computers

MDPI

Article

Measuring Undergraduates' Motivation Levels When Learning to Program in Virtual Worlds

Juan Gabriel López Solórzano [1], Christian Jonathan Ángel Rueda [2] and Osslan Osiris Vergara Villegas [3,*]

1 Facultad de Contaduría Pública y Administración, Universidad Autónoma de Nuevo León, San Nicolás de los Garza 66455, Mexico; juan.lopezsr@uanl.edu.mx
2 Research, Technological Development and Postgraduate, Universidad Politécnica de Santa Rosa Jáuregui (UPSRJ), Santiago de Querétaro 76220, Mexico; cangel@upsrj.edu.mx
3 Industrial and Manufacturing Engineering Department, Universidad Autónoma de Ciudad Juárez, Ciudad Juárez 32310, Mexico
* Correspondence: overgara@uacj.mx

Abstract: Teaching/learning programming is complex, and conventional classes often fail to arouse students' motivation in this discipline. Therefore, teachers should look for alternative methods for teaching programming. Information and communication technologies (ICTs) can be a valuable alternative, especially virtual worlds. This study measures the students' motivation level when using virtual worlds to learn introductory programming skills. Moreover, a comparison is conducted regarding their motivation levels when students learn in a traditional teaching setting. In this study, first-semester university students participated in a pedagogical experiment regarding the learning of the programming subject employing virtual worlds. A pre-test-post-test design was carried out. In the pre-test, 102 students participated, and the motivation level when a professor taught in a traditional modality was measured. Then, a post-test was applied to 60 students learning in virtual worlds. With this research, we have found that the activity conducted with virtual worlds presents higher motivation levels than traditional learning with the teacher. Moreover, regarding gender, women present higher confidence than men. We recommend that teachers try this innovation with their students based on our findings. However, teachers must design a didactic model to integrate virtual worlds into daily teaching activities.

Keywords: motivation;virtual worlds; programming; teaching–learning process; Minecraft education edition

Citation: López Solórzano, J.G.; Ángel Rueda, C.J.; Vergara Villegas, O.O. Measuring Undergraduates' Motivation Levels When Learning to Program in Virtual Worlds. *Computers* **2024**, *13*, 188. https://doi.org/10.3390/computers13080188

Academic Editor: Stelios Xinogalos

Received: 12 June 2024
Revised: 19 July 2024
Accepted: 29 July 2024
Published: 31 July 2024

1. Introduction

Students generally enroll onto the Introduction to Programming (IP) course in the first college semester because it is imperative in a curriculum of technology-related careers. For undergraduates, IP represents a fundamental pillar throughout their careers since what is learned will be applied in later subjects such as advanced programming. Programming is a complex subject for many students, with high failure and dropout rates. In Mexico, there are high failure rates of more than 70% in most universities. Moreover, the students who pass frequently do not obtain high grades, which coincides with what Groher et al. [1] explained.

IP students face several difficulties that cause their low performance. The work by Qian and Lehman [2] classifies the difficulties into three categories: (i) syntactic, (ii) conceptual, and (iii) strategic knowledge. Syntactic knowledge refers to the programming language rules, including poorly balanced parentheses, forgetting the semicolon at the end of an instruction, or using undeclared or uninitialized variables. Conceptual knowledge is associated with understanding the fundamental constructs of programming and how the computer works as a means to program. Examples of difficulties are not understanding variables, assignments, conditions, cycles, and sequential execution of code. Finally, strategic knowledge is related to applying syntactic and conceptual knowledge in problem solving. Difficulties in this category include the student not being clear about the appropriate cycle for a particular problem or not knowing how to read or debug the code. Another fundamental aspect that must be considered is motivation, because there is a close relationship between motivation and academic performance [3,4]. Specifically, demotivation is associated with the poor performance of students in educational settings [5].

Researchers and teachers continuously search for new methods to motivate and awaken interest in programming. For example, thanks to the advancement of Information and Communication Technologies (ICTs), 3D environments are frequently used in educational settings. Three-dimensional environments are interactive and immersive and can promote student motivation in teaching–learning processes because they use narrative ludic immersion [6]. Immersive digital environments include Augmented Reality (AR), Virtual Reality (VR), and Virtual Worlds (VWs). Due to the complexity of these three technologies, this research uses the Three-Dimensional Immersive Digital Environments (TDIDE) macro-concept proposed by [7] as a theoretical foundation. Particularly, the TDIDE employed in this research is the virtual world because it represents an educational innovation that, when properly implemented by the teacher, can significantly increase the motivation and engagement of students in the learning process [8–10]. Moreover, game elements can be used when designing VWs. The study of Pellas et al. used the "Learning Mechanics-Game Mechanics" model [11] to identify the most used game mechanics in the game prototypes designed in VWs. The game mechanics allow students to keep attention, participation, and engagement. The most used game mechanics were story, realism, and roleplay [12]. On the other hand, Pellas and Mistakidis identified many studies in which the game system, the narrative, and aesthetics were applied [13].

Currently, intending to make programming more accessible and help students overcome programming challenges, teachers and researchers have been working on two main aspects: (i) pedagogical approaches and (ii) tools. Pedagogical approaches include flipped classrooms, collaborative learning, feedback, game-based learning, metacognition, and problem-solving. The teachers can combine the approaches without losing their individual effectiveness [14]. On the other hand, tools include visual, automated assessment, programming environments, support tools, microworlds, and plagiarism detection [15,16]. Modern visual programming tools are widely used, and Scratch is especially effective [14,17]. It is important to note that developing tools requires effort and time [18], and many of them are not used outside the institution where they were developed [15]. This coincides with what was found in a study on three-dimensional immersive digital environments applied to teaching programming [19]. Therefore, to facilitate the incorporation of technological tools in the teaching–learning process, the proposal is not to develop tools but instead use the existing virtual worlds that are ready to be used.

In the literature, there are records since 24 years ago about the use of VWs for teaching programming [20,21]. The main reference in educational environments is "Second Life" [22], which is mainly used by people averaging 48 years of age [23]. Second life would be the benchmark for first-generation VWs that failed to maintain their popularity [24]. Consequently, an interesting research area is generated to learn more about the new generation of VWs, such as Minecraft and Roblox, that currently enjoy popularity and can also be used in learning environments [25]. Also, it is essential to understand that today's students grew up using technologies such as the Internet, video games, and computers [26].

As far as we know, there are no clues in the literature about works related to teaching/learning with virtual worlds of functions in programming courses. Therefore, the novelty of this work lies in the use of MEE and its corresponding elements to design a didactic activity for learning functions in an introductory programming course.

This study aims to measure the motivation level of students who used virtual worlds for learning functions in IP and compare it with the motivation level regarding traditional classroom learning. The main contributions of this paper are as follows:

- We explain the virtual world's design and the activities to help students learn functions in Minecraft Education Edition (MEE).
- We analyze and assess the motivation level of Mexican undergraduates using the virtual world MEE.
- We present evidence that a didactic model based on virtual worlds for learning programming is suitable for increasing students' motivation.

The rest of the paper is organized as follows. Section 2 discusses related studies on using virtual worlds, particularly MEE, for learning IP. Section 3 explains the research design, the phases that comprise the practice of MEE, and the instruments applied. The results are shown in Section 4. Finally, Section 5 presents the conclusions.

2. Literature Review

Many research studies have been published regarding teaching/learning programming difficulties in educational settings (on-site and online) [14,27,28]. Particularly, we analyzed the papers related to teaching/learning IP in virtual worlds and papers regarding the assessment of students' motivation.

Esteves et al. used Second Life for teaching–learning programming following an action research approach. The students were asked to develop a project within the VW using Linden Scripting Language (LSL). The main outcome of the research is a practical framework consisting of the following elements: (i) communication, (ii) project, (iii) methodology, (iv) classroom, and (v) lessons. The author's recommendations when using VWs include using private and public communication channels for explanations, visual projects, and project-based learning. The results showed that students were enthusiastic and motivated when using LSL [20].

Rico et al. developed a platform called V-LeaF to teach the IP course. The proposed approach used OpenSim as a virtual world engine. Secondary-level students and teachers used LSL as the programming language. As general findings, the study reports that teachers and students indicate that using VWs is a satisfactory experience, as it allows collaboration and cooperation [29].

Hulsey et al. presented Curiosity Grid, a virtual world supported by OpenSim. The VW was used to introduce programming concepts to high school girls in a summer course. The students solved ten design, construction, and scripting challenges using LSL. Pre-camp and post-camp surveys were administered to evaluate the effectiveness of the proposal. The results showed a positive increase in the students' attitudes towards programming [30].

Kao presented JavaStrike, a Java language runtime environment developed in Unity. JavaStrike supports the visualization, execution, and debugging of Java language code. The authors also presented CodeBreakers, a game incorporating JavaStrike as a code execution engine. CodeBreakers can be used to teach programming at basic, intermediate, and advanced levels. Twenty-seven participants evaluated the platform positively in terms of competence, autonomy, immersion, and controls. The participants expressed a need to incorporate more instructions or support for beginner Java programmers [31].

Kutay and Oner used Minecraft to teach programming to 20 high school students. The design corresponds to a pre-test and post-test to investigate the changes in the knowledge of computational concepts. For the pre-test, the knowledge was assessed using Scratch, and the post-test using Minecraft. The results indicate that students performed better when tested on computational concepts after using Minecraft [32].

Debugging and testing were the practices that students employed the most in Minecraft activities because they are excellent visual feedback mechanisms. The student can see the results of their code in real-time within the virtual world. However, in the study conducted with high school students, no significant differences regarding motivation were obtained when using Minecraft [33].

A teaching method to facilitate and motivate learning introductory programming using Minecraft was proposed in [34]. The proposal comprises three abstraction levels: (i) designing with building blocks, (ii) building with code blocks, and (iii) general programming and computational thinking. The study involved 16 university students with little or no programming knowledge. The results indicate that the students consider the proposed abstraction levels adequate and facilitate our understanding of programming concepts. Unfortunately, students mentioned that the Minecraft coordinate system is complicated.

In Bile's study, MEE was used for a programming course involving 189 students aged 8–10. The didactic method allowed the students to concretize their knowledge through practical experience. The positive results show that learning through MEE promotes the fast and solid learning of complex concepts [35].

Sajjanhar and Faulkner used the "Second Life" virtual world to teach programming using a constructivist approach. The study involved 12 undergraduate and master's students. Students used Scratch for Second Life to perform activities on variables, cycles, and conditional structures. The findings indicated that the proposed approach effectively teaches basic programming concepts [36].

The research by [37] conducted a Minecraft Hour of Code study involving 63 students aged 10–12. Hour of Code consists of tutorials designed to introduce basic programming skills. The results indicate that Hour of Code helped students understand the concept of programming and fostered a positive disposition towards programming.

Alsaadi et al. used an educational tool called "Creative Programmer", a Minecraft "mod" for teaching basic programming concepts. The tool was used by seven children aged 7–15 years. Three activities were performed: defining variable data types, writing commands to create an algorithm, and understanding the concept of repetition (cycles). The results revealed that the children improved their performance regardless of whether they had previous programming knowledge [38].

Teaching programming with VW encompasses two paradigms: (i) the declarative and (ii) the imperative. The declarative includes logic programming, and the imperative includes procedural programming. Vosinakis et al. presented MeLoISE for teaching–learning programming in Prolog. The authors conducted two studies to evaluate performance, collaboration, and user experience. The results revealed satisfactory student performance, positive collaboration, and good user experience [39].

2.1. Assessing Motivation in Introduction to Programming

Motivation is a complex process and represents a state of energy that manifests itself in goal-directed behaviors [40]. The attention, relevance, confidence, and satisfaction (ARCS) model was chosen because it has been used worldwide and at different educational levels [4], and also because it is based on the expectancy–value theory, which is one of the contemporary motivation theories to learn [41]. The expectancy–value theory explains that people are motivated to perform an activity if two conditions are met: (i) If they perceive that the activity has utility or personal value and (ii) If they perceive themselves capable of performing the activity successfully.

According to Keller, for a person to be motivated and stay motivated, four conditions must be met: (i) attention, (ii) relevance, (iii) confidence, and (iv) satisfaction [42].

- Attention. It is the first condition for motivation and also for learning. The main challenge is to catch the student's attention and keep it.
- Relevance. It is related to whether the course is relevant to the students and meets their needs.
- Confidence. The expectation of success in a particular task is important, as it influences the student's persistence and achievement.
- Satisfaction. When any activity is performed successfully, it is necessary for the person to feel good about it.

The study by Fang et al. analyzed the applications of the ARCS model. A total of 55 studies were part of their analysis. The most important findings indicate that ARCS is most used at the higher level. Most studies (n = 47) applied learner-centered environments, including AR, VR, and VWs. The most commonly used pedagogical approach was multimedia learning [3]. According to Garzón et al., in their classification of pedagogical approaches, multimedia learning is defined as learning that allows students to construct their knowledge when they have access to other media besides the text. Multimedia learning was also used for the present study since MEE is a constructivist educational tool that provides multiple media such as text, image, sound, 3D models, chat, and NPCs. In addition, multimedia learning includes game-based learning [43].

Some examples of using the ARCS model in teaching programming are the following.

In the study by Bakar et al., Turtle Graphics was employed to maintain high motivation levels when teaching the basics of Java programming. A total of 167 higher-level students participated in the study. The ARCS model was used to assess the students' motivation. The results revealed that the students demonstrated high motivation in all four motivational aspects.

The ARCS model was also employed to assess motivation when using teaching methodologies. Choi et al. present the effects of using the flipped classroom in a block-based programming course. The results revealed that the experimental group that participated in the flipped classroom obtained higher motivation levels than students in the control group that employed traditional teaching techniques [44].

2.2. Pedagogical Foundations of Virtual Worlds

According to Girvan and Savage, VWs have several features such as (i) client–server environment; (ii) construction tools; (iii) programming tools; (iv) communication tools; (v) 3D environment; (vi) avatars, and (vii) multiple users. Teachers and researchers can combine the features in various ways and apply them to the educational environment. However, it is important to ground learning experiences developed in VWs pedagogically. Currently, immersive experiences using VWs are usually based on constructivism (or some of its variants, e.g., social constructivism [36], communal constructivism [45], or constructionism [46,47]). The present research is grounded in constructionism, whose fundamental principle is the creation of meaningful and shareable objects that, during the process, students actively explore, test, and improve their understanding [48,49].

Students learn through active participation, collaboration, problem-solving, and exploration, i.e., "learning by doing". Due to the generally constructionist approach of VWs, the common practices are (i) problem-based learning, (ii) collaborative learning, and (iii) game-based learning [50,51]. VWs, compared to traditional 2D platforms, include technological capabilities such as a sense of presence, different types of communication, avatars, expressiveness, and real-time simulation [52]. The capabilities allow students to experience-based learning [50].

2.3. Gender in E-Learning and VW

Studies regarding the gender variable indicate no significant differences in learning outcomes concerning e-learning [53–55]. However, in the VWs, the opposite happens. Women have more social activity, use learning opportunities and resources, participate in construction activities, and appreciate the entertainment value of VWs [56]. Therefore, it is advisable to study gender. According to Lin et al., learners' gender should be considered in the design of VWs since women use and perceive the systems differently than men [57]. For example, in the work of Korlat et al., women value e-learning, the support of their teachers, and the intrinsic value and engagement in their learning more than men [58].

3. Materials and Methods

After analyzing the research problem to be solved, the following two research questions were posed.

- RQ1. Could IP learners using MEE achieve higher motivation levels than traditional learning?
- RQ2. Could men obtain higher motivation levels than women when learning IP using MEE?

The approach of this study is quantitative with an exploratory and descriptive scope. The purpose is to determine whether there is a significant difference in the motivation levels of students who use virtual worlds compared to traditional learning.

ARCS model was chosen because it has been used worldwide and at different educational levels [4], and also because it is based on the expectancy–value theory, which is one of the contemporary motivation theories to learn [41].

3.1. Instruments

Keller developed the Instructional Materials Motivation Survey (IMMS) based on the ARCS model to measure motivation [59]. IMMS is composed of 36 items on a 5-point Likert scale. IMMS can be used to calculate the overall motivation level or for each ARCS model subscale. There is evidence in the recent literature that different adaptations of IMMS have been used [3]. For example, Loorbach et al. evaluated the instrument's theoretical and statistical aspects. Using structural equation modeling, they showed that the 36 items of the original IMMS could be reduced to 12. Loorbach et al. called it the Reduced Instructional Materials Motivation Survey (RIMMS) [60]. RIMMS is based on the ARCS model, which includes attention, relevance, confidence, and satisfaction. The instrument contains 12 items and uses a Likert-type scale that ranges from 1 (not true) to 5 (very true). RIMMS is valid and reliable since it reflects the conditions and principles of the ARCS model [60] and has been applied in teaching using augmented reality [61]. Therefore, the present study used RIMMS to measure student motivation.

RIMMS was used in the pre-test to measure motivation in the traditional modality that corresponds to face-to-face classes taught by the teacher. Additionally, RIMMS was employed in the post-test when students experimented with virtual worlds.

The proposed method comprises five stages: (i) analysis of virtual worlds; (ii) thematic content; (iii) virtual world design; (iv) materials design; and (v) activity implementation.

3.2. Analysis of Virtual Worlds

In the first stage, we analyzed the technological tools for teaching programming. The literature has many virtual worlds with different characteristics and computational requirements. However, due to the costs, we decided to analyze those that do not require a gamer-type computer, that is, those that do not need a dedicated Graphics Processing Unit (GPU) to conduct fluid execution.

Therefore, the virtual worlds considered were Roblox, Minecraft Education Edition (MEE), Mozilla Hubs, OpenSim, and Decentraland. After the analysis, MEE was selected because it is a VW designed for educational purposes and based on constructionist theory [62]. MEE integrates Scratch and supports programming languages, such as Javascript and Python, which are widely used in software development. Minecraft is similar to microworld because it is an incubator of knowledge [48]. Novel programmers can use Minecraft to generate, experiment, debug, collaborate, and share the products created through code. MEE offers the possibility of concretizing abstract concepts and facilitating understanding by incorporating visual elements [63]. Minecraft is special because it is part of the most recent 3D interactive platforms, such as Roblox and Fortnite, and has become the model for the next development of the Metaverse [64]. Minecraft, unlike Roblox and Fortnite, has a version for educational environments. In addition, MEE is suitable due to its accessibility and availability compared to VR, which needs specialized equipment that cannot be easily acquired [65].

3.3. Thematic Content

When students learn to program, they face several challenges, including creating conditions, loops, and functions. However, when the conditions and cycles are mastered, learning to break a program into more manageable pieces called functions is essential. Therefore, the thematic content chosen for the activity of this paper corresponds to the topic of making "functions" in Python. In introductory programming, students must understand functions and know how to use or create them. Functions are self-contained modules of code that accomplish a specific task and can be reused in any part of a program.

In MEE, the students can use predefined functions to manipulate the virtual world, and more importantly, new functions can be created. The activity requested from students is similar to the study conducted by [66], where the Logo language was used to control a turtle and draw different figures. In MEE, an "agent" helps students perform construction tasks within the world using programming instructions. An example of a MEE agent (robot) is shown in Figure 1.

Figure 1. Example of an agent in MEE.

Some of the functions that can be performed by the agent and used in practice with MEE are summarized in Table 1.

Table 1. Functions to control the agent.

Function	Description
agent.move(FORWARD,1)	Moves the agent one block forward
agent.move(BACK,1)	Moves the agent one block backward
agent.move(LEFT,1)	Moves the agent one block to the left
agent.move(RIGHT,1)	Moves the agent one block to the right
agent.move(UP,1)	Moves the agent one block up
agent.move(DOWN,1)	Moves the agent one block down
agent.turn_left()	Turn the agent to the left
agent.turn_right()	Turn the agent to the right

3.4. Virtual World Design

Virtual worlds in higher education have mainly been applied for virtual classrooms, discussions, simulations, games, and field trips [6,67]. We selected the virtual classroom's didactic strategy for the virtual world's design. The strategy consists of replicating or simulating a classroom or space where the teaching–learning occurs. Elements incorporated in MEE, such as blackboards, boards, and agents, were used to present information to the students, as shown in Figure 2. Although MEE provides the teacher with many pre-designed lessons created by other teachers or Minecraft administrators, many of these activities target students aged 8–10 and 11–13 [68]. Therefore, the virtual world was designed with university-level students in mind and uses textual programming, commonly employed in programming courses.

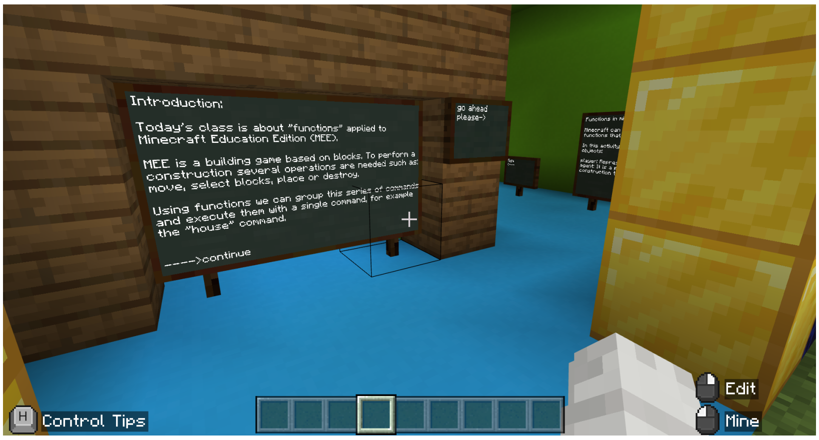

Figure 2. Elements used in MEE for learning functions.

The overall design of the virtual world can be seen in Figure 3. The design consists of a main structure with an entrance, non-player characters (NPCs), and whiteboards with directions that students must follow during the activity.

Figure 3. View of the space built in MEE.

In the virtual world, the student can make changes using commands in Python language. Moreover, the student can visualize the command execution or programming activities in real-time. Figure 4 shows the MEE command editor with an example of Python code.

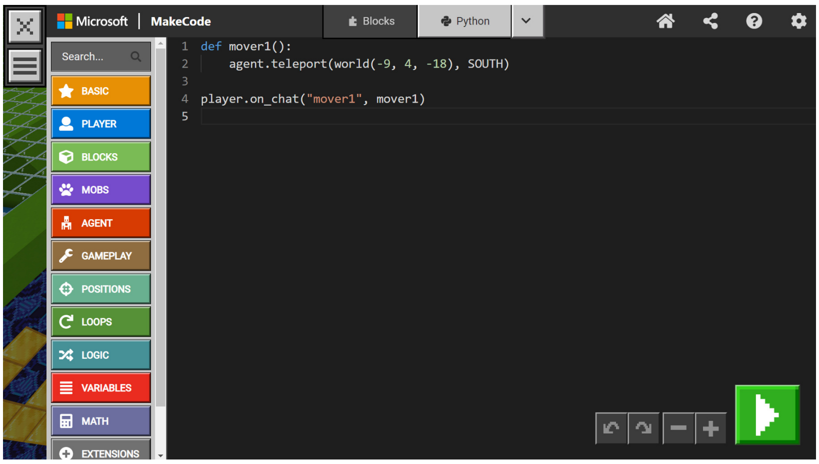

Figure 4. The code editor in MEE.

Generally, the experience consisted of presenting the student with information regarding the use of functions within MEE. Students must understand that functions are used to allow the execution of tasks within the virtual world. During the learning process, the student was asked to perform a series of programming activities described in Table 2.

Table 2. Activities requested from students in MEE.

Activity	Description
A1. Use Microsoft Make Code	Write code that will be executed within the virtual world to move the "agent" to a specific location.
A2. Create a function to be executed in the chat	Use the commands in Table 1 to create a function to move the agent in the virtual world. The function's execution should be done on demand when the function name is typed in the chat.
A3. Final activity	Define a function for the agent to follow a route marked in yellow. The start and end of the route are marked in red and green, respectively, as observed in Figure 5.
A4. Bonus activity	Place blocks on the marked area (see Figure 6), with a minimum height of one block. The student was asked to use conditional structures and cycles. Additionally, the design of a function that allows the agent to place a block as it moves through the virtual world was requested.

Figure 5. The final activity in MEE.

Figure 6. The bonus activity in MEE.

3.5. Material Design

In this stage, PowerPoint presentations and PDF files were generated with information on how the student could move around in MEE using a desktop computer or mobile device. The documents also provided information on MEE concepts and some commands used within the virtual world. An example of this information is shown in Figure 7.

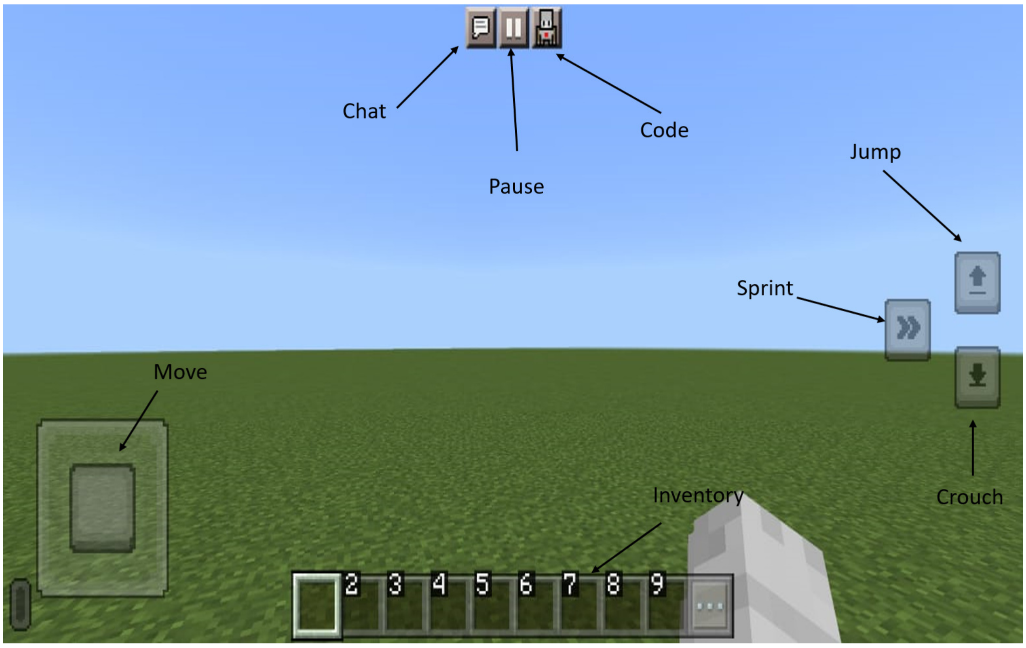

Figure 7. MEE controls for mobile devices. The inventory uses a number for each slot.

Design of Didactic Materials in MEE

Figure 8 shows the phases for designing the didactic material used in the VW. Each phase is explained below.

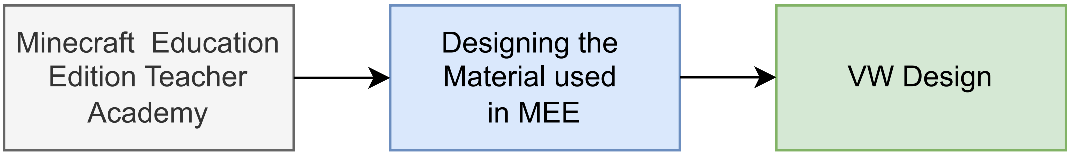

Figure 8. Phases for designing the didactic material used in MEE.

- Phase 1—Minecraft Education Edition Teacher Academy: A face-to-face and online Minecraft Education Edition Teacher Academy certification was conducted. The certification comprises three sections: (i) beginner, (ii) intermediate, and (iii) advanced. In the beginner section, the designer starts playing in MEE, placing blocks, and managing the materials inventory. In the intermediate, the classroom mode is used. Topics include setting up the classroom and managing NPCs, whiteboards, and posters. Finally, the advanced section focuses on programming with Microsoft Make Code.
- Phase 2—Designing the Material used in MEE: All the information concerning functions and the description of each activity to be performed were stored in a plain text file for later use in the MEE. The information was copied and pasted into the VW in the design phase.
- Phase 3—VW Design: The phase started with a paper design of the general structure of the VW. Subsequently, construction proceeded using the following resources provided by MEE:
    - Board (3×3 blocks). Used to present information on functions and instructions.
    - Poster (2×1 blocks). Used to provide information on activities.
    - Slate (1×1 block). Used for signaling within the VW.
    - NPC. Used for presenting information interactively.
    - Agent. A programmable element that students use to execute commands within the VW.

- Miscellaneous blocks. Blocks such as concrete, glass, and lights were used to construct the building.

3.6. Activity Implementation

Figure 9 shows the activities conducted during the study. The workflow is explained in the following subsubsections.

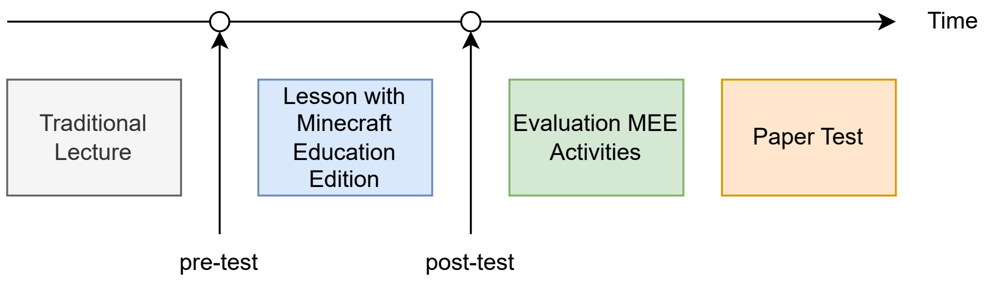

Figure 9. Workflow of the study.

3.6.1. Traditional Lecture

Before using the virtual world in MEE, the teacher taught students the topic of programming functions in the traditional modality.

The traditional approach is teacher-centered, dominated by lectures, in-class exercise resolution, and homework [69,70]. Topics covered included variables, operators, control structures, and functions. After the face-to-face classes, students were asked to answer the instrument to evaluate their motivation level (pre-test). It is essential to mention that, when students answered the survey, they did not know that the activity would be performed by employing MEE in the future. This was decided to avoid bias.

3.6.2. Lesson with Minecraft Education Edition

Before starting the practice, the students were asked to install MEE on their computer equipment or mobile devices because the institution does not have the infrastructure to carry out the activity. Some students could not conduct the activity due to several inconveniences, such as (i) their equipment did not meet the minimum requirements for installing the tool, and (ii) in some devices, the code execution (essential for the activity with MEE) did not work. After the installation, to familiarize the students with the game, they were instructed to log in to MEE with their institutional account and play in the virtual world called "Movement". The movement virtual world is located in the Templates section of MEE and was designed to be experienced using a keyboard or touch device.

The immersive activity was conducted online. Therefore, a session in Microsoft Teams was scheduled to conduct the study. During the study, one professor was connected to respond to the students' queries. Microsoft Teams Chat was used to send evidence of the activities performed.

During the session, the teacher shared the file Activity_MEE_Intro_Prog.mcworld with the group and provided the necessary indications to import the virtual world. Once the world was correctly imported, the students could perform activities related to the "functions" topic.

At the end of the MEE activity, the students sent evidence of the activities through a private Microsoft Teams chat. In addition, students were asked to fill out the survey to measure their motivation level using MEE (post-test).

3.6.3. MEE Activities Evaluation and Paper Test

After the activity with MEE, the teacher graded the activities performed by the students. Finally, a paper test was administered.

4. Results

Two groups of the Introduction to Programming course of the August–December 2022 semester participated in the study. One hundred and two students participated in the traditional class, and only sixty students participated in the activity applying the virtual world. Forty-two students did not carry out the activity with MEE because they did not have the necessary computer equipment to install the tool or decided not to participate. The sample is non-probabilistic because it is convenient since the participants belong to assigned class groups. The paired sample of the experiment is composed of 60 students.

4.1. Results with RIMMS

The results of the RIMMS instrument were analyzed for the materials offered by the teacher and with MEE. The following subsections discuss the results for each dimension of the ARCS model.

4.1.1. Results for Attention

Figure 10 shows that many students agree that the content quality, the information organization, and the interaction used in MEE helped them maintain their attention. It was observed that students were more participative during the activity with MEE. Moreover, some students share evidence screenshots of the exercises they performed in the virtual world.

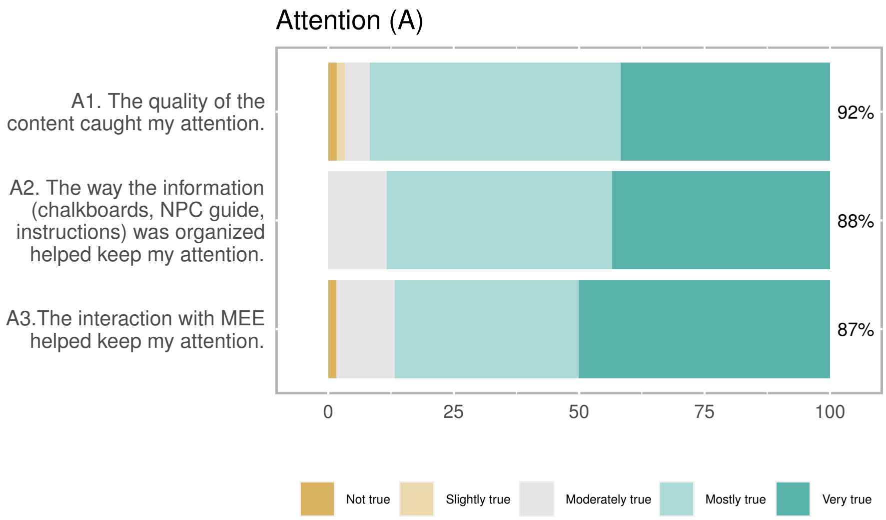

Figure 10. Attention in MEE.

The RIMMS results for the teacher's material are shown in Figure 11. It is essential to highlight that many students agree that the variety of readings, exercises, and illustrations employed by the teacher helped maintain their attention in class. During the lectures, the teacher indicates that the students paid attention to what was explained, but when asked to solve an exercise, it becomes evident that attention is not synonymous with learning or understanding the topic.

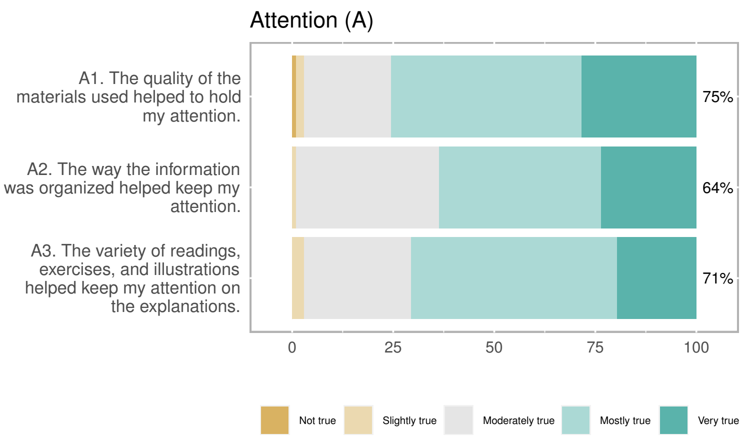

Figure 11. Attention teacher's material.

4.1.2. Results for Relevance

Using MEE in teaching programming promotes students' attention, indicating that the content shown in MEE is relevant (see Figure 12).

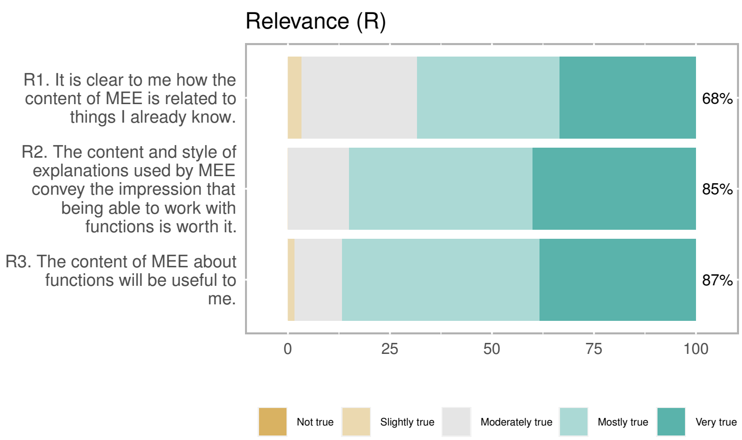

Figure 12. Relevance in MEE.

The results on the relevance of the lectures given by the professor are shown in Figure 13. Students agree that it would be relevant to go deeper into the topic and that employing the contents provided by the professor would be useful. In contrast, students disagree or are undecided that the teacher's material is related to topics they already knew.

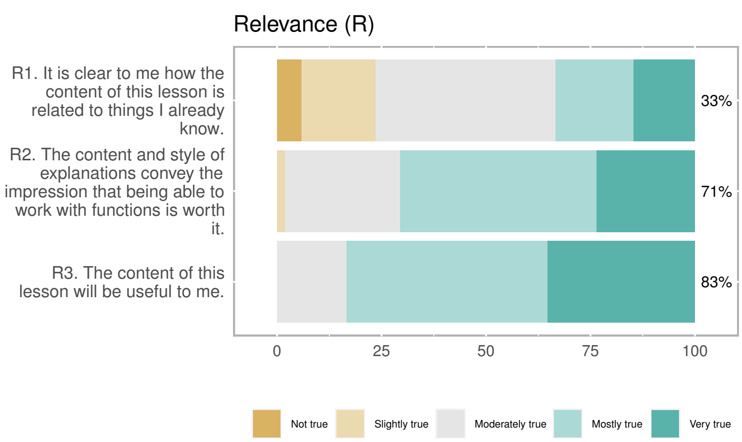

Figure 13. Relevance of teacher's material.

4.1.3. Results for Confidence

Figure 14 shows that students agree that they could learn the content presented in MEE and that this is due to the clarity of the concept explanations. On the other hand, students do not agree that they could pass a test on the topic.

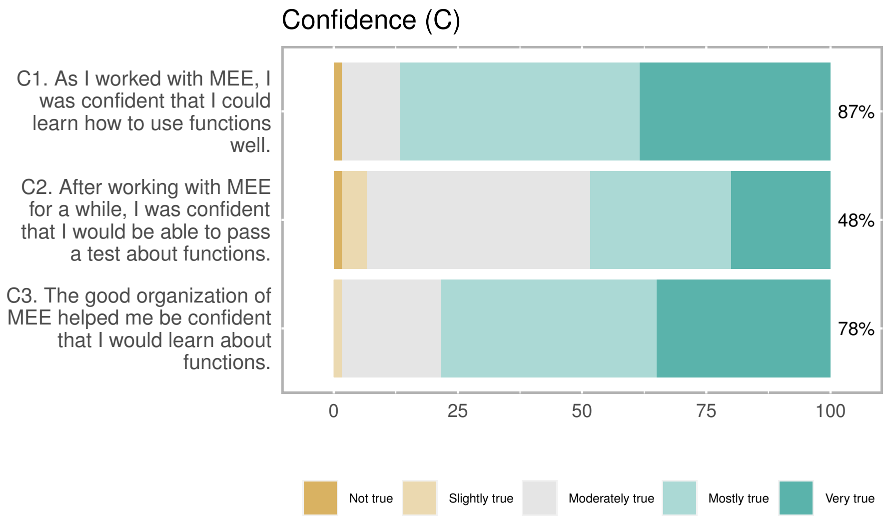

Figure 14. Confidence in MEE.

The results corresponding to the teacher's material can be seen in Figure 15. It is important to highlight that it does not matter whether the teacher's material is used; most students do not agree or are undecided about whether they will pass an exam on the topic. However, students are also confident that the clarity of the explanations and work in class would be conducive to learning and the correct application of the topics raised by the teacher.

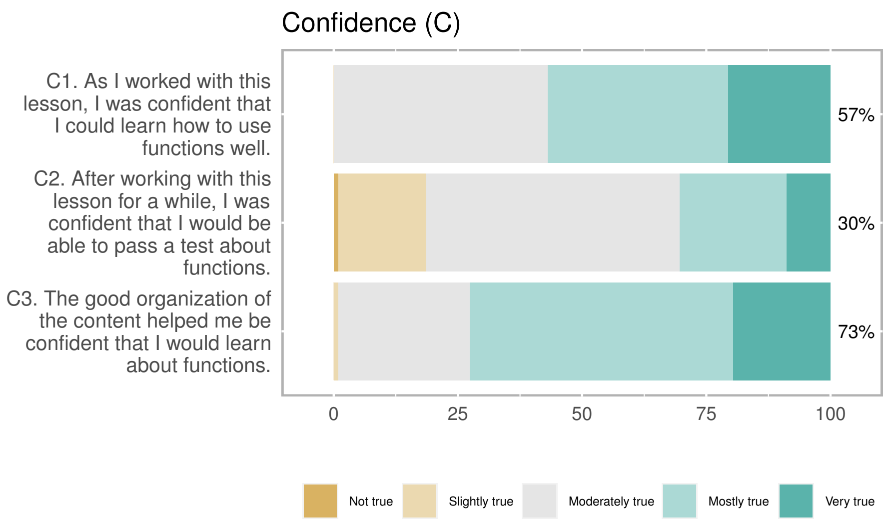

Figure 15. Confidence in the teacher's material.

4.1.4. Results for Satisfaction

Finally, the Satisfaction dimension is presented. Most students experimented with high satisfaction rates using MEE (Figure 16) and the material offered by the teacher (Figure 17).

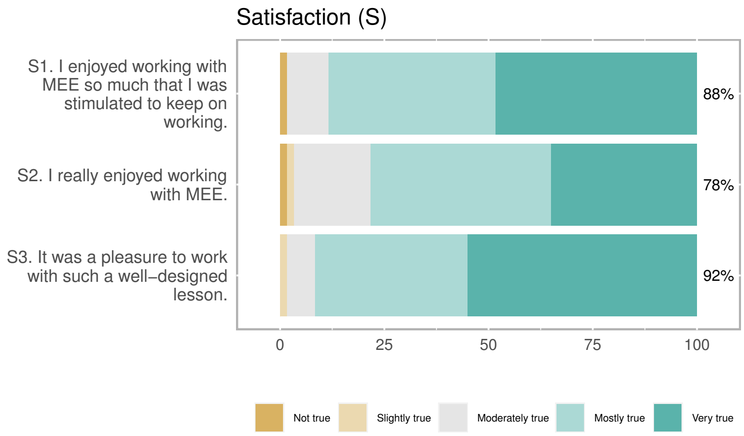

Figure 16. Satisfaction in MEE.

Figure 17. Satisfaction with the teacher's material.

The results were obtained by considering only a paired sample of 60 students, corresponding to the students participating in the MEE activity and the teacher's classes. From the total sample, 18 are women, and 42 are men. Women were observed to have higher motivation levels than men when using MEE. Figure 18 shows that all the factors (attention, relevance, confidence, and satisfaction) are higher for women.

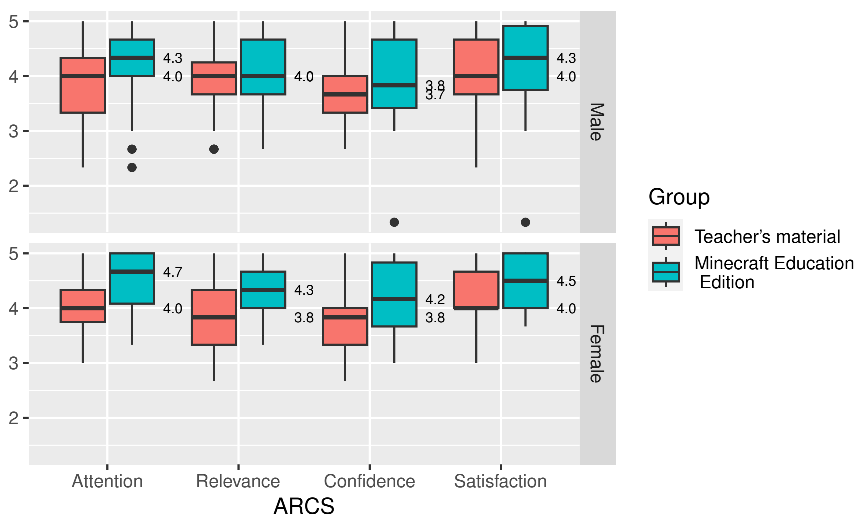

Figure 18. ARCS comparison by gender. The circles correspond to outliers.

Finally, considering the complete sample (n = 60), the results obtained in each factor show an increase in student motivation when using MEE, as seen in Figure 19. The figure shows the pre-test corresponding to the material or class taught by the teacher in their traditional lecture and the post-test in which the virtual world MEE was used.

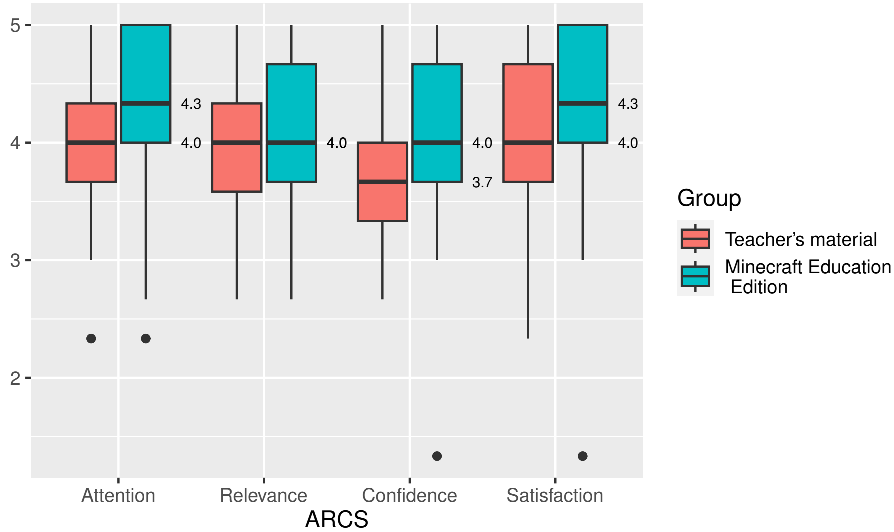

Figure 19. ARCS pre-test and post-test results. The circles correspond to outliers.

In addition to the results obtained with the RIMMS survey, it was possible to verify that the students were very committed and understood the material since, among the 60 students, 73% (n = 44) sent evidence of the exercises carried out within MEE. In contrast, in the traditional modality, when they were asked through an exam to solve a problem using functions with the Python language, only 22% of the students solved it correctly, and 78% (n = 47) did not do it correctly or did not even try to solve it, leaving the answer blank. On the other hand, some students commented that the activity was very fun. Some of the messages sent by the students were the following:

- "The activity was fun. I easily realized the last activity, called bonus. I will send the results as soon as possible."
- "Professor, I was able to do the activity. The instructions say to use an if, for, or while. I only used an if and a while. I could not modify more things in the game because I can not use other blocks, but it works."

4.2. Evaluation in MEE and Test

The students' activities performed in MEE were sent to the teacher for evaluation. In addition, the paper-based test was reviewed. The test contained an exercise about functions. The results are shown in Table 3.

Table 3. Results of MEE and test.

Evaluation	M	SD	Pass	Fail
MEE evaluation	59.33	36.81	43	17
Paper test	31.08	36.40	13	47

As can be observed in Table 3, the mean for MEE was 59.33. The low mean is because the 17 failed students were graded with zero since they did not send their evidence to the professor for evaluation.

4.3. T-Student Test Results

The normality test indicated that the data were normally distributed. Therefore, the paired t-test with a 5% level of significance was conducted. The aim was to determine whether a significant difference exists between the motivation levels in traditional modality and employing MEE. On average, students who participated in MEE had a higher motivation level (M = 4.18, SE = 0.080) than those who participated in the traditional modality (M = 3.91, SE = 0.071). Therefore, the difference −0.27, 95% CI [−0.44,−0.09] is significant, t(59) = −3.09, *p*-value = 0.003.

Table 4 summarizes the results for each ARCS dimension. As can be observed, all show significant differences except satisfaction, which shows a *p*-value greater than 0.05.

Table 4. T-student results for the ARCS four dimensions.

Variable	t	*p*-Value	CI	Mean Difference
Attention	−3.39	0.001	[−0.53, −0.14]	−0.33
Relevance	−3.11	0.003	[−0.44, −0.10]	−0.27
Confidence	−2.56	0.013	[−0.48, −0.06]	−0.27
Satisfaction	−1.96	0.054	[−0.43, 0.004]	−0.21

4.4. Discussion

Our research shows that, using VW, such as MEE, increases students' motivation compared to the teacher's material in obtaining the answer to RQ1. However, because the proposed research design is not longitudinal, it does not allow us to rule out the novelty effect of using MEE. However, it is important to mention that 78% of the students had played video games in the paired sample. This could counteract the novelty effect caused by immersive technologies such as MEE. The novelty effect may appear when students are introduced to the new technology and even more so when 60% of the sample has not used video games in educational settings. Another factor that should be considered is that, in general, students respond positively to using ICTs in the learning process [71]. In addition, it should be considered that the age of the students in the present study affects their sense of presence, interaction, and satisfaction in VWs [72].

The results presented in this paper agree with the findings of Wang et al. since their serious VR game satisfies the elements of the ARCS model, as all items had means higher than four [73].

In the traditional modality, it is alarming that 78% of students did not even try to solve a written programming problem. However, with MEE, 73% were interested in programming activities; this may be due to how fun these types of environments are for students and that they allow them to understand the IP material better. Therefore, teaching IP with virtual worlds is important because of the tool and the type of problems that students can perform. In addition, it is advantageous to change classic problems in programming teaching, such as "calculating the average of n grades", to more visual and interesting problems for students, such as building objects or moving their "agent" in the virtual world. Moreover, when using MEE in IP, teachers have a ready environment where students can create artifacts. MEE provides immediate visual feedback, and the student can modify the code to obtain the desired results. Through trial and error, the student can learn to debug the code [70]. Solomon and Papert used a similar approach with their Turtle Graphics using Logo [66].

MEE is a tool that captures students' attention, and the content shown or the things that can be created make it a relevant virtual world to deepen and increase interest in programming [74]. Regarding satisfaction, students indicate that working with MEE is very satisfactory. When it was proposed to use MEE, many students were surprised and took the proposal with great pleasure, which aligns with the comments mentioned in [75].

It is important to highlight that confidence about the test presented the lowest levels in using MEE and the professor's material. This agrees with [76], as exams generally produce moderate anxiety levels in students.

Regarding the results obtained concerning gender, high levels of motivation were observed in women, which coincides with what is indicated by [77,78] and answers RQ2. Women like games more when they are used for educational purposes. It is important to highlight that women express having higher levels of confidence than men, contrary to what was found in [79], where the authors indicate that men have greater self-efficiency (confidence) than women. Therefore, virtual worlds could be an ideal alternative for women to improve their IP performance.

In summary, based on the motivation results and students' performance, we demonstrated that teaching functions in virtual worlds could mitigate the high failure rates in programming in the first semesters of higher education. Therefore, 72% (n = 43) of the students had satisfactory results in MEE versus only 22% (n = 13) who passed the paper-based exam. The virtual worlds offer professors an innovative way to motivate and involve the students in a learning environment [37]. Furthermore, although there may be different student-centered environments that foster motivation, such as Scratch [80] and App Inventor [81], an evolution to immersive technologies is needed. MEE as an educational game and new metaverse tool [64] will undoubtedly be part of the future trends in computer programming education.

5. Conclusions

It would be important to highlight the need for a didactic model for using virtual worlds in a teaching programming, allowing teachers to incorporate this innovation in classrooms. The proposed teaching model could help teachers face possible problems or difficulties when using virtual worlds in the PI teaching–learning process. According to [19], few studies indicate virtual model implementation problems. However, issues are always present. For example, in this study, a technical problem arose that prevented the execution of code within the MEE on some of the students' computers. The issue was resolved with an MEE update. In addition, infrastructure problems related to the computing equipment necessary for using the tool could be addressed with a multiplatform virtual world. The tool can be used on a tablet, desktop, mobile phone, Chromebook, browser, or video game console. In the case of the present study, many students did not participate in the activity because their equipment did not meet the minimum MEE requirements.

The present study deals with the following limitations:

- The sample was not random, as class groups assigned to the teacher were taken.
- The study was conducted online, not in the institution's computer lab.
- The activity was not performed with MEE in collaborative mode. Each student individually worked on their activities.
- Students used personal computers, and participation was affected as many did not have computers that could properly run MEE.
- From the teacher's point of view, applying the VMs requires additional time and effort compared to traditional teaching.

In future research, it would be important to determine the effects of virtual worlds in other more advanced programming courses, higher-level subjects, and long-term studies.

Author Contributions: Conceptualization, J.G.L.S., C.J.A.R. and O.O.V.V.; methodology, J.G.L.S., C.J.A.R. and O.O.V.V.; software, J.G.L.S., C.J.A.R. and O.O.V.V.; validation, J.G.L.S., C.J.A.R. and O.O.V.V.; formal analysis, J.G.L.S., C.J.A.R. and O.O.V.V.; investigation, J.G.L.S., C.J.A.R. and O.O.V.V.; resources, J.G.L.S., C.J.A.R. and O.O.V.V.; data curation, J.G.L.S., C.J.A.R. and O.O.V.V.; writing—original draft preparation, J.G.L.S., C.J.A.R. and O.O.V.V.; writing—review and editing, J.G.L.S., C.J.A.R. and O.O.V.V.; visualization, J.G.L.S., C.J.A.R. and O.O.V.V.; supervision, J.G.L.S., C.J.A.R. and O.O.V.V.; project administration, J.G.L.S., C.J.A.R. and O.O.V.V. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank Universidad Autónoma de Nuevo León.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Acknowledgments: Juan Gabriel López Solórzano would like to thank Consejo Nacional de Humanidades Ciencias y Tecnologías (CONAHCYT) for the doctoral scholarship.

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

Abbreviations

The following abbreviations are used in this manuscript:

AR	Augmented reality
ARCS	The attention, relevance, confidence, and satisfaction model
GPU	Graphics processing unit
ICTs	Information and communication technologies
IMMS	Instructional Materials Motivation Survey
IP	Introductory programming
LSL	Linden scripting language
MEE	Minecraft education edition
NPC	Non-player characters
RIMMS	Reduced Instructional Materials Motivation Survey
TDIDE	Three-dimensional immersive digital environments
VR	Virtual reality
VW	Virtual worlds

References

1. Groher, I.; Vierhauser, M.; Sabitzer, B.; Kuka, L.; Hofer, A.; Muster, D. Exploring diversity in introductory programming classes: An experience report. In Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Software Engineering Education and Training, Pittsburgh, PA, USA, 22–24 May 2022; pp. 102–112. [CrossRef]
2. Qian, Y.; Lehman, J. Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. *ACM Trans. Comput. Educ.* **2017**, *18*, 1–24. [CrossRef]
3. Fang, X.; Ng, D.T.K.; Leung, J.K.L.; Xu, H. The applications of the ARCS model in instructional design, theoretical framework, and measurement tool: A systematic review of empirical studies. *Interact. Learn. Environ.* **2023**, 1–28. [CrossRef]
4. Li, K.; Keller, J.M. Use of the ARCS model in education: A literature review. *Comput. Educ.* **2018**, *122*, 54–62. [CrossRef]
5. Howard, J.L.; Bureau, J.S.; Guay, F.; Chong, J.X.Y.; Ryan, R.M. Student Motivation and Associated Outcomes: A Meta-Analysis From Self-Determination Theory. *Perspect. Psychol. Sci.* **2021**, *16*, 1300–1323. [CrossRef] [PubMed]
6. Angel Rueda, C.J.; Valdés Godínes, J.C.; Douglas Rudman, P. Categorizing the educational affordances of 3-dimensional immersive digital environments. *J. Inf. Technol. Educ. Innov. Pract.* **2018**, *17*, 83–112. [CrossRef] [PubMed]
7. Angel Rueda, C.J. Diseño de un Modelo didáctico para Introducir al Maestro en el uso de los Mundos Virtuales con Fines Educativos. Ph.D. Thesis, Universidad Autónoma de Querétaro, Santiago de Querétaro, Mexico, 2018.
8. Ghanbarzadeh, R.; Ghapanchi, A.H. Investigating various application areas of three-dimensional virtual worlds for higher education. *Br. J. Educ. Technol.* **2018**, *49*, 370–384. [CrossRef]
9. Huang, Y.C.; Backman, S.J.; Backman, K.F.; McGuire, F.A.; Moore, D. An investigation of motivation and experience in virtual learning environments: A self-determination theory. *Educ. Inf. Technol.* **2019**, *24*, 591–611. [CrossRef]
10. Shonfeld, M.; Greenstein, Y. Factors promoting the use of virtual worlds in educational settings. *Br. J. Educ. Technol.* **2021**, *52*, 214–234. [CrossRef]
11. Arnab, S.; Lim, T.; Carvalho, M.B.; Bellotti, F.; De Freitas, S.; Louchart, S.; Suttie, N.; Berta, R.; De Gloria, A. Mapping learning and game mechanics for serious games analysis. *Br. J. Educ. Technol.* **2015**, *46*, 391–411. [CrossRef]
12. Pellas, N.; Mystakidis, S.; Christopoulos, A. A Systematic Literature Review on the User Experience Design for Game-Based Interventions via 3D Virtual Worlds in K-12 Education. *Multimodal Technol. Interact.* **2021**, *5*, 28. [CrossRef]
13. Pellas, N.; Mystakidis, S. A Systematic Review of Research about Game-based Learning in Virtual Worlds. *J. Univ. Comput. Sci.* **2020**, *26*, 1017–1042. [CrossRef]
14. Scherer, R.; Siddiq, F.; Sánchez Viveros, B. A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions. *Comput. Hum. Behav.* **2020**, *109*, 106349. [CrossRef]
15. Pears, A.; Seidman, S.; Malmi, L.; Mannila, L.; Adams, E.; Bennedsen, J.; Devlin, M.; Paterson, J. A survey of literature on the teaching of introductory programming. In Proceedings of the Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education, Dundee, UK, 25–27 June 2007; pp. 204–223. [CrossRef]

16. Omer, U.; Farooq, M.S.; Abid, A. Introductory programming course: Review and future implications. *PeerJ Comput. Sci.* **2021**, *7*, e647. [CrossRef]
17. Noone, M.; Mooney, A. Visual and textual programming languages: A systematic review of the literature. *J. Comput. Educ.* **2018**, *5*, 149–174. [CrossRef]
18. Kanika; Chakraverty, S.; Chakraborty, P. Tools and Techniques for Teaching Computer Programming: A Review. *J. Educ. Technol. Syst.* **2020**, *49*, 170–198. [CrossRef]
19. López Solórzano, J.G.; Ángel Rueda, C.J. Revisión sistemática de los entornos digitales inmersivos tridimensionales en la enseñanza de la programación. *Rev. De Educ. A Distancia (RED)* **2023**, *23*, 1–24. [CrossRef]
20. Esteves, M.; Fonseca, B.; Morgado, L.; Martins, P. Improving teaching and learning of computer programming through the use of the Second Life virtual world. *Br. J. Educ. Technol.* **2011**, *42*, 624–637. [CrossRef]
21. Cooper, S.; Dann, W.; Pausch, R. Alice: A 3-D tool for introductory programming concepts. *J. Comput. Sci. Coll.* **2000**, *15*, 107–116.
22. Reisoğlu, I.; Topu, B.; Yılmaz, R.; Karakuş Yılmaz, T.; Göktaş, Y. 3D virtual learning environments in education: A meta-review. *Asia Pac. Educ. Rev.* **2017**, *18*, 81–100. [CrossRef]
23. Au, W.J. *Making a Metaverse That Matters: From Snow Crash & Second Life to a Virtual World Worth Fighting for*; John Wiley and Sons: Hoboken, NJ, USA, 2023.
24. Gregory, S.; Scutter, S.; Jacka, L.; McDonald, M.; Farley, H.; Newman, C. Barriers and Enablers to the Use of Virtual Worlds in Higher Education: An Exploration of Educator Perceptions, Attitudes and Experiences. *J. Educ. Technol. Soc.* **2015**, *18*, 3–12.
25. Büşra, A.Ç.; Ediz, Ç. Top concerns of user experiences in Metaverse games: A text-mining based approach. *Entertain. Comput.* **2023**, *46*, 100576. [CrossRef]
26. Prensky, M. Digital Natives, Digital Immigrants. *Horizon* **2001**, *9*, 6.
27. Robins, A.; Rountree, J.; Rountree, N. Learning and Teaching Programming: A Review and Discussion. *Comput. Sci. Educ.* **2010**, *13*, 137–172. [CrossRef]
28. Shokaliuk, S.; Bohunenko, Y.; Lovianova, I.; Shyshkina, M. Technologies of distance learning for programming basics on the principles of integrated development of key competences. *CTE Workshop Proc.* **2020**, *7*, 548–562.
29. Rico, M.; Oz, G.M.M.; Alaman, X.; Camacho, D.; Pulido, E. A Programming Experience of High School Students in a Virtual World Platform. *Int. J. Eng. Educ.* **2011**, *27*, 52.
30. Hulsey, C.; Pence, T.B.; Hodges, L.F. Camp CyberGirls: Using a virtual world to introduce computing concepts to middle school girls. In Proceedings of the 45th ACM Technical Symposium on Computer Science Education, Denver, CO, USA, 6–9 March 2013; ACM: New York, NY, USA, 2014; pp. 331–336. [CrossRef]
31. Kao, D. JavaStrike: A Java programming engine embedded in virtual worlds. In Proceedings of the 14th International Conference on the Foundations of Digital Games, San Luis Obispo, California, 26–30 August 2019; ACM: New York, NY, USA, 2019; pp. 1–5. [CrossRef]
32. Kutay, E.; Oner, D. Coding with Minecraft: The Development of Middle School Students' Computational Thinking. *ACM Trans. Comput. Educ.* **2022**, *22*, 1–19. [CrossRef]
33. Aktaş Kumral, H.; Çam, E. The Effect of Coding Education with Minecraft EDU on Students' Problem Solving Skills and Their Attitudes towards Coding. *Uluslararası Türk Eğitim Bilim. Derg.* **2023**, *2023*, 1–40. [CrossRef]
34. Mørch, A.I.; Andersen, R. Programming with Minecraft Bedrock Up: Modeling, Coding, and Computational Concepts. In *End-User Development*; Spano, L.D., Schmidt, A., Santoro, C., Stumpf, S., Eds.; Lecture Notes in Computer Science; Springer Nature: Cham, Switzerland, 2023; Volume 13917, pp. 230–240. [CrossRef]
35. Bile, A. Development of intellectual and scientific abilities through game-programming in Minecraft. *Educ. Inf. Technol.* **2022**, *27*, 7241–7256. [CrossRef]
36. Sajjanhar, A.; Faulkner, J. Second life as a learning environment for computer programming. *Educ. Inf. Technol.* **2019**, *24*, 2403–2428. [CrossRef]
37. Demirkiran, M.; Tansu, F. An investigation on primary school students' dispositions towards programming with game-based learning. *Educ. Inf. Technol.* **2021**, *26*, 3871–3892. [CrossRef]
38. Alsaadi, B.; Alsaadi, B.; Alghamdi, A.; Alfhaid, M.; Almuallim, N.; Meccawy, M. Learning While Playing: Introducing Programming Concepts to Children in Minecraft. *Int. J. Online Biomed. Eng. (iJOE)* **2022**, *18*, 4–24. [CrossRef]
39. Vosinakis, S.; Anastassakis, G.; Koutsabasis, P. Teaching and learning logic programming in virtual worlds using interactive microworld representations. *Br. J. Educ. Technol.* **2018**, *49*, 30–44. [CrossRef]
40. Schunk, D.H.; Meece, J.L.; Pintrich, P.R. *Motivation in Education: Theory, Research, and Applications*, 4th ed.; Pearson New International Edition; Always Learning; Pearson: London, UK, 2014.
41. Cook, D.A.; Artino, A.R. Motivation to learn: An overview of contemporary theories. *Med. Educ.* **2016**, *50*, 997–1014. [CrossRef]
42. Keller, J.M. Development and use of the ARCS model of instructional design. *J. Instr. Dev.* **1987**, *10*, 2–10. [CrossRef]
43. Garzón, J.; Kinshuk; Baldiris, S.; Gutiérrez, J.; Pavón, J. How do pedagogical approaches affect the impact of augmented reality on education? A meta-analysis and research synthesis. *Educ. Res. Rev.* **2020**, *31*, 100334. [CrossRef]
44. Choi, W.C.; Lei, H.; Mendes, A.J. Motivating the New Generation: Using Flipped Classroom and ARCS Model to Enhance Block-Based Programming Education. In Proceedings of the 2023 IEEE Frontiers in Education Conference (FIE), College Station, TX, USA, 18–21 October 2023; pp. 1–8. [CrossRef]

45. Girvan, C.; Savage, T. Identifying an appropriate pedagogy for virtual worlds: A Communal Constructivism case study. *Comput. Educ.* **2010**, *55*, 342–349. [CrossRef]
46. Girvan, C.; Savage, T. Virtual worlds: A new environment for constructionist learning. *Comput. Hum. Behav.* **2019**, *99*, 396–414. [CrossRef]
47. Girvan, C.; Tangney, B.; Savage, T. SLurtles: Supporting constructionist learning in Second Life. *Comput. Educ.* **2013**, *61*, 115–132. [CrossRef]
48. Papert, S.A. *Mindstorms*; Hachette: London, UK, 2020; OCLC: 1239803576.
49. Papert, S. A Critique of Technocentrism in Thinking about the School of the Future. In *Children in the Information Age*; Elsevier: Berlin/Heidelberg, Germany, 1988; pp. 3–18. [CrossRef]
50. Duncan, I.; Miller, A.; Jiang, S. A taxonomy of virtual worlds usage in education. *Br. J. Educ. Technol.* **2012**, *43*, 949–964. [CrossRef]
51. Mystakidis, S.; Berki, E.; Valtanen, J.P. Deep and Meaningful E-Learning with Social Virtual Reality Environments in Higher Education: A Systematic Literature Review. *Appl. Sci.* **2021**, *11*, 2412. [CrossRef]
52. Pellas, N.; Kazanidis, I.; Konstantinou, N.; Georgiou, G. Exploring the educational potential of three-dimensional multi-user virtual worlds for STEM education: A mixed-method systematic literature review. *Educ. Inf. Technol.* **2017**, *22*, 2235–2279. [CrossRef]
53. Yu, Z.; Deng, X. A Meta-Analysis of Gender Differences in e-Learners' Self-Efficacy, Satisfaction, Motivation, Attitude, and Performance across the World. *Front. Psychol.* **2022**, *13*, 897327. [CrossRef] [PubMed]
54. Yu, Z. The effects of gender, educational level, and personality on online learning outcomes during the COVID-19 pandemic. *Int. J. Educ. Technol. High. Educ. Vol.* **2021**, *18*, 14. [CrossRef] [PubMed]
55. Chen, S.; Yang, S.J.; Hsiao, C. Exploring student perceptions, learning outcome and gender differences in a flipped mathematics course. *Br. J. Educ. Technol.* **2016**, *47*, 1096–1112. [CrossRef]
56. Choi, G.; Chung, H.; Kim, Y. Are Stereotypes Relative to Gender Usage Applicable to Virtual Worlds? *Int. J.-Hum.-Comput. Interact.* **2012**, *28*, 399–405. [CrossRef]
57. Lin, M.; Tutwiler, M.S.; Chang, C. Gender bias in virtual learning environments: An exploratory study. *Br. J. Educ. Technol.* **2012**, *43*, E59–E63. [CrossRef]
58. Korlat, S.; Kollmayer, M.; Holzer, J.; Lüftenegger, M.; Pelikan, E.R.; Schober, B.; Spiel, C. Gender Differences in Digital Learning during COVID-19: Competence Beliefs, Intrinsic Value, Learning Engagement, and Perceived Teacher Support. **2021**, *12*, 637776. [CrossRef] [PubMed]
59. Keller, J.M. *Motivational Design for Learning and Performance: The ARCS Model Approach*; Springer: New York, NY, USA, 2010. [CrossRef]
60. Loorbach, N.; Peters, O.; Karreman, J.; Steehouder, M. Validation of the Instructional Materials Motivation Survey (IMMS) in a self-directed instructional setting aimed at working with technology. *Br. J. Educ. Technol.* **2015**, *46*, 204–218. [CrossRef]
61. Hernández Moreno, L.A.; López Solórzano, J.G.; Tovar Morales, M.T.; Vergara Villegas, O.O.; Cruz Sánchez, V.G. Effects of using mobile augmented reality for simple interest computation in a financial mathematics course. *PeerJ Comput. Sci.* **2021**, *7*, e618. [CrossRef]
62. Cipollone, M.; Schifter, C.C.; Moffat, R.A. Minecraft as a Creative Tool: A Case Study. *Int. J. Game-Based Learn.* **2014**, *4*, 1–14. [CrossRef]
63. Nkadimeng, M.; Ankiewicz, P. The Affordances of Minecraft Education as a Game-Based Learning Tool for Atomic Structure in Junior High School Science Education. *J. Sci. Educ. Technol.* **2022**, *31*, 605–620. [CrossRef]
64. Chua, H.W.; Yu, Z. A systematic literature review of the acceptability of the use of Metaverse in education over 16 years. *J. Comput. Educ.* **2023**, *11*, 615–665. [CrossRef]
65. Hamad, A.; Jia, B. How Virtual Reality Technology Has Changed Our Lives: An Overview of the Current and Potential Applications and Limitations. *Int. J. Environ. Res. Public Health* **2022**, *19*, 11278. [CrossRef] [PubMed]
66. Solomon, C.J.; Papert, S. A case study of a young child doing turtle graphics in LOGO. In Proceedings of the National Computer Conference and Exposition on—AFIPS '76, New York, NY, USA, 7–10 June 1976; p. 1049. [CrossRef]
67. Ghanbarzadeh, R.; Ghapanchi, A.H. Uncovering educational outcomes deriving from students' acceptance and involvement with 3D virtual worlds. *Educ. Inf. Technol.* **2021**, *26*, 311–337. [CrossRef]
68. Bar-El, D.; E. Ringland, K. Crafting Game-Based Learning: An Analysis of Lessons for Minecraft Education Edition. In Proceedings of the International Conference on the Foundations of Digital Games, Bugibba, Malta, 15–18 September 2020; pp. 1–4. [CrossRef]
69. Hopcan, S.; Polat, E.; Albayrak, E. Whether to flip Extreme Apprenticeship: Which is more effective in programming instruction? *Educ. Inf. Technol.* **2022**, *27*, 10731–10756. [CrossRef] [PubMed]
70. Tsai, C.Y. Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. *Comput. Hum. Behav.* **2019**, *95*, 224–232. [CrossRef]
71. Kozlova, D.; Pikhart, M. The Use of ICT in Higher Education from the Perspective of the University Students. *Procedia Comput. Sci.* **2021**, *192*, 2309–2317. [CrossRef]
72. Rahman, M.H.A.; Phon, D.N.E.; Utama, N.I.; Halim, N.D.A.; Kasim, S. Sense of Presence and Learning Satisfaction among Students of Different Age Groups in a 3-D Virtual World. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2019**, *9*, 711–716. [CrossRef]

73. Wang, S.T.; Liu, L.M.; Wang, S.M. The Design and Evaluate of Virtual Reality Immersive Learning—The Case of Serious Game "Calcium Looping for Carbon Capture". In Proceedings of the 2018 International Conference on System Science and Engineering (ICSSE), New Taipei City, Taiwan, 15 March 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–4. [CrossRef]
74. Zorn, C.; Wingrave, C.A.; Charbonneau, E.; Laviola, J.J. Exploring Minecraft as a conduit for increasing interest in programming. In Proceedings of the International Conference on Foundations of Digital Games, Chania, Crete, Greece, 14–17 May 2013.
75. AlJanah, S.; Teh, P.S.; Tay, J.Y.; Aiyenitaju, O.; Nawaz, R. Minecraft as a Tool to Enhance Engagement in Higher Education. In *Research and Innovation Forum 2022*; Visvizi, A., Troisi, O., Grimaldi, M., Eds.; Springer Proceedings in Complexity; Springer International Publishing: Cham, Switzerland, 2023; pp. 465–476. [CrossRef]
76. Deloatch, R.; Bailey, B.P.; Kirlik, A. Measuring Effects of Modality on Perceived Test Anxiety for Computer Programming Exams. In Proceedings of the 47th ACM Technical Symposium on Computing Science Education, Memphis, TN, USA, 2–5 March 2016; pp. 291–296. [CrossRef]
77. Gómez-Gonzalvo, F.; Molina, P.; Devís-Devís, J. Which are the patterns of video game use in Spanish school adolescents? Gender as a key factor. *Entertain. Comput.* **2020**, *34*, 100366. [CrossRef]
78. Joiner, R.; Iacovides, J.; Owen, M.; Gavin, C.; Clibbery, S.; Darling, J.; Drew, B. Digital Games, Gender and Learning in Engineering: Do Females Benefit as Much as Males? *J. Sci. Educ. Technol.* **2011**, *20*, 178–185. [CrossRef]
79. Tellhed, U.; Björklund, F.; Kallio Strand, K. Sure I can code (but do I want to?). Why boys' and girls' programming beliefs differ and the effects of mandatory programming education. *Comput. Hum. Behav.* **2022**, *135*, 107370. [CrossRef]
80. Erol, O.; Kurt, A.A. The effects of teaching programming with scratch on pre-service information technology teachers' motivation and achievement. *Comput. Hum. Behav.* **2017**, *77*, 11–18. [CrossRef]
81. Pérez-Jorge, D.; Martínez-Murciano, M.C. Gamification with Scratch or App Inventor in Higher Education: A Systematic Review. *Future Internet* **2022**, *14*, 374. [CrossRef]