10:
    emerge(10)
if (x-10)>=0:
    forward(5)
```

	1	2	3	4	5
ANSWER	○	○	○	○	○