computers

MDPI

Article

The PECC Framework: Promoting Gender Sensitivity and Gender Equality in Computer Science Education

Bernadette Spieler [1,*] and Carina Girvan [2,3]

[1] Centre for Education and Digital Transformation, Zurich University of Teacher Education, 8090 Zurich, Switzerland
[2] School of Social Sciences, Cardiff University, Cardiff CF10 3WT, Wales, UK; girvanc@tcd.ie
[3] School of Education, Trinity College Dublin, The University of Dublin, D02 PN40 Dublin, Ireland
* Correspondence: bernadette.spieler@phzh.ch

Abstract

There are increasing expectations that we should live in a digitally and computationally literate society. For many young people, particularly girls, school is the one place that provides an opportunity to develop the necessary knowledge and skills. This environment can either perpetuate and reinforce or eliminate existing gender inequalities. In this article, we present the "PLAYING, ENGAGEMENT, CREATVITIY, CREATING" (PECC) Framework, a practical guide to supporting teachers in the design of gender-sensitive learning activities, bringing students' own interests to the fore. Through a six-year, mixed-methods, design-based research approach, PECC—along with supporting resources and digital tools—was developed through iterative cycles of theoretical analysis, empirical data (both qualitative and quantitative), critical reflection, and case study research. Exploratory and instrumental case studies investigated the promise and limitations of the emerging framework, involving 43 teachers and 1453 students in secondary-school classrooms (including online during COVID-19) in Austria, Germany, and Switzerland. Quantitative data (e.g., surveys, usage metrics) and qualitative findings (e.g., interviews, observations, classroom artefacts) were analyzed across the case studies to inform successive refinements of the framework. The case study results are presented alongside the theoretically informed discussions and practical considerations that informed each stage of PECC. PECC has had a real-world, tangible impact at a national level. It provides an essential link between research and practice, offering a theoretically informed and empirically evidenced framework for teachers and policy makers.

Keywords: inclusion; gender; computer science; secondary education

Academic Editor: Paolo Bellavista

Received: 1 April 2025
Revised: 29 May 2025
Accepted: 17 June 2025
Published: 25 June 2025

Citation: Spieler, B.; Girvan, C. The PECC Framework: Promoting Gender Sensitivity and Gender Equality in Computer Science Education. *Computers* **2025**, *14*, 249. https://doi.org/10.3390/computers14070249

1. Introduction

Globally, the first two decades of the 21st century witnessed ever-increasing calls for digital literacy and computer science (CS) to become components of every child's education. In many countries, these calls came from industry, particularly in Europe, which saw a shortage of qualified applicants for jobs [1]. This pressure finally put these subjects on the curriculum, despite being advocated for many years previously by various groups, including educators themselves. As a result, the importance of CS education has been heavily discussed in the media and by governments. The underrepresentation of women in the computing industry is an important dimension in these discussions, as is the underrepresentation of women in CS and related degree programs. Jobs in software engineering are clearly male-dominated, as illustrated in the statistical data produced by [2]. In 2021, 81.1% of those employed as ICT professionals in the EU were men, compared to 18.9% of women. This disparity is even more pronounced in Austria, Germany, and Switzerland.

In response to the demand for more IT professionals and a future society that is digitally literate, internationally, we have seen changes in education, including the introduction of new compulsory subjects and courses. For example, a computing curriculum was introduced in England [3], CS was made a mandatory subject in many states of Germany [4], and the Swiss "Curriculum 21" was introduced, with the obligatory module "Media and Informatics" [5]. Furthermore, opportunities for the cross-curricular teaching of skills have been developed, for example, in the Welsh Digital Competence Framework, first introduced in 2016 [6], and Digital Basic Education in Austria, starting in 2022 [7]. Additionally, there has been a surge in out-of-school programs, e.g., "Computer Science for All" in the USA [8] and different coding clubs around the world (e.g., Raspberry Pi, CodeClub, CoderDojo, Roberta).

However, current curriculum initiatives for CS do not consider important gender differences in interest, self-efficacy, or sense of belonging [9]. It is widely recognized that gender gaps in CS do not start at the industry or university level but between the ages of 12 to 15 [10,11], or even earlier in childhood [12,13], through societal, cultural or parental biases, which may lead to girls' self-censorship and lower engagement in science, technology, engineering and maths (STEM) subjects in general, including CS subjects [14,15].

The desire to be socially connected and respected is a strong initial motivation to study a certain field [16]. Consequently, young women and girls question whether their abilities and interests are harmonious with the field of computing [17,18]. This is underpinned by predominant stereotypes in CS and the absence of role models in this field [19–21]. In fact, the gender gap has been found to widen as children get older. This was illustrated in a report by the American Association of University Women 2018, which found that 31% of girls between 11–13 in the US describe jobs in coding and programming as "not for them" [22]. By 14–18 years old, the percentage jumps to 40%, and by 18 or 19, 58% of girls share this opinion.

While there are a lot of coding resources available for teachers, e.g., eLearning courses (e.g., iMoox.at, Udemy.com), unplugged activities (e.g., Bebras Challenge; CS Unplugged), or guides (e.g., European Code Week, Codesters), they are mostly task- or action-focused and rarely consider a gender dimension. At the same time, many teachers who have not studied CS themselves and are expected to teach this subject in an interdisciplinary way have little or no previous experience of training in CS and are therefore reliant on such resources or concepts. There are, however, some initiatives that specifically seek to engage women and people of color in the design of their curriculum materials (such as Code.org) and feature inspiring videos and role models. Furthermore, Roberta is an educational robotics program in Europe which purposefully engages students with a gender-sensitive view [23]. The didactic material and course concept are described as gender-balanced, with the aim to raise girls' interest in technical topics and strengthen girls' self-confidence. While such initiatives are a welcome development that can change teachers' perspectives on programming activities, they are rare and niche examples.

Within this broader context, the first author was engaged in the "No One Left Be- hind" (NOLB) project (European Commission, Horizon 2020, Grant Agreement Number 645215), which focused on creating inclusive classrooms to enable digital game development and experiences for primary- and secondary-school students as part of their formal education. Over the two and a half years of the project, interactions with teachers and students as well as our research evidence pointed towards a need to develop new tools, new game-based teaching and learning resources, and a framework specifically to help teachers to create gender-conscious CS activities in their classrooms. This was the impetus which led to the initial development of the "PLAYING, ENGAGEMENT, CREATVITIY, Creating" (PECC) Framework, which aims to (1) provide a theoretical model for teachers and trainers who want to design computer science activities according to gender-sensitive standards; (2) to generate useful resources with which to implement the PECC Framework; and (3) to measurably change pupils' and especially girls' intrinsic motivation to engage in CS education.

As Anderson and Shattuck (2012) note, "design-based interventions are rarely if ever designed and implemented perfectly; thus, there is always room for improvements in the design and subsequent evaluation" [24]. This held true for PECC, which was iteratively developed following a design-based research (DBR) approach [25] over six years from 2017 to 2023. This method was interventionist, systematic, naturalistic, and iterative, and through it artefacts and practices were developed and trialed. This paper charts the development of the PECC Framework, identifying the promise and limitations of the framework through empirical research. Critical, reflective dialogue between the authors and with teachers was a key component of this process, as was close engagement with the existing theoretical and empirical literature. We begin by discussing the pertinent background literature, outlining both the need for gender-conscious pedagogy in CS education and the theoretical foundations of PECC. Following a discussion of DBR, we draw on the description by diSessa and Cobb (2004) of their own development of ontological innovations to structure our presentation of the development of PECC, including the results of empirical studies in multiple contexts, reference to theory and our own reflective discussions [26]. Finally, we explore the real-world impact of PECC in classrooms and policy to date and consider the limitations and future potential of the framework.

Although the PECC framework is based on gender-sensitive pedagogy and design-based research, it also contributes to the field of intelligent learning environments. Specifically, the framework includes digital platforms such as Pocket Code and the Create@School app, which enable interactive, game-based, and data-driven learning. These tools enable personalized feedback and promote learning motivation, which is in line with the goals of adaptive and responsive computer systems. Furthermore, PECC takes into account the human-centered dimension of intelligent educational technologies by promoting inclusive design and equal access, especially for underrepresented groups in the field of computer science. In this way, PECC offers a practical approach to designing inclusive and intelligent learning environments.

The aim of this paper is to present the development and refinement of the PECC Framework as a gender-sensitive pedagogical approach for computing education. The objectives are (1) to describe the design-based research process through which the PECC Framework was iteratively developed, (2) to identify and analyze the strengths and limitations of the framework based on empirical evidence from multiple case studies, (3) to show how theoretical insights, empirical data, and practitioner reflections were combined to inform the framework's evolution, and (4) to illustrate the practical and policy-level impact of the framework in real-world educational settings.

2. Literature Review

The initial development of the PECC Framework was rooted in early empirical evidence from the NOLB project and existing evidence from a review of female students in CS education, focusing on understanding stereotypes, negative effects, and positive motivators [9]. As the framework has developed over time, the authors have engaged in dialogue over a range of theoretical ideas and empirical studies. Those that are key to the development of PECC are outlined below, beginning, as this research did, by rethinking gender dimensions in CS classrooms to illustrate the complexity of inclusive learning environments, before moving on to outline the rationale for focusing on games and how constructionism acts as a guiding and underpinning pedagogic theory throughout the development of the PECC Framework.

2.1. Gendered Engagement

There are many factors which influence a person's education and career trajectories over a lifetime. Within CS, it is well established that there is a combination of social, psychological, and structural factors that compound the current gender gap. Varma (2010) describes how teachers' and parents' own lack of knowledge about IT-related professions can result in the transmission of false, stereotyped ideas to children and young people [27]. Stereotypes can be a critical factor in how people envisage a role and whether they envisage themselves fulfilling that role or not [28]. While female role models have the potential to address stereotypes directly and to actively counteract them, weaken, and possibly even dissolve them [29], we argue the very rare appearance of female role models can also strengthen the idea that a female computer scientist is not the norm. Influenced by their teachers, parents, peers, and popular media, a young person's sense of self in relation to a stereotype is, therefore, likely to encourage or discourage them to pursue a subject or career. In the case of CS, Cheryan (2012) found that the stereotype of a computer scientist was not compatible with the self-concept of many young girls [30]. At the same time, Hur, Andrzejewski, and Marghitu (2017) found that girls were less likely to be exposed to programming-related activities and were generally less likely to be encouraged to engage with computers and programming by parents than boys [31]. However, they also argue that early engagement with programming activities is essential when it comes to getting young people, and particularly girls, involved in this area later on.

2.2. Gender and Pedagogy

Given the importance of early participation in programming activities to engage girls, but also considering the prevailing stereotypes of computer scientists, we considered how teachers can design learning experiences to best support girls to see themselves as computer scientists. It is important to note at this point that we view formal education as the most viable route to engage all young people. While there are many successful after-school and out-of-school activities or clubs, these are simply inaccessible to portions of society and attendance can be influenced by parents and peers [32].

Happe et al. (2021) note that teachers need certain guidelines for curriculum design, but that a significant challenge is teachers' lack of awareness of how to utilize the diversity of students in the CS classroom. They highlight how some teachers may unknowingly create a learning environment that favors male students and leaves female students feeling less competent [11]. While the curriculum may be largely gender-neutral, teachers themselves may have internalized implicit or explicit gender biases [33]. No one is free from prejudice, but it is important to be aware of one's own. In 1995, Connell defined the term "hegemonic masculinity," describing how society associates men primarily with power and economic achievement [34]. The "nerd identity" in CS, represents competence in using computers as a form of masculine ability. In addition, studies indicate that teachers are more likely to associate boys with the concepts of logic, competition, and independence than girls [28,35–39] and that these gender biases are implicitly passed on in the classroom [40] and shape students' attitudes and motivation [40,41]. Lindner et al. (2022) found that female teachers in Switzerland shared gender-specific beliefs about their students' ability in mathematics and future careers that fit with stereotypical norms, which assume male students are more interested and more competent at mathematics [18]. This reflects Makarova's (2019) study that found that female students viewed mathematics and physics as male subjects [42].

Additionally, while the curriculum may be gender-neutral, teaching materials are often not and use male-centered representation in images and language [43,44]. Several studies point to the importance of showing a realistic image of women in technology (i.e., real role models) and portraying women in a visible and audible way [45]. Language sensitivity is especially important in languages with gender-specific nouns, such as German or Italian, in order to address all people equally. Wetschanow (2008) points out that language is an important and powerful means of producing 'gender' [46].

To address these issues, the authors of this paper have explored the utilization of a "gender-sensitive pedagogy" [47]. In practice, gender-sensitive pedagogy means encouraging children to explore and create their own learning situation [48], as well as performance-based praise, reflection on the attention paid, interventions in the project phases, and designing "gender-neutral" tasks [49]. In addition, teachers should encourage discussions and dialogues that focus on individual experiences and understanding [50]. In this way, a "safe" environment can be created for those who have little prior knowledge of CS. In addition, many children have an unrealistic idea of technical professions, which affects children's self-efficacy [17,51]. Thus, gender-conscious approaches have been embedded within the PECC Framework and illustrated through practical guidance for teachers.

2.3. Gender-Sensitive Pedagogy and Games

While a gender-neutral approach is advocated, there are also arguments for addressing the gender imbalance by focusing on activities which are more favorable to girls. In today's digital world, digital games are becoming increasingly important and can make a valuable contribution in education, bringing many advantages, but also naturally challenges. The annual JAMES study (Youth, Activities, Media—Swiss Survey, [52]) shows that video games play an important role in the daily lives of children and adolescents: 60% of children and adolescents in Switzerland play daily or several times a week. About three quarters of the young players spend an average of 80 min on video games on weekdays. On weekends, the playing time is twice as high. In this sense, there is much to suggest the value of integrating games as anchors into teaching. The JAMES study also shows that gender plays an important role (65% of girls play compared to 93% of boys). Other factors such as socioeconomic status, origin, and type of school, on the other hand, have little influence. Therefore, the desires of girls should be specifically considered. One of the most frequently mentioned advantages is the motivation and fascination that come from playing games. By integrating games into teaching, a varied learning environment can be created.

Games can strengthen self-efficacy and self-confidence by offering challenges and motivating students to persevere and constantly improve [53]. This motivation can make the learning process more effective and sustainable. Since students are learning while doing something they enjoy, they are more motivated to actively participate in the learning process. Furthermore, games can assist students in developing important skills such as problem-solving, decision-making, or critical thinking. Many games require students to solve complex problems, make decisions, and evaluate the consequences of their actions. These skills are not only important in the school context, but also in their later professional life.

While professional game development is about creating an optimized, holistic product on a large scale, the requirements for creating digital games in primary and secondary schools can and must be scaled down. In his book "The Art of Game Design," Schell (2020) reasons that small games consist of 4 basic elements: mechanics, story, aesthetics, and technology [54]. Brathwaite and Schreiber (2008) further narrow down the development of a game to two competencies: the development of the rules of a game and the content of a game [55]. In that way, game design can encompass various disciplines, such as developing the game idea and story, programming the game, creating visuals and sounds, and setting the rules that define a game. To understand what it means to be a game designer, necessary game elements should be explained (e.g., game genres or mechanics, see [56,57]), which resonates with the argument that classroom interventions for girls should focus on providing guidance and tutorials [58]. Female students often find small group tutorials more useful than male students find them and enjoy them because they accomplish small steps while they get one step closer to the end of the program. Thus, a scaffolded approach to game design is seen as effective for helping female students to develop fundamental computing concepts by completing almost-finished games [59]. When considering the characteristics of problem-solving strategies, Happe et al. (2021) found that girls and boys approach problem-solving differently [11]. While girls tend to define problems more broadly, boys tend to approach problems in isolation. This broad approach can put more pressure on girls as they strive to consider every detail. Annis and Nesbitt (2017) suggested that an effective strategy might be a mix of both approaches [60]. In addition, the holistic thinking attributed to girls and women is related to the concept of Computational Thinking (CT), an important aspect of our technology-driven society. CT is about organizing extensive information into complex patterns and identifying connections—competencies attributed to girls and women.

2.4. Constructionism and PECC

"Every maker of video games knows something that the makers of curriculum don't seem to understand. You'll never see a video game being advertised as being easy. Kids who do not like school will tell you it's not because it's too hard. It's because it's boring" —(Papert, 1993, p. 17) [61].

Constructionism is the underpinning pedagogic theory of the PECC Framework, informing a clear branch of the literature on CS education. Constructionist learning activities focus on the iterative construction of personally meaningful and shareable artifacts (whether a robot, a game, a song, etc.) which are created to explore, test, and extend understanding [62]. Here, we outline how constructionism informs the four key elements of PECC, PLAYING, ENGAGEMENT, CREATVITIY, and CREATING, concepts which were returned to on many occasions when discussing empirical results and proposed changes.

Within a constructionist paradigm, games and the programming of games have been found to provide an effective medium for teachers to motivate students in general to interact, be creative, communicate, and learn [63]. More broadly, the influence of constructionism is clear (if not explicit) in concepts such as CT and "tinkering", the Maker movement, and in many national and global extracurricular initiatives, such as Code.org, CodeClub and CoderDojos. In the context of the Maker scene, tinkering refers to trying things (including coding) without the fear of failure in a playful way [64], which in turn informs the PLAYING element of PECC. Another approach to the introduction of CS underscores playfulness and is influenced by constructionism in "CS Unplugged". CS Unplugged activities, which provide a way to teach fundamental principles of CS in a playful way without actually using a computer, e.g., by using paper, cards, string, crayons, and movement [65].

In our digitized world, it is increasingly important to educate young people to become more than passive consumers of digital media, giving them the skills and tools to become active creators [66]. This is where the concept of Making comes into play, a pedagogical approach that focuses on action-oriented learning and thus sharpens the bridge between digital and analog learning content [67,68]. In this context, the ability to use technological tools in a creative and self-determined way becomes key to future educational and career opportunities [69]. Active and action-oriented learning in both analog and digital realms is promoted in an enjoyable and sustainable way. Making offers students the opportunity to develop hands-on skills in a self-directed learning environment [68]. For teachers, Making can show an innovative approach to stimulate interest through creative design or digital projects. It allows people to create their own projects using different tools [70]. Above all, this appeals to different interests. The state of research on Maker Education is diverse and growing. Studies have shown that Maker Education can increase learners' engagement, creativity, and interest in STEM subjects. In addition, Making embeds learning in authentic contexts and helps learners to see the relevance and application of what they are learning. While the Maker movement is gaining traction in the private sector through FabLabs or out-of-school makerspaces, opportunities for Maker Education are increasingly being found. For example, the Swiss Curriculum 21 calls for computer science to be taught through independent discovery and activity-based tasks [5].

Communication between students about their work, and the process of learning with peers, teachers, and collaborators, is seen as an indispensable part of a student's learning experience [61]. Knowing that the artefacts created will be shared with others, and not just a teacher, can impact a student's motivation and the design of their artefact [71]. This sharing could occur within a group before sharing takes place externally with others, highlighting the importance of team-based learning and a safe learning environment within the 'ENGAGEMENT' element of the PECC Framework. Additionally, collaborative team-based activities can help to address the stereotype of the 'lone programmer'. That students' constructions, or artefacts, should be personally meaningful is associated with 'ENGAGEMENT', 'CREATVITIY' and the motivational element of 'interest'. Creativity is pervasive in constructionist activities [72], which is apparent in the iterative approaches encouraged by teachers and used by students. With this comes risk-taking and recognition that there can be more than one 'answer' to a problem, encouraging the learner to return and revise their solution as their ideas and knowledge develop.

Constructionism has also informed the design of many tools for use in the classroom, including tools created as part of the design-based research process described herein. Constructionist tools such as Pocket Code [73] and Scratch [74] are designed to support learners engaging in construction by providing them with "low-floor" (easy-to-use) tools [75]. At the same time, these tools are characterized as having "high ceilings" in that they are powerful tools and can be used by someone new to programming and by those who are experienced and seeking to create something complex. Another popular tool for constructionist learning is the microworld [76]. Both affective and behavioral engagement increased when students used constructionist microworlds to create their own concrete models of abstract physics concepts.

3. Methodology

3.1. Overview of DBR

Design-based research (DBR) provides a pragmatic approach to research which is interventionist, theory-producing, naturalistic, and iterative [25,26] on the one-hand, and results in artefacts and practices which are impactful on both teaching and learning on the other [77]. That is to say, it creates a bridge between research and practice with the intention of having a real-world impact alongside theory development, something which Brown (1992) characterizes as an ongoing and critical tension in design experiments [78]. Unlike traditional experimental approaches with control and treatment groups, DBR does not aim for isolated causal attribution under controlled conditions. Instead, it produces theoretically grounded and empirically informed insights through iterative cycles of intervention, reflection, and refinement in real-world contexts. The ambition behind DBR is the development of effective interventions which can be successfully transposed from the initial intervention or experimental classroom to an 'average' classroom that does not rely on the engagement of the researcher. It is the development of theory that allows for this transposition.

DBR practices involve the development of tools, curricula, and theories and, as a result, the generation of findings that are validated through the consequences of their use, namely an impact on the practice and resources that can be adopted elsewhere. It is in this vein that the PECC Framework has been iteratively developed over six years with initial work on the development of technological tools such as Pocket Code [73] and game-based learning experiences [79], followed by the use and further refinement of PECC in the context of emergency remote education during the COVID-19 pandemic[80].

Barab and Squire (2004) state the goal of DBR as being "to lay open and problematize the completed design and resultant implementation in a way that provides insight into the local dynamics [77]. This involves not simply sharing the designed artefact, but providing rich descriptions of context, guiding and emerging theory, design features of the intervention, and the impact of these features on participation and learning" (p. 8) [77]. This has been achieved in the development of PECC through the use of case studies and experimental pilots, described in multiple publications and professional development resources, for both academic audiences and practitioners. These pilots are referenced in the following sections.

To us, part of the problematization of the completed design, implementation, and results requires critical reflection and engagement with different actors. In the case of the latter, there was repeated engagement with teachers involved in the implementations, who provided feedback to the researchers. The former resulted in the involvement of the second author, who was not involved in any of the implementations and therefore provided an outsider perspective on the results and planned changes to the model, resulting in many in-depth discussions about the purpose of PECC, the implications of results, and how planned changes resonated (or not) with the guiding theoretical concepts and empirical research findings from allied areas.

3.2. Research Overview

Results from the "No One Left Behind" (NOLB) project, raised the question of gender equality in CS education, following on from the aim of identifying differences in programming games and approaches between girls and boys. The lead author of this article was significantly involved in the Austrian arm of the pilot project whilst also undertaking her doctoral studies [81]. As part of these studies and informed by her experience on the NOLB project, she identified a clear need for a theoretical framework for teachers and trainers who want to design CS activities according to gender-sensitive standards; to generate useful resources with which to implement the PECC Framework; and to effect measurable changes in girls' intrinsic motivation to study and pursue careers in CS.

The primary goal of the research described herein was to increase gender equality in CS education in schools. The PECC Framework emerged as a theoretical framework for gender-sensitive computer science education, with accompanying teacher guidelines, tools and resources. It was developed through multiple interventions in multiple naturalistic settings, described in the following section.

Although this research does not include a classical pre/post-test design with statistical comparison between matched groups, we systematically collected and analyzed both quantitative (e.g., pre/post-surveys, usage metrics) and qualitative (e.g., interviews, observations, artefacts) data throughout the different iterations of PECC. These data informed both the theoretical development and the practical refinement of the framework. The research is replicable through the described materials, study settings, instruments, and phases, and builds on recognized DBR validation strategies such as triangulation, transparency, and iterative theory–practice alignment.

The gradual refinement of the PECC Framework spanned six years, from 2017 to 2023, and involved an extensive network of 1453 secondary-school students and 43 teachers in three countries—Austria, Germany, and Switzerland. During this period, several projects were initiated and implemented that interacted with each other and were consistent with the PECC philosophy. Following ethical approval from the first author's institutions, we gained permission from the school and teacher to conduct each study, and legal guardians of the children were informed of the research and provided their informed consent. Data were collected using both qualitative and quantitative research methods, leading to the development and presentation of four different iterations of the PECC Framework during this period.

To improve the transparency and accessibility of the empirical foundation, Table 1 provides an overview of the main studies conducted between 2017 and 2023. Each of these studies contributed quantitative and qualitative data to the development of PECC. While detailed data analyses are presented in the respective publications, this article synthesises the findings across these multiple iterations to highlight the framework's evolution and impact. This approach aligns with common practice in design-based research, where cumulative evidence is built through iterative, context-sensitive investigations [82].

Table 1. Overview of empirical studies informing different development phases of PECC Framework [80,81,83–91].

Project	Teachers/Students	Methods	Data Collected	References	Phase
NOLB 2015–2017	11/478	surveys, game analysis, observations	pre/post-surveys, AttrakDiff, project analysis (rating scala on game elements, learning goal achievement etc.)	[81,83–86]	1
Coding Weeks 2018–2019	0/61	surveys, interviews	project analysis (rating scala on game elements, learning goal achievement etc.), pre/post-surveys on interest, self-efficacy etc., cognitive measurements	[87,88]	2
RemoteMentor 2018–2019	16/61	focus group discussions, game analysis	tutee/tutor behavior, Computational Thinking scala	[89]	3
Coding4All 2020–2021	0/24	interviews, analytics	support tool usage (logdata)	[80]	3
Code'n' Stitch 2019–2021	4/229	pre/post-tests, observations	intrinsic motivation scale, gender-specific observations	[90]	4
Making at School	12/ca. 600	Interviews, focus groups	success factors for Making	[91]	4

In the following section, we present the development of PECC. We draw on the work of diSessa and Cobb (2004), namely using the description of their development of ontological innovations as a structure with which to present the development of PECC through four phases [26]. Within each phase, the specific research objectives, an overview of the studies, and the key findings informing the development of PECC are presented. Throughout the four phases, the overarching goals were to develop a theoretical framework for teachers and trainers who want to design CS activities according to gender-sensitive standards; to generate useful resources with which to implement the PECC Framework; and to effect measurable changes in girls' intrinsic motivation.

4. Development of PECC

4.1. Phase I: Identifying a Need and Initial Conceptualisation (May 2018)

Research Objectives

During the NOLB project, we were confronted with different challenges. This led to multiple studies using different methods and analysis approaches, including the identification of "surprising events" that resulted in further developments which needed to be trialed in the research (e.g., the research design or further development of the app). At the end of the project, the question of what could be evaluated remained open. Even within the Austrian pilot project, there were many variances between classes, for example, in how programming was introduced, in the duration of the intervention, in settings, or in the exploration. Both teachers and researchers involved in the project changed over time and a lot of data was poorly analyzed. It felt, therefore, almost impossible to compare the different cycles/classes. However, the different game projects created by students and the qualitative and quantitative questionnaires, mainly related to the second project cycle, ended up being very useful.

During the project, we were able to identify some teachers who planned the lessons with us in detail and provided feedback on all aspects. At this point, we designed more exploratory exercises to see what our students wanted to program as a game. At least, that is what we thought we were accomplishing. We asked students to design their own games using storyboards, with the expectation that they would complete the games in a limited number of units. However, most of the games could not be completed. Therefore, the idea of predefined templates to start with became more and more prevalent. The result was that the children were often very motivated to program games in the beginning, but equally often found the app they were using too difficult. The gender differences we wanted to observe were made more obvious because of this orientation, as it was mainly the girls who felt insecure with this kind of technology.

This context led to four hypotheses examined in three Austrian schools, which acted as pilot sites for the NOLB project:

We hypothesize that girls and boys

- use different game design elements in the games they create (as reported in [83])—H1.
- perceive the programming units differently (e.g., in terms of what they consider important, their focus during the activities, the conditions they view as supportive for success, and their learning goals; reported in [84])—H2.
- rate the Create@School app differently according to its perceived attractiveness (as reported in [85])—H3.
- have different prerequisites that influence their ability to achieve the intended learning goals, such as their preferred approach (e.g., use of templates) or the number of completed units (as reported in [86])—H4.

These aimed to determine the differences between boys and girls in computer science classes and used "strategic essentialism" to draw attention to the enduring efficacy of socially constructed categories in CS [92,93]. This concept refers to the strategic use of essential or simplistic identity categories to achieve specific political or social goals. This can be criticized because it often confirms "stereotypical" predictions [94]. In the context of STEM promotion, this could mean that the category of “girls” or “women” is used strategically to justify certain policies or programs that target the underrepresentation of this group in STEM subjects. However, it is important to emphasize that strategic essentialism is not a plea for a rigid or stereotypical view of gender. Rather, it recognizes that gender identities are diverse and complex and calls for this complexity to be taken into account, even though simplistic categories are often quickly used for strategic purposes. In our case, it was helpful to raise awareness of gender issues in CS and help to establish a more inclusive classroom. In addition, we conducted a literature review on female students in computer science education with a focus on understanding stereotypes, negative effects, and positive motivation [9].

Research Context

The research involved 478 students (281 girls, 197 boys) and 11 teachers in 22 classes over two and a half years. The project results were validated in three cycles—a feasibility study, followed by a first cycle (13 classes, n = 272, f = 165, m = 107) and a second cycle (9 classes, n = 206, f = 116, m = 90)—typically with 20–30 pupils per class. The lessons were conducted at three schools in Graz, Austria, with classes from grade 6 to grade 12 (10 to 17 years old). Most students participated in the Pocket Code lessons in Grade 9. The reason for this is that CS is only a compulsory subject for one hour per week in Grade 9. Thus, we also introduce the app in subjects like fine art, English, and physics.

All classes followed a similar sequence: First, we introduced the project, the research team, and the app itself, with example games and an explanation/demonstration of the user interface. Second, this was followed by two or four double lessons where students worked on their specific project (the main part of the lesson) and where students programmed their own games or made extensions to existing games (predefined templates). Lower-grade students (grades 6 and 7 and classes with fewer than four units) were given a scaffold or pre-coded template that was missing certain parts of the code (indicated by note-bricks), e.g., collision detection or the use of tilt sensors. In other cases, students started by drawing a storyboard to get a clear picture of their game and to identify the game elements, characters, and programming concepts needed (as described in [83]). Finally, students showed their games to their classmates (in plenary), followed by a feedback session. The team visited all units to take notes on observed problems, gender differences, mistakes, or difficulties with the app itself. Different pre- and post-surveys, observations, documentation, video, and image material were collected.

Teachers who introduced this gamified approach during the feasibility study and the first cycle used the app Pocket Code. Teachers who had lessons during the second cycle used the new app version for school purposes, with the name Create@School. Teachers could decide whether students should work in small groups on a common program, where each student creates a game level that is combined into one big game at the end; whether they should work in groups of two on the same program (either on one or two tablets); or whether they should work individually.

Key Findings

To answer H1, the individually created projects (f = 37, m = 34) were analyzed in terms of the game elements (e.g., game genre, theme, visual/sound design, the main characters used, and other programming patterns). Girls’ games were found to have significantly different design patterns compared to boys’ programs. Adventure genres were preferred; more narrative, emotional, and imaginative elements were used; they were more likely to use their own graphics (i.e., own artwork) and programming; on average, more objects were added.

H2 was answered using quantitative and qualitative surveys with 131 students (f = 68, m = 63) to explore their experiences with the Create@School app and the lessons in which the app used [83]. There were clear divisions between male and female students' experiences, with girls giving both positive and negative feedback on the design of the learning activity itself, while boys focused on programming and the app itself. Surprisingly, answers from girls also included statements that they did not like coding at all, whereas this was not an answer given by boys.

An AttakDiff test (https://www.attrakdiff.de, accessed on 18 June 2025) was used to answer H3, involving 350 students from Austria and Spain (198 boys and 152 girls, [84]). The framework evaluates (1) usability and usefulness as perceived by the users, (2) the satisfaction of the users with the app, and (3) the attractiveness of the app. The app received very similar ratings from students of both genders. Female participants rated the app as more technical than human, more complicated than simple, and more confusing than clearly structured. Male students generally found the app more appealing than female students. They also rated the app as more likable, inviting, appealing, and motivating than girls.

To test H4, all students' projects, including those based on predefined templates, were analyzed according to the achievement of learning objectives (f = 240, m = 160, [86]). Learning objectives were defined together with the individual teachers and communicated to the class. Girls in lower grades were more likely to achieve the learning objectives. Female students achieved the learning objective more easily with pre-coded templates or in cross-curricular courses with specific learning objectives. During the field observations, the team found that girls were more cautious and uncertain about using the app. For example, they asked questions such as "Can I click this button?" or "What should I do next?". In higher grades, observations showed that girls needed more motivation from teachers and were less proud of having completed their program at the end.

During our research, several unexpected findings emerged that had a significant impact on our understanding of group work. First, group activities exhibited primarily cooperative rather than collaborative features. For example, each group member designed his or her own level of a game, and these individual efforts were later integrated into a final group outcome. Students faced different challenges and questions at different stages, all with the common goal of completing their respective levels within the allotted time. We found that the distribution of work in the all-girl groups was generally less effective than for male participants, who often prioritized creative activities, such as drawing, over programming tasks. Most teachers recognized this imbalance and shifted from group-based tasks to individual work in the second cycle. In these lessons, although students worked independently, they all addressed the same problem, creating an environment conducive to communication and collaboration.

Towards the end of the NOLB project and informed by the results above and conversations with teachers, it became clear that there was a need to bring these various elements together in a more structured framework to support teachers and all those who work with games and students. Although this was not the original plan, we felt these findings fit well into a framework. The GMTF developed during NOLB used game-based methods to unify concepts, practices, and pedagogical approaches across all curricula. It was based on the principles of Universal Design for Learning (UDL, [95]) and focused on the "what," "how," and !"why" of learning. As a result, it was very tailored to games, took a broader view of organizational school components (thus only applicable in school settings), and the gender component was only marginally considered. However, it provided a good starting point for PECC. The first version was clearly aimed at teachers (who were also involved in its creation) and was complemented via a checklist (indicating situations where gender plays a role) and a lesson plan [81].

Initial Conceptualization of PECC

Drawing on the findings outlined above, we identified four key components of a gender- sensitive pedagogical framework—PLAYING, ENGAGEMENT, CREATVITIY, and CODING (PECC)—as illustrated in Figure 1. Essential elements within the 4 key components are shown in the circle segments.

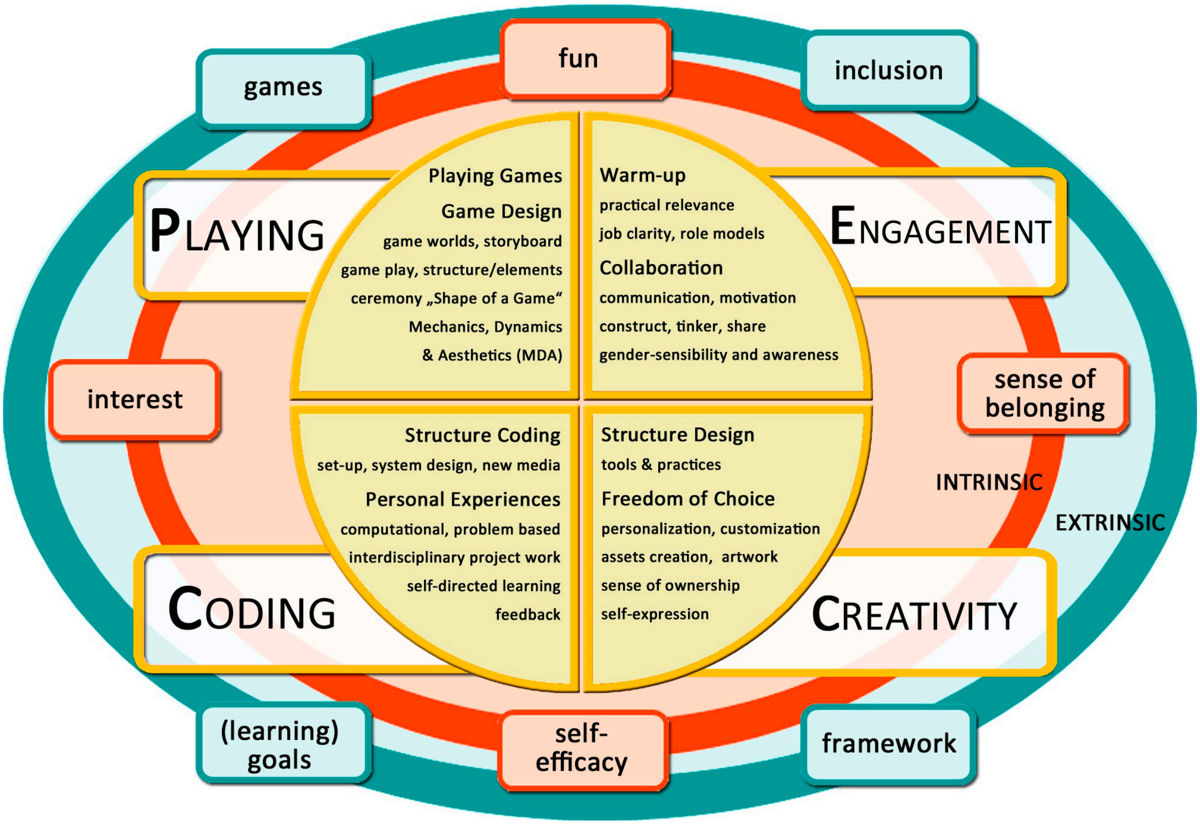

Figure 1. Initial conceptualization of PECC [81].

The results of H1 were incorporated into the PLAYING component. *Playing Games*: Before programming their own games, pupils should first have the opportunity to play existing games that also match their interests. This helps them get a sense of good and bad games/game design. *Game Design*: Various game elements should be explained beforehand, e.g., in the form of a storyboard. This should also serve as support for less experienced players.

The outcomes of H2 formed the ENGAGEMENT and CREATVITIY parts of PECC. Regarding ENGAGEMENT, the emphasis was mainly on a suitable introduction to CS and programming, presented within the element *warm-up*. Here, it is vital to highlight professional perspectives and careers in CS and also to show role models (see Section 2.2, Gender and Pedagogy). The next focus was on the element of *cooperation* and not on "group work". Here, the results of H3 were very influential and showed that the girls were more likely to achieve the learning goal when they worked on a common problem and were able to support each other but were less likely to achieve it when they worked in teams where the programming tasks were distributed to team members—see unexpected events.

A large number of open-ended responses (see H2) reflected a desire for creativity in elements, e.g., drawing, photographing, and personalizing characters, and then finally creating personalized games. Therefore, the following possible opportunities were included to support *freedom of choice*: personalization, customization, asset creation, artwork, sense of ownership, and self-expression.

The last study, H4, mainly influenced the CODING part of PECC. Here, the consideration of different methods, working practices, and prior knowledge of the learners is essential. *Structure Coding*: Depending on the context, an introduction with a predefined template or smaller mini-games may be appropriate to provide an initial scaffolded introduction to coding or programming concepts. *Personal Experience*: Preferably, programming should be designed in a problem-oriented and interdisciplinary way. Self-directed learning and situational feedback from instructors/teachers are helpful.

In addition to the findings from the NOLB project, an extensive review of the literature highlighted the importance of the four intrinsic and extrinsic motivational factors that should be considered in order to engage all genders in such gamified program activities [9]. These are incorporated in the green and red bands, which encircle the rest of the action (see Figure 1). While the representation and the extrinsic motivators of PECC have changed in further progressions, the four intrinsic characteristics have remained unchanged: *interest*, *self-efficacy*, *sense of belonging*, *and fun*. These values closely align with constructionist pedagogic values [9]. To determine whether these intrinsic parameters improve with PECC interventions, a questionnaire was developed. Different research influenced this questionnaire (e.g., [96,97]). It was used during the most recent studies, but could be validated for the first time with a larger sample during the Code'n'Stitch project (e.g., distribution, reliability analysis)—see Section "5. Final Version of PECC".

4.2. Phase II: Evaluation and Development of PECC (2017—2019)

Research Objectives

The initial PECC Framework was implemented and evaluated in several school and out-of-school contexts over two years (2017–2019), along with practical resources for teachers. There were several additional aims of these studies, including developing a better understanding of what girls in particular need in order to support their engagement; understanding the experiences of learners and teachers; and developing existing tools and creating additional tools to support teachers and students.

Research Context

In 2017, we conducted two exploratory case studies: one with a class of girls only (n = 10) and one with a mixed-sex class (boys = 5, girls = 7). Besides the questionnaire, in-depth interviews were conducted with four girls (two from each group) to derive insights from the working process and programming experiences. The lesson structure itself was discussed with the teacher beforehand and the curriculum was linked to the elements of PECC. Field notes were also taken during and immediately after class. Resources were adopted to support teachers and students in their game design: this included the PECC Game Assessment Template [88], a storyboard for game development, a checklist for teachers [81], and an example lesson plan—see Figure 2. In addition, a new version of the app called "Luna&Cat" has been released. This app was developed based on the PECC with the goal of supporting female users in the development of games. This is achieved through increased awareness of the content used (new featured games by girls for girls), a themed UI, new assets suggested by girls, and the development of a community of like-minded (female) programmers. Overall, the structure of the lessons was maintained over the different courses, but more time was allocated for the first three phases of the framework.

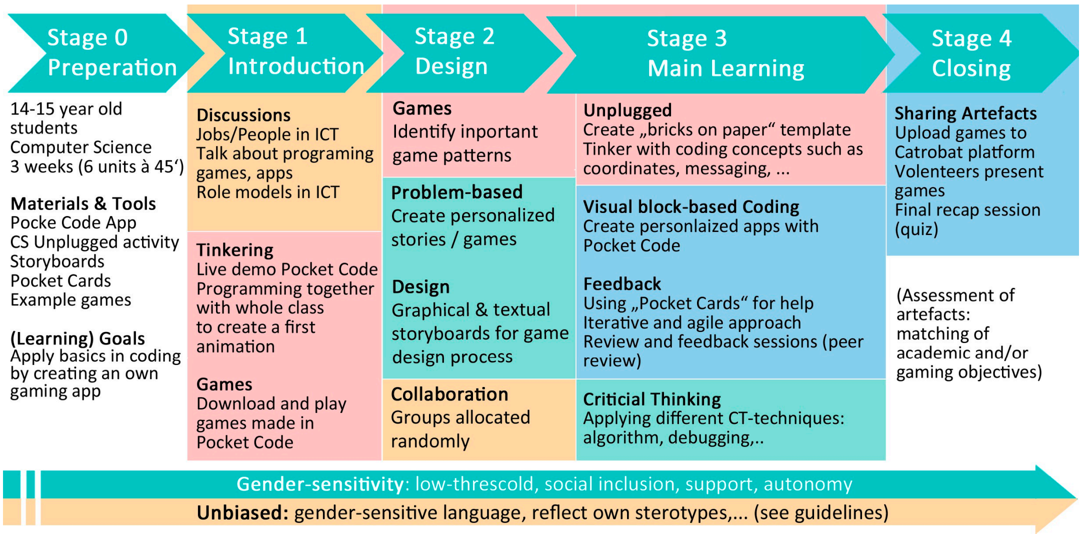

Figure 2. Example lesson plan for PECC. Colors match PECC Framework in Figure 3.

These resources were implemented in a second series of studies that focused on the PECC Framework in action outside school. In 2018, the first Girls Coding Week (GCW) was introduced to children as an out-of-school activity for five days during the summer [87]. Overall, 13 girls aged between 11 to 14 participated in the GCW (average age: 12.8). In 2019, the GCW was introduced as a four-day event with 28 girls, with a parallel mixed-gender group (20 in total, with 13 boys and 7 girls). Across the two groups, the average age was 12.9. Again, the questionnaire was used and, additionally, the revised PECC Game Assessment Template was used to evaluate the games created by the students [88].

Key Findings

The first two exploratory case studies demonstrated how the PECC Framework and its individual elements could be applied in a school setting, supported by the following lesson plan structure: preparation, introduction, design, main learning, and closure (see Figure 2). Together, these steps were found to be effective in achieving the predefined learning goals. The interviews yielded valuable insights into girls' views of both themselves and working with boys, while a qualitative content analysis, following Mayring (2014), was used to map a holistic perspective of the lesson and explore the framework in experimental form [98]. Descriptive analysis of the notes also revealed details of gender and group issues and how these influenced PECC activities and interventions of the teacher [48].

Putting the PECC Framework into practice yielded several insights and led to adaptations to it. Essentially, 3/4 of the framework is concerned with introducing different concepts (e.g., unplugged, game design, playing games), creating a common understanding (discussion, warm-up exercises), and other preparations or methods (e.g., teamwork, asset design), and only the last item is concerned with programming. In this first series of studies, however, the focus was mainly on programming or programming concepts and less on engagement. Thus, very quickly, stereotypical situations became visible in both the researcher's observations and from the interviews, which are also often observed in the literature on CS education (e.g., [99]). The atmosphere in the class, especially in the mixed-gender group, was often perceived as being stressful and noisy. Girls would often choose to wait for someone to help them (teacher, coach) or ask the teacher directly, while boys would work together, with one often guiding the group. Girls also worked very precisely and spent more time designing their games, whereas boys acted in a more task-driven way, with the net result being that most of the boys finalized the game, which was not the case for girls.

Gender-sensitive teaching tries to counteract this gender bias in CS classes and challenges them (see Section 2.2 Gender and Pedagogy). For example, if female students rate their performance lower than their male peers and are therefore hesitant to participate in class actively, a gender hierarchy is confirmed in which boys tend to be positioned as knowledgeable and girls tend to be positioned as unknowledgeable. The fact that girls prefer a safer environment without pressure and competition, and with more time to complete their tasks, could be misinterpreted by the teacher to mean that these girls are less capable, although this simply shows a different way of working or approaching work [11].

The findings of the second series of studies during the summer courses showed that there was a focus on conceptual and design elements and less on programming components [79]. The games and elements of gaming that female teenagers tended to like, create, and play with mostly followed stereotypical expectations (e.g., animals or girls as main characters, adventure genre, nature theme etc.). In contrast to our experiences in heterogeneous course settings, girls felt prouder when presenting their games in front of their peers/parents. The PECC Assessment Template was applicable and the students' projects showed considerable variance in design, game elements, and complexity [88]. Third, we evaluated the learning value of PECC game design activities. Students used a graphical storyboard template which helped them to stick to game design elements (with four areas for the title, instruction, game, and end screen) and a textual storyboard template to support students in making decisions about their game elements (to define the title, gameplay, game mechanics, dynamics, and athletics (MDA), etc.). Working memory, arithmetic skills, and creativity were assessed using well-known psychological measures (e.g., for working memory [100]; or creativity [101]). The results show that different concepts of CT applied in games are associated with separate cognitive measures [79]. The number of design elements correlated with both working memory and arithmetic skills. The number of game elements correlated with creativity, whereas the complexity of the project was only predicted by age.

Reflective Discussions

In early 2019, initial dialogues evolved with the second author of this paper. Given her experience with frameworks that enable collaborative work, a series of exciting reflective discussions about the PECC Framework started, resulting in many new versions of the framework. Discussions began by exploring the shared experiences of the authors in different educational research settings in relation to the first version PECC, including formal and non-formal secondary education, monolingual and bilingual learning environments, and single-sex and mixed-sex education settings, with students from low to high socio-economic backgrounds using a variety of digital tools and approaches (not just games) to support their learning. Fundamental questions were asked about decisions made in relation to pedagogic theory, popular models of introducing coding to young people, and recent research evidence, enabling us to explore tensions between the first phase of research, the initial PECC Framework, and the results of the second phase of research. For example, although not made explicit in the presentation of the PECC Framework, the constructionist idea of providing students with an opportunity to create a personally meaningful artefact informed and continued to inform our work in subsequent iterations, as well as decisions about the priority and/or relevance of elements in the framework. The importance of sharing games with others throughout the process of construction in order to gain feedback was also raised. Another example was the role of intrinsic and extrinsic motivators, how they related to specific areas of the framework, and how the framework itself could be designed to facilitate these. Finally, we explored the importance of having a framework which could be quickly understood by busy teachers with supporting materials, not only in terms of the resources already generated but future developments. As a primary result, the framework was simplified in general.

Changes to PECC

The empirical evidence and reflective discussions had several impacts on the PECC Framework:

PLAYING: This category has been more broadly defined and equated with playful learning. Now, elements of playing are considered essential to the overall instructional experience and are included in different phases of the lesson plan. For example, different programming constructs were taught step-by-step and supported with CS Unplugged activities, guided tasks, and constructive challenges (phases: instruction, main learning). Together, 10 modular coding units were designed to provide all the basic knowledge needed to program one's own game [87]. In this way, children with different prior experiences can discover those before starting to think about and program their individual games. Also, for an introduction, playing games oneself was included as an essential component to give a sense of good games/good game design to those who may not play games on their own. In addition, the focus on game design was shifted to the key component of CREATVITIY: It became equally important to introduce game elements such as genres, themes, game mechanics/dynamics/aesthetics (MDAs, [57]), or other frameworks to introduce games to non-gamers. The various templates (storyboard, assessment) were proven to be supportive tools for this phase and were introduced in the last of the 10 units.

ENGAGEMENT: The focus was primarily on creating "safe" learning environments, initially conceptualized as girls-only activities. To ensure a suitable starting point for everyone, discussions in small groups were introduced at the beginning. Here, not only should one's own experiences or favorite games/apps be discussed, but so should the work people carry out in CS. As a final result, each group designed and presented a poster and a drawing of a person working in computer science.

The key component coding was also adapted further. For example, a feedback round was introduced during the creation phase of one's own games. This also had the purpose of giving new ideas to children who may have been less motivated up to that point. The sharing of ideas was also emphasized by presenting the games at the end of the week. Parents and other people were invited and the children received certificates. This was perceived as a very positive event (and was also an additional motivation for many). In the end, the concept of coding seemed somewhat misplaced and more like a list of missing keywords. In addition, the focus should not be on programming, but on creation overall. Therefore, this component was renamed creation, with an emphasis on artefacts being created and shared.

Some of these changes were rapidly implemented due to a new project, "Code'n'Stitch", with school courses starting in the Fall of 2019 focused on programming digital designs and patterns that can be realized with a programmable embroidery machine [102]. Thus, creating games was not the focus of this activity.

Based on regular reflective discussions and the studies described above, a new version of the PECC Framework was published [48]—see Figure 3. The main differences were the focus on fewer and more relevant concepts, new components such as CS Unplugged activities, the creation of "safe" learning environments, the introduction of different game elements, and the renaming of the main category "CODING" to "CREATING".

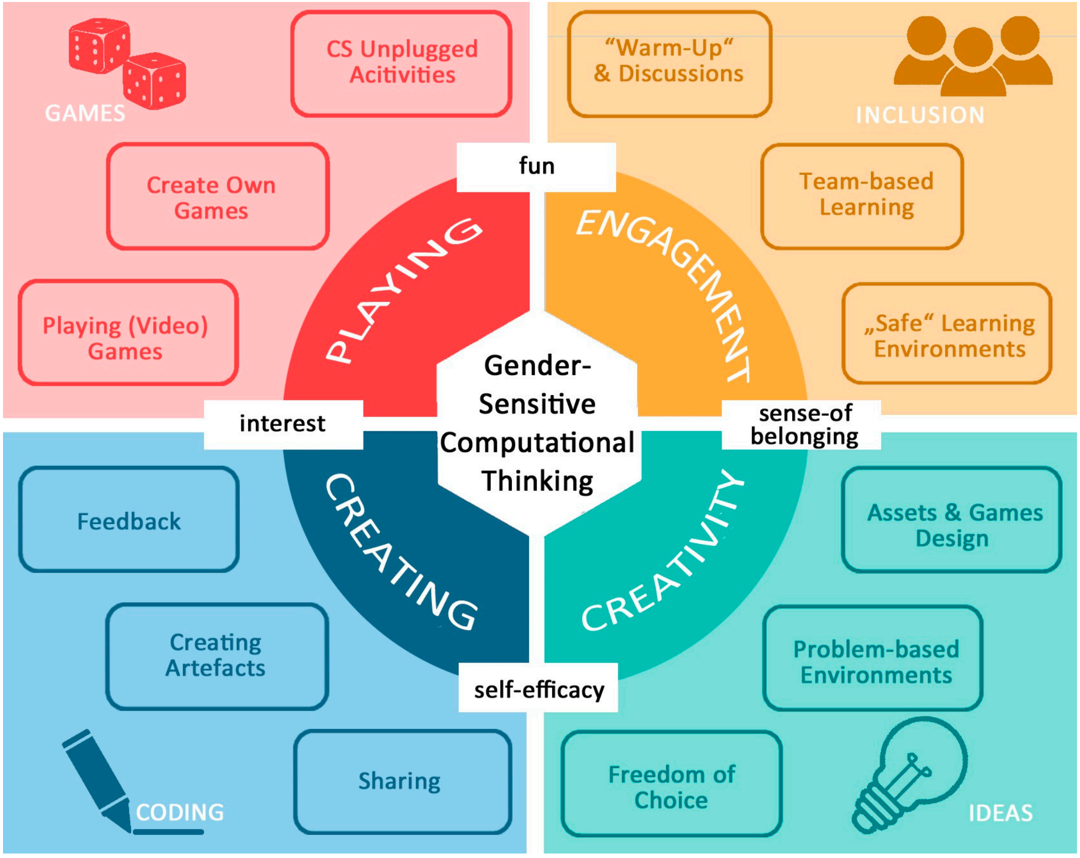

Figure 3. Result of second phase of PECC (2018–2019) [48].

4.3. Phase III of PECC (2020–2021)

Research Objectives

In 2018, the "RemoteMentor" project started. This used PECC as the underpinning framework design and involved tutors in regular school classes. In this use of the PECC Framework, we encountered several obstacles that complicated the process. First, the brevity of the sessions provided little time for discussion or playful introduction. At the same time, students had the opportunity to interact with tutors, who were themselves CS students and role models, through direct personal attention. These experiences proved invaluable for future adaptations of the PECC Framework, particularly during the COVID-19 pandemic, and for the implementation of Girls Coding Weeks as online courses. As PECC evolved for online adaption, a wealth of new resources was created, primarily to provide more online support and guidance to remote tutors.

Research Context

During the RemoteMentor project, 61 female students aged 14 to 15 received one-to-one mentoring (16 tutors) through smartphones for their coding project during their regular CS and arts lessons [89]. The aim of the one-year investigation was to analyze gendered aspects during the online tutoring process and the output of the collaborative coding project. Data were collected through focus group discussions (8 groups with 47 girls), evaluation of the online tutoring units (123 remote coding sessions of 30–40 min), and analysis of the final games (n = 61) in regard to the applied CT concepts. In addition, tutorials were developed for tutors, including rules and instructions; additional apps were used for online sessions (e.g., Zoom, Discord); and new features of the Pocket Code App were deployed (integrating scenes and the grouping of objects).

Although these coding units followed the PECC Framework, the results initially did not directly inform the framework. However, they became more interesting in March/April 2020, in the context of the COVID-19 pandemic, where students had to be home-schooled and many activities could only be carried out online. Due to the pandemic, a coding week that was originally planned to be on-site had to be offered and digitized as an online course, i.e., as an out-of-school activity, in Hildesheim, Germany. However, the design of the course "Coding4All" followed the idea of the "GirlsCodingWeek" [87]. For the course, small videos and audio tutorials were created, GIFs were animated, and CS Unplugged activities were provided. Each of the 10 chapters of GCW was divided into the following units: Introduction to various control constructs (e.g., loops, selections, etc.), including a CS Unplugged activity explaining the concept, a step-by-step task, a final challenge, and a quiz. Comments could be added to each chapter and individual solutions were submitted. The children received feedback from the tutors. Each day, the children worked on 3–4 units of the course, and on the last day they developed their own game. The entire course was freely available.

The first online "Coding4All" course was offered online in April 2020 (4 days) [80]. The workshop was actively accompanied by two one-hour online sessions daily in small groups of a maximum of 10 participants and one trainer. These collaborative meetings are essential to ensure the best possible support is provided to the participants, as well as teamwork. On the one hand, the participants work on the individual units of the course alone. On the other hand, if students needed help, they had several options: First, they could fill out an online form and choose between three options: (a) book a session with a specific trainer via Zoom, (b) receive feedback on their uploaded programs, or (c) ask a question and receive a written answer. Participants could also join the helpline without prior registration. Another option, available on the second day, was to leave comments on the lectures online (visible to all participants) and ask questions via the comments section. On the first day, we started with 24 participants between 10 and 15 years of age (average age 12.16; 10 girls, 14 boys). Quantitative data was collected during the course (e.g., durations of meetings, questions asked, uploaded programs). After the course, 10 telephone interviews were conducted with course participants (6 girls, 4 boys) to obtain deeper insight into their online experiences. We were particularly interested in how girls made use of various online support services (compared to boys).

Key Findings

The research question of the RemoteMentor project was "Does the tutoring style of the tutors and the sex of the tutors have an impact on the experience and success of programming activities for girls?" The sociological analysis suggested that the nature of tutoring, not the tutors' gender, influenced the girls' programming experiences [89]. Girls were more involved if the tutoring style matched their attitude; girls with active or insecure passive attitudes benefited from more supportive tutoring. In terms of CT (as part of the PECC Game Assessment Template), girls with an attitude that is more active applied more concepts, while girls with an attitude that is more passive incorporated more advanced concepts, which were possibly introduced by tutors. Stereotypes were noted, with some girls critical of the project's intent. The brief, task-based tutor–student relationship limited potential role model impacts. Despite initial preferences for female tutors, girls grew confident with online tutoring regardless of the tutor's gender, although they wanted to keep the same tutor for further sessions.

For the "Coding4All" project, the first author examined gendered differences in the use of online support services [80]. Results showed that girls preferred to ask for one-to-one meetings, chat messages, or short voice messages to communicate with the tutors. Boys were less likely to book meetings but used the full range of communication tools available to them. Girls had longer (almost 3 min) and more frequent conversations with trainers (76.79% of the sessions were booked by girls), and 73.21% preferred to talk to their already known trainer, showing similar results to the previous project.

Changes to PECC

The results of both projects showed that PECC was not only suitable for use in traditional classrooms but in online courses; however, it required some adjustments in this regard—see Figure 4. In particular, the extrinsic motivators (outside of the framework) needed refinement. On the one hand, the main component PLAYING was deposited when finding a suitable entry point into CS. This was due to the fact that often little time was available for a playful introduction (especially in regular school lessons). Nevertheless, it is necessary to introduce the individual to often abstract programming concepts through playful experimentation, playing/creating games by oneself, or CS Unplugged activities. Online, where it is often even more difficult to check whether what has been learned has actually been understood, it is important to give time for trial and error and for tinkering with tools. As seen in the literature and based on our reflection, there is more than one way to achieve this. Therefore, the term *tinkering* was added as an important part of the PLAYING component. For ENGAGEMENT, inclusion remained as the key extrinsic motivator (see the framework in Figure 1). Also, online platforms facilitated engaging discussions, collaborative teamwork, and the establishment of *"safe" learning environments*, particularly during group activities. Each day, a new topic from the realm of CS, such as binary numbers or job opportunities in this field, was brought to the forefront of discussion and everyone was encouraged to participate. Within the scope of the RemoteMentor project, teamwork was primarily fostered between the tutor and tutee. Based on the findings from the two projects, *support* was chosen as the new extrinsic motivator for the CREATING phase, reflecting the needs of girls in both projects to receive help when needed. As *coding* still plays a big part of all activities, this was added again as an activity as well as suitable *debugging* strategies. Here, it is important that teachers/tutors do not give a direct answer to questions, but that these are worked out together with the tutee. As a concept, debugging is introduced centrally within the agile approach through short, iterative phases based on the Scratch cycle (Imagine, Create, Experiment, Share, Reflect [102]).

In reflective discussions between the co-authors, *freedom of choice* within the component CREATVITIY began to be viewed as problematic. While the idea of freedom in terms of students being able to come up with their own projects and pursue their own interests was a core constructionist idea within the previous version of the framework, it was challenging to put the concept into practice. Reflecting on her observations of recent and past implementations, the first author noted the challenge faced by some groups to come up with an initial idea, which resulted in delays and ultimately meant some students would not complete their projects. In 2000, Schwartz coined the term "tyranny of freedom" arguing that excessive freedom can be experienced as a kind of tyranny, a concept used by the second author when considering the educational implications of a lack of structure or expectations for students creating in digital environments [103]. However, the sense of freedom to be able to pursue one's own ideas remained important. Thus, in the new version of the framework, the terms *freedom* and *own ideas* were exchanged at a conceptual level, meaning that at a practical level, the focus was now on students developing their own ideas but most likely within parameters decided by a teacher, rather than complete freedom of choice.

A version of the PECC Framework was published in 2022 in an edited German volume entitled *Competence Models for the Digital Transformation: Orientation and Practical Examples*. The main revisions concerned the extrinsic parameters, with a focus on suitable entry points through tinkering activities, inclusion, freedom and suitable support (feedback, debugging, etc.) during creating.

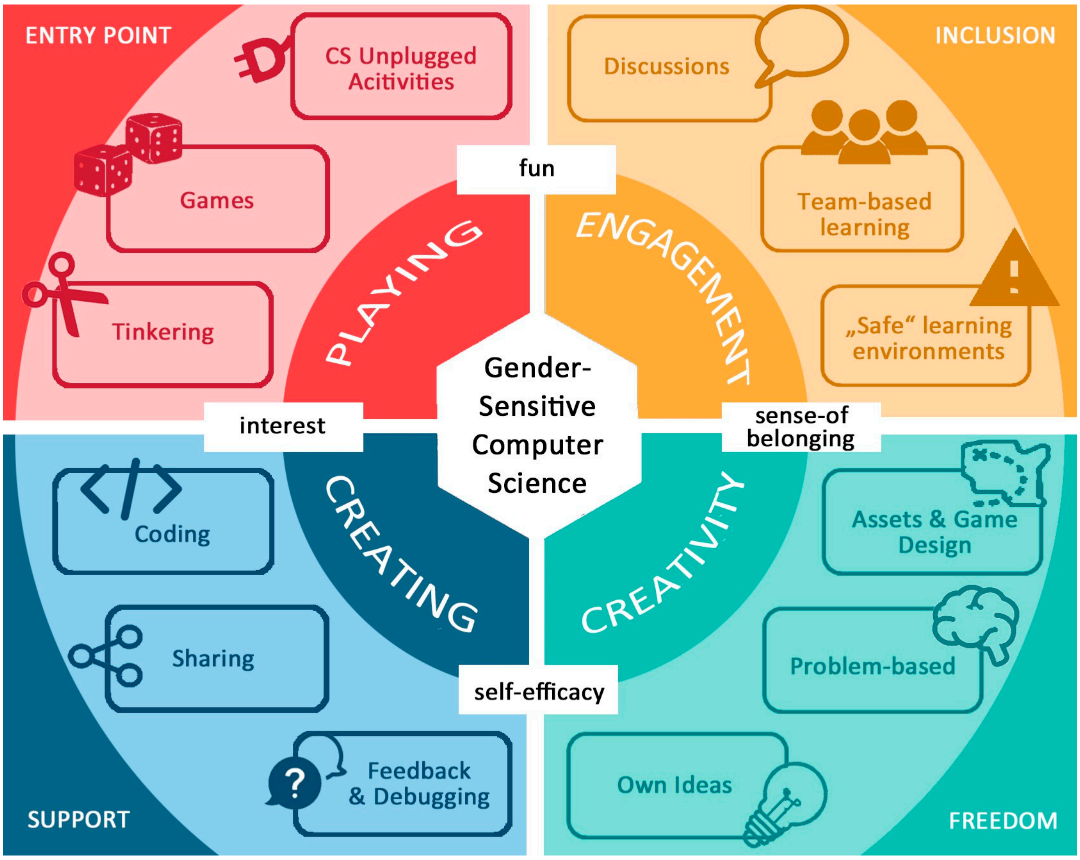

Figure 4. Result of third phase of PECC (2020–2021).

5. Final Version of PECC (2020–2023)

Research Objectives

The final version of our framework was redesigned during the project "Making at School" (2021–2023; [91]). As shown in the literature review, Making combines many aspects that are already part of the PECC Framework. Therefore, it made sense to use, extend, and evaluate PECC in Making classes as well.

Research Context

As part of the two-year national project "Making at School", the Zurich University of Teacher Education (PHZH) developed different scenarios for Making for 5th- to 9th-grade pupils (aged 10 to 15 years old). This time the focus was very much on the teachers' point of view. In the project, 12 teachers were involved in about 60 classes (ca. 600 pupils). Different methods, such as interviews and focus group discussions, were conducted with the teachers involved, and lesson plans and final statements were collected. Special attention was paid to various details involved in planning the lesson, such as focusing on specific competences, learning content, or the structure of the unit. Many new resources have been developed to support teachers, e.g., planning tools to support students in developing their own ideas, new video tutorials, and several online courses. These are openly available.

In addition, this project made use of the results of the Code'n'Stitch project conducted in 2018–2020 at TU Graz. The project accompanied three Austrian schools, including 229 pupils (123 girls, 106 boys) and 4 teachers. The evaluation of questionnaire (pre/post-testing the intrinsic motivators of PECC) and the programmed/stitched designs should constitute the structure of PECC. Therefore, a special focus was directed toward a gender-sensitive design of the school workshops. For example, at the beginning of the project, a course on gender equality and diversity was held, featuring all project members and trainers who were going to run the courses in the schools. The material was prepared attractively and aesthetically, dialogues on role models were initiated (e.g., in the form of course cards), a focus was placed on problem-solving (e.g., in the form of abstracting the desired patterns), great emphasis was placed on the design and creative process (two of the lessons were dedicated to their designs), and our own version of the app with the name Embroidery Designer was developed further. Besides the new possibilities that programming offers for handicraft lessons, a huge variety of tutorials for different patterns (easy to medium) were made available, for example, stars, flowers, or mine creeper figure. The second cycle was conducted online (during COVID-19) and aimed to verify the tutorials developed and adaptation of the app in the school context based on the student's experiences. After the second cycle, the following resources were created: First, a Wiki page was created to serve as a knowledge repository of all the materials. Second, we developed an Instagram account to post new tutorials, photos, and videos. Third, we made YouTube videos with small beginner tutorials and linked them on the Wiki pages.

Key Findings

As of 2023, three of the Making courses offered have already been tested: the basics of Making, digital pattern creation (using Embroidery Designer), and laser cutting 1 × 1. Results from focus group discussions, interviews, and the written summaries of teachers could be described as follows: Despite its promising possibilities, the implementation of the Making approach in schools faces considerable challenges as it represents a new kind of student-centered learning. In addition to the need to acquire and teach the necessary technical skills, organizational hurdles must be overcome and pedagogical concepts developed to meet the specific requirements of Making. Here, we see the need for more demand-oriented consulting concepts to equip educational institutions with tools and strategies that additionally aim to increase the acceptance of technology. Here, the TPACK model [104] also points out that teachers need technological knowledge (operating tools) and pedagogical knowledge (accompanying learning processes) in addition to content knowledge (finding suitable problems) for the implementation and accompaniment of Making activities. Therefore, in the future, teachers must be trained primarily in their role as multipliers for the didactically meaningful use of playful Making activities in combination with programming.

The course "digital pattern creation" is based on the results of the aforementioned Code'n'Stitch project. The results from the first cycle (late 2019) showed exciting results from the questionnaire and the review of the final programmed/printed patterns, which encouraged us to use "digital patterns", which is particularly appropriate for gender-sensitive CS. The study showed once again that there were gender differences in students' perceptions of intrinsic motivation in interest and self-efficacy when it came to programming. Although boys showed significantly more interest in the activity at the beginning, this decreased after the activity. For girls, on the other hand, interest was high at the beginning and remained at a similar level after the intervention. It is possible that gender stereotypes about programming also influenced this initial interest. For example, boys may have shown more interest at T1 (pre-questionnaire) due to societal expectations or prejudices. These may have been less influential at T2, resulting in a smaller difference in interest (nevertheless, all scores remained between agreement and strong agreement). Furthermore, all showed an overall decrease in self-efficacy related to programming after the intervention and there were significant differences in the sense of belonging and enjoyment of fun and games, with boys showing higher scores in both aspects. Finally, girls showed a higher level of agreement with both post-questions, indicating a greater sense of pride in their designs, which was a significant finding. In addition, girls were more interested in realizing their own designs as final products (creating their own designed patterns), while boys are more inclined to use existing tutorials for different shapes or designs for their products. These findings may indicate that the intervention, although generally considered valuable, may not have been fully responsive to the needs and interests of all participants. Due to the lack of time, not all elements of PECC could be included and this time the main focus was on the creation of designs and programming. Nevertheless, this shows that it is essential to empower girls through their programming and achieve better overall outcomes. Incorporating elements identified in the framework as positive—such as discussions, teamwork and using different entry points—could provide a more comprehensive and effective approach to improving girls' experiences in STEM education. The scale results of the questionnaire suggest that the questions for the item "self-efficacy" may need to be adjusted.

In 2023, new findings from the literature, a survey of representatives of various STEM promotion initiatives in Switzerland, focus group discussions with the target group of girls, and applicable and gender-sensitive recommendations for STEM activities were presented and prepared in the form of a checklist. In cooperation with the network "Planet MINT" (an association of Swiss organizations promoting STEM activities), a "Code of Conduct" is being developed for STEM initiatives in Switzerland (taking into account diversity, integration and gender sensitivity, but also the requirement to include pedagogical expertise, etc.) and the guidelines will be broken down to achievable goals. These are designed as a set of cards for STEM organizations and show concrete examples of implementation.

Changes to PECC

Based on these findings and reflection over six years of research and development, a final refined version of PECC has been developed—see Figure 5. Based on the two projects, the following adjustments were made: In the PLAYING phase, the term *Making* was added to the term *tinkering*. In general, introducing Making fosters active learning, enhances problem-solving skills, encourages creativity, boosts self-efficacy, prepares students for future careers, instills lifelong learning, and promotes inclusive education (see Section 2.4 on constructionism and PECC). In the ENGAGEMENT phase, a specific *student-centered* work method was added. Especially in the Making projects, it can be observed that teachers increasingly take on the role of coach and the work is oriented towards students' individual projects. In the CREATVITIY phase, the focus was less on games and more on *design and storytelling* (which is also an important aspect of Making). Here, new and innovative products are imbued with life and innovative ideas are brought to fruition. Emphasis was also placed on students' own projects and the term "freedom" (extrinsic parameter) was replaced with *autonomy*, which is more a common term in everyday school life. In the phase of CREATION, "coding" was replaced by *programming*. Especially in German-speaking countries, the term coding is less common and can also be confused with "coding/encoding and data structures". In addition, *showing* was added to *sharing;* especially in Making (and also in constructionism theory), it is important to present the finished artefacts and to bring their advantages to the attention of the audience.

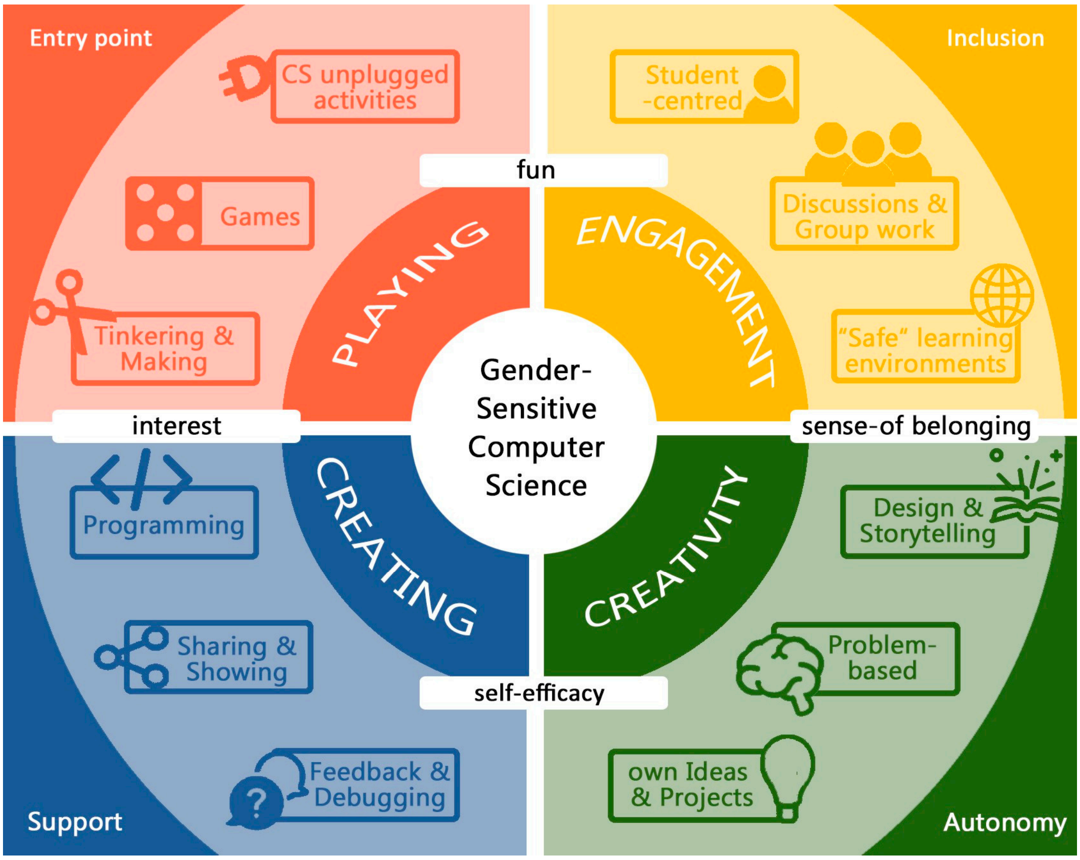

Figure 5. Final version of PECC, 2023. Puplished in the German volume entitled *Competence Models for the Digital Transformation*, 2022.

The next section presents the final version of the PECC in detail, outlining all its components.

5.1. PLAYING: Ensuring a Suitable Entry Point to Programming for All

As an entry point to CS, playing can make the learning process more engaging, more enjoyable, and less intimidating, especially for beginners. This approach promotes creativity, reduces the fear of making mistakes, and encourages exploration and experimentation—aspects that can motivate and retain the interest of girls, in particular in programming, by making CS more tangible and fun and less stressful.

CS Unplugged activities help pupils learn programming by allowing them to understand and explore basic CS concepts away from the computer. These activities use games, puzzles, and other interactive methods to demonstrate principles such as algorithms, binary numbers, data compression, and CT. By presenting these concepts in a tangible, non-intimidating way, children can grasp abstract ideas more easily and develop a solid foundation that can make learning programming languages much easier later on. This approach also allows for flexibility in tailoring activities to a variety of interests, encouraging more girls to participate and recognize the importance of CS to their own lives.

Creating one's own *games* helps students to get started with programming by providing an engaging, hands-on context for understanding and applying programming concepts. Girls can also relate to the narratives and characters in games, which can stimulate their interest in game design and programming. Instructions and templates give students a starting point and support their learning, while creating a game helps them develop their own knowledge structures. The process involves problem-solving, logic, design, and programming skills, all of which are part of programming. With our app "Pocket Code", children can create their own games using a simple visual block-based programming language. *Playing games* helps students by revealing important game structures and rules and by giving them the opportunity to redesign existing games. This enhances their understanding of game design and logic, which are critical elements in CS and programming. It also encourages creativity, strategic thinking, and problem-solving skills, making learning more enjoyable and interesting. By giving students the opportunity to learn by playing a variety of games, individual interest can be recognized.

Tinkering and Making activities can serve as an entry point into programming and CS, especially for girls. By relating to the real world and encouraging creativity, these activities resonate with girls, who often seek context when learning. The hands-on nature of these tasks demystifies technology and makes it less intimidating. Through trial and error, students develop critical problem-solving skills and resilience-related aspects of programming. The collaborative elements of these activities can foster a sense of community and create a supportive and inclusive learning environment.

5.2. ENGAGEMENT: Cultivating Inclusive Learning Environments

ENGAGEMENT is critical in CS education because it promotes deeper learning, persistence, and success. An *inclusive*, engaged environment fosters a *sense of belonging*, reduces gender bias, and empowers students, especially girls, to actively participate, promoting more diverse and balanced representation in the field.

A *student-centered* approach can significantly improve student engagement and inclusion, especially for girls, in several ways. First, it focuses on students' interests and experiences, which can make learning more meaningful and engaging. Giving students, especially girls, the opportunity to work on projects that are relevant to their lives can increase their motivation to actively participate. Second, it emphasizes collaboration, which can foster a sense of community and belonging and is particularly appealing to girls, who often value social interactions in their learning. Finally, a student-centered approach promotes autonomy and problem-solving and helps all students build confidence in their abilities, which is critical for inclusion and success in areas such as programming.

Discussions and group work can help students in CS in many ways and even provide a foundation for future IT careers. Collaborative processes can allow a deeper understanding of complex concepts to be obtained. When students discuss and solve problems together, they benefit from multiple perspectives, improving their problem-solving skills and understanding. Furthermore, discussions provide a platform for students to articulate their thoughts and ideas, which builds confidence in their knowledge. Initial discussions can break down preconceptions to show a variety of roles and paths in the IT industry and help students envision their own possible career paths and strengths. Also, real role models can be invited (e.g., students, trainees) and showcases can be presented. Additionally, the importance of CS skills for all jobs can be emphasized more.

"Safe" learning environments play a critical role in CS education, especially in engaging girls. A sense of belonging can make students feel more comfortable and confident in participating and asking questions, which is critical for learning and mastering the material. However, some students feel uncomfortable asking questions because they are afraid of being judged or uninformed. This discomfort can affect their learning and engagement. To alleviate this, teachers can create a supportive, nonjudgmental environment where all questions are welcome and valued and where mistakes are seen as opportunities to learn. A competitive environment can cause anxiety. While some competitions can motivate students, an overemphasis on winning can discourage learners, especially girls, who often prefer cooperative learning environments to competition.

5.3. CREATIVITY: Sparking Autonomy in Learners

Creativity in CS encourages innovative problem-solving. It promotes autonomy and enables all students to take responsibility for their learning process. Furthermore, creativity increases self-efficacy, boosts confidence in student abilities, and thus motivates and empowers learners, especially girls, in this male-dominated field.

Design and storytelling can greatly facilitate the learning of programming by stimulating creativity and promoting self-efficacy. Integrating design into programming tasks allows students to express their creativity and individuality. Personalization, customization, and asset creation give students a sense of ownership over their work. These elements provide opportunities for self-expression, which can increase engagement and motivation—key factors for effective learning. This hands-on approach is especially well received by girls. Storytelling is another powerful tool. By incorporating narrative elements into programming tasks, students can better visualize and understand abstract concepts. It also taps into the power of empathy and emotional engagement, which helps students, especially girls, see the relevance and impact of their work, which can increase their interest and motivation.

Problem-based learning activities can significantly improve programming skills and foster creativity. By exploring different content and encouraging creative thinking, these activities stimulate students' curiosity and problem-solving skills. Problem-based activities present information not as isolated facts but as components of a larger system working toward a common goal. For example, programming your own game requires an understanding of various elements such as game logic, control structures, and user interfaces. When devising solutions to problems, students must think outside the box, experiment with different strategies, and learn through trial and error. Problem-solving tasks in Making can align with the 17 Sustainable Development Goals (SDGs), thereby demonstrating the potential to address real-world issues.

Implementing their *own ideas & projects* can significantly improve the programming skills and creativity of students. Through this process, students are not just passive recipients of knowledge but active creators who apply what they learn in meaningful ways. Implementing their own ideas fosters a creative process that allows students to bring their individuality to their projects. This approach not only makes learning more interesting and enjoyable but also gives students a sense of ownership and pride in their work. The Making approach fits perfectly with this aspect and can be especially beneficial for girls. It allows them to express their personalities and interests and provides a platform for self-expression and creativity. By overcoming challenges in their projects, girls can realize their potential and resilience, which promotes their continued interest and success in programming.

5.4. CREATING: Catering Support for Individual Learning Styles

Various forms of support such as tutorials, videos, feedback, and 1:1 tutoring are essential when creating programs, especially for girls. These resources not only provide guidance and encouragement but also boost students' interest, confidence, and autonomy in programming, fostering their engagement in the field.

Programming, especially when using visual programming languages such as Scratch or Pocket Code, can greatly enhance the creative development of students by providing opportunities for personal experiences, self-directed learning, and problem-solving. Programming, by its very nature, requires creativity and innovative thinking. As students write code or edit visual blocks on a platform, they experiment with different solutions, refine their ideas, and learn to overcome challenges. Programming, and visual programming languages in particular, offer students the flexibility to learn at their own pace, choose their own projects, and direct their own learning path. This sense of autonomy can increase motivation and engagement, deepen understanding, and build confidence in their own abilities.

Sharing and showing projects can greatly enhance the creative process by building confidence, motivation, and a sense of community. This is consistent with the principles of constructionism and Maker Education theory, which emphasize the value of public construction and the presentation of the finished product. This process allows learners to externalize their understanding, reflect on their ideas, and refine their work based on feedback. For girls, who often seek validation and social connection when learning, this can be especially encouraging. With products created as a Making activity, students can showcase their learning journey and the skills they have developed. This can enhance their learning experience, inspire them to take on new challenges, and give them a sense of belonging. In addition, sharing finished artifacts with an online community creates a sense of global connectedness, which can make programming more engaging and meaningful.

Feedback and debugging are integral parts of the learning process in programming and can be particularly powerful for students. Peer feedback provides students with valuable information to improve their work and can boost their confidence. Some students, especially girls, need more affirmation, particularly when working with new technologies. Positive, constructive feedback can raise their expectations of success and increase the extent to which they value the task. It reassures them that they are making progress, validates their efforts, and encourages them to keep learning. Debugging can be an important learning experience because it promotes problem-solving, resilience, and a deeper understanding of the programming language. By viewing mistakes not as failures but as normal and essential parts of the learning process, students can develop a healthy, growth-oriented attitude toward challenges. This can be especially beneficial for girls who are more likely to doubt their abilities or fear failure.

6. Conclusions

The PECC (PLAYING, ENGAGEMENT, CREATVITIY, CREATING) Framework was rigorously developed and iteratively refined over four phases and six years, informed by empirical data from classroom studies, student feedback, and teacher input across diverse educational settings. The application of the framework was successful in promoting the intended concepts and activities of PLAYING, ENGAGEMENT, CREATVITIY, and CREATING. Across multiple interventions, the framework proved to be a practical and adaptable guide for teachers, particularly effective in supporting girls' participation, while maintaining equitable learning outcomes for all genders.

The success of the framework was demonstrated in several studies, with similar results observed for girls and boys. The framework proved effective in solving problems from the previous phase and ensured the adoption of a comprehensive and balanced approach to learning. The final version of the framework included all of the important components but was designed in a way that did not overwhelm teachers. The practicality of the framework was underscored by its use in our case studies as well as in larger contexts such as online courses. The framework has found its place in constructionism, Making, and coding/game-oriented learning environments and has been actively referred to and used in schools and courses in Austria, Germany, and Switzerland.

Safe learning environments, opportunities for self-expression, and a hands-on approach to learning were among the key aspects of the PECC Framework most valued by teachers. However, the most important point is that the PECC Framework is not only theoretical but also provides practical resources in a tangible and research-informed way in order to implement gender-sensitive pedagogy. This can be seen in the practical outcomes and resources that have been created and used, paving the way for a new movement in CS education.

While the framework addresses many aspects of teaching and learning in CS, future iterations could further explore aspects that were not as prominent in this final version. We recognize that the framework only addresses gender and ignores intersectional issues such as race, socioeconomic status, social and science capital, and so on. Indeed, as Shaw et al. (2023) highlight in the context of North American education, "by ignoring the intersectional experiences of Black women and girls navigating the computing ecosystem, researchers risk making incorrect generalizations that can lead to ineffective approaches for increasing Black women and girl's representation and retention in computing" (p.55) [86]. In the same vein, the framework has been developed through research in the German-speaking countries. As previously mentioned, language plays a significant role when the names of jobs are gendered but it is also worth considering that there may be cultural influences which mean that the operationalization of the framework may be different in other countries. However, the engagement and retention of women in CS education and careers is currently a significant challenge which does not appear to be going away any time soon. There are many issues that need to be addressed, from workplace culture to societal views. Within the context of secondary education, PECC provides a theoretically informed and research-evidenced model, with practical resources and real-world impact, that represents one promising route to addressing the challenge.

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

Funding: With institutional support of University of Hildesheim and Graz University of Technology. This research was funded by H2020 [645215], NetIdee [2335], FEMtech [14975187/866755], and DIZH-Innovationprogram [Making at School].

Data Availability Statement: The data from the Code'n'Stitch study are openly available at [https://zenodo.org/records/7635652, accessed on 18 June 2025] and [https://zenodo.org/records/7642304, accessed on 18 June 2025]. Data from the other studies (NOLB, RemoteMentor, Coding4All, Girls Coding Week, and Making at School) are not publicly available due to privacy or ethical restrictions.

Acknowledgments: This work was supported by the H2020 project "No One Left Behind", Grant Agreement Number 645215, NetIdee "RemoteMentor", Funding Year 2017/Project Call #12/ProjectID: 2335, FEMtech Research projects 2017, "Code'n'Stitch", eCall/FFGNr. 14975187/866755, and DIZH-Innovationprogram Project Call, "Making at School" (https://dizh.ch/2022/01/03/making-im-unterricht/, accessed on 18 June 2025).

Conflicts of Interest: We have no conflicts of interest to disclose.

References

1. European Statista. Female Students Underrepresented in ICT. Available online: https://ec.europa.eu/eurostat/en/web/products-eurostat-news/-/edn-20190425-1 (accessed on 21 April 2025).
2. European Statista. Share of Women Working in STEM Fields in 2021, by Country. Available online: https://www.statista.com/statistics/1116527/share-women-stem-country/ (accessed on 21 April 2025).
3. GOV.uk. National Curriculum in England: Computing Programmes of Study. Available online: https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study (accessed on 21 April 2025).
4. German Informatics Society (GIS). Informatikunterricht in Deutschland. Available online: https://informatik-monitor.de/2024-25 (accessed on 21 April 2025).
5. D-EDK. Lehrplan 21: Modul Medien und Informatik; Zürich. Available online: https://zh.lehrplan.ch/index.php?code=b%7C10%7C0&la=yes (accessed on 21 April 2025).

6. Welsh Government. *Digital Competence Framework*; Welsh Government: Cardiff, UK. Available online: https://hwb.gov.wales/curriculum-for-wales/cross-curricular-skills-frameworks/digital-competence-framework/ (accessed on 21 April 2025).
7. Federal Ministry of Education, Science and Research. Digitale Grundbildung. Available online: https://www.bmb.gv.at/Themen/schule/zrp/dibi/dgb.html (accessed on 21 April 2025).
8. Ryoo, J.J. Pedagogy That Supports Computer Science for All. *ACM Trans. Comput. Educ.* **2019**, *19*, 1–23. [CrossRef]
9. Spieler, B.; Oates-Indruchová, L.; Slany, W. Female Students in Computer Science Education: Understanding Stereotypes, Negative Impacts, and Positive Motivation. *J. Women Minor. Sci. Eng.* **2020**, *26*, 473–510. [CrossRef]
10. Gabay-Egozi, L.; Shavit, Y.; Yaish, M. Gender Differences in Fields of Study: The Role of Significant Others and Rational Choice Motivations. *Eur. Sociol. Rev.* **2015**, *31*, 284–297. [CrossRef]
11. Happe, L.; Buhnova, B.; Koziolek, A.; Wagner, I. Effective Measures to Foster Girls' Interest in Secondary Computer Science Education. *Educ. Inf. Technol.* **2021**, *26*, 2811–2829. [CrossRef]
12. Bers, M.U.; Flannery, L.; Kazakoff, E.R.; Sullivan, A. Computational Thinking and Tinkering: Exploration of an Early Childhood Robotics Curriculum. *Comput. Educ.* **2014**, *72*, 145–157. [CrossRef]
13. Beyer, S. Women in Computer Science: Deterrents. In *Encyclopedia of Computer Science and Technology*; CRC Press: Boca Raton, FL, USA, 2017; pp. 871–879.
14. Dasgupta, N.; Stout, J.G. Girls and Women in Science, Technology, Engineering, and Mathematics: Steming the Tide and Broadening Participation in STEM Careers. *Policy Insights Behav. Brain Sci.* **2014**, *1*, 21–29. [CrossRef]
15. Lakanen, A.-J.; Kärkkäinen, T. Identifying Pathways to Computer Science: The Long-Term Impact of Short-Term Game Programming Outreach Interventions. *ACM Trans. Comput. Educ.* **2019**, *19*, 1–30. [CrossRef]
16. Baumeister, R.F.; Leary, M.R. The need to belong: Desire for interpersonal attachments as a fundamental human motivation. *Psychol. Bull.* **1995**, *117*, 497–529. [CrossRef]
17. Master, A.; Cheryan, S.; Meltzoff, A.N. Computing Whether She Belongs: Stereotypes Undermine Girls' Interest and Sense of Belonging in Computer Science. *J. Educ. Psychol.* **2016**, *108*, 424. [CrossRef]
18. Lindner, J.; Makarova, E.; Bernhard, D.; Brovelli, D. Toward Gender Equality in Education—Teachers' Beliefs about Gender and Math. *Educ. Sci.* **2022**, *12*, 373. [CrossRef]
19. Galdi, S.; Cadinu, M.; Tomasetto, C. The Roots of Stereotype Threat: When Automatic Associations Disrupt Girls' Math Performance. *Child Dev.* **2014**, *85*, 250–263. [CrossRef] [PubMed]
20. Young, D.M.; Rudman, L.A.; Buettner, H.M.; McLean, M.C. The Influence of Female Role Models on Women's Implicit Science Cognitions. *Psychol. Women Q.* **2013**, *37*, 283–292. [CrossRef]
21. Vrieler, T.; Nylén, A.; Cajander, Å. Computer Science Club for Girls and Boys—A Survey Study on Gender Differences. *Comput. Sci. Educ.* **2021**, *31*, 431–461. [CrossRef]
22. American Association of University Women. The Stem Gap: Women and Girls in Science, Technology, Engineering and Mathematics. 2018. Available online: https://www.aauw.org/resources/research/the-stem-gap/ (accessed on 21 April 2025).
23. Bredenfeld, A.; Leimbach, T. The roberta initiative. In Proceedings of the Simulation, Modeling, and Programming for Autonomous Robots, Darmstadt, Germany, 15–18 November 2010; pp. 558–567. [CrossRef]
24. Anderson, T.; Shattuck, J. Design-based research: A decade of progress in education research? *Educ. Res.* **2012**, *41*, 16–25. [CrossRef]
25. Cobb, P.; Confrey, J.; DiSessa, A.; Lehrer, R.; Schauble, L. Design experiments in educational research. *Educ. Res.* **2003**, *32*, 9–13. [CrossRef]
26. diSessa, A.A.; Cobb, P. Ontological innovation and the role of theory in design experiments. *J. Learn. Sci.* **2004**, *13*, 77–103. [CrossRef]
27. Varma, R. Why so few women enroll in computing? gender and ethnic differences in students' perception. *Comput. Sci. Educ.* **2010**, *20*, 301–316. [CrossRef]
28. Heyder, A.; Steinmayr, R.; Kessels, U. Do teachers' beliefs about math aptitude and brilliance explain gender differences in children's math ability self-concept? *Front. Educ.* **2019**, *4*, 34. [CrossRef]
29. Kerr, B.; Robinson Kurpius, S.E. Encouraging talented girls in math and science: Effects of a guidance intervention. *High Abil. Stud.* **2004**, *15*, 85–102. [CrossRef]
30. Cheryan, S. Understanding the paradox in math-related fields: Why do some gender gaps remain while others do not? *Sex Roles* **2012**, *66*, 184–190. [CrossRef]
31. Hur, J.W.; Andrzejewski, C.E.; Marghitu, D. Girls and computer science: Experiences, perceptions, and career aspirations. *Comput. Sci. Educ.* **2017**, *27*, 100–120. [CrossRef]
32. Lawlor, G.; Byrne, P.; Tangney, B. "codeplus"—Measuring short-term efficacy in a non-formal, all-female cs outreach programme. *ACM Trans. Copmuting Educ.* **2020**, *20*, 1–18. [CrossRef]
33. Stoet, G.; Geary, D.C. The gender-equality paradox in science, technology, engineering, and mathematics education. *Psychol. Sci.* **2018**, *29*, 581–593. [CrossRef] [PubMed]

34. Connell, R.W. *Masculinities*; University of California Press: Berkeley, CA, USA, 1995.
35. Fennema, E.; Peterson, P.L.; Carpenter, T.P.; Lubinski, C.A. Teachers' attributions and beliefs about girls, boys, and mathematics. *Educ. Stud. Math.* **1990**, *21*, 55–69. [CrossRef]
36. Tiedemann, J. Parents' gender stereotypes and teachers' beliefs as predictors of children's concept of their mathematical ability in elementary school. *J. Educ. Psychol.* **2000**, *92*, 144. [CrossRef]
37. Tiedemann, J. Teachers' gender stereotypes as determinants of teacher perceptions in elementary school mathematics. *Educ. Stud. Math.* **2002**, *50*, 49–62. [CrossRef]
38. Gentrup, S.; Rjosk, C. Pygmalion and the gender gap: Do teacher expectations contribute to differences in achievement between boys and girls at the beginning of schooling? *Educ. Res. Eval.* **2018**, *24*, 295–323. [CrossRef]
39. Seegers, G.; Boekaerts, M. Gender-Related Differences in Self-Referenced Cognitions in Relation to Mathematics. *J. Res. Math. Educ.* **1996**, *27*, 215–240. [CrossRef]
40. Keller, C. Effect of Teachers' Stereotyping on Students' Stereotyping of Mathematics as a Male Domain. *J. Soc. Psychol.* **2001**, *141*, 165–173. [CrossRef]
41. Galdi, S.; Maass, A.; Cadinu, M. Objectifying Media: Their Effect on Gender Role Norms and Sexual Harassment of Women. *Psychol. Women Q.* **2014**, *38*, 398–413. [CrossRef]
42. Makarova, E. *Gendersensible Berufsorientierung und Berufswahl: Beiträge aus Forschung und Praxis*; Hep Verlag ag: Bern, Switzerland, 2019.
43. Makarova, E.; Wenger, N. *GESBI-Forschungsbericht: Gendergerechtigkeit in Lehrmitteln für Naturwissenschaftlichen Unterricht. Handreichung für Lehrpersonen*; Universität Basel, Institut für Bildungswissenschaften: Muttenz, Switzerland, 2019.
44. Medel, P.; Pournaghshband, V. Eliminating Gender Bias in Computer Science Education Materials. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, Seattle, WA, USA, 8–11 March 2017; pp. 411–416. [CrossRef]
45. Formanowicz, M.M.; Cislak, A.; Horvath, L.K.; Sczesny, S. Capturing Socially Motivated Linguistic Change: How the Use of Gender-Fair Language Affects Support for Social Initiatives in Austria and Poland. *Front. Psychol.* **2015**, *6*, 1617. [CrossRef] [PubMed]
46. Wetschanow, K. *Geschlecht Lernen. Gendersensible Didaktik und Pädagogik*; Buchmayr, M., Ed.; Studienverlag: Innsbruck, Austria, 2008; pp. 195–213.
47. Spieler, B.; Girvan, C. Das PECC-Framework: Gender-Sensibilität und Spielerische Programmierung in der Informatischen Grundbildung. In Proceedings of the DELFI 2020—Die 18 Fachtagung Bildungstechnologien der Gesellschaft für Informatik e.V., Online, 14–18 September 2020; Gesellschaft für Informatik e.V.: Bonn, Germany, 2020; pp. 247–258, ISBN 978-3-88579-702-9, ISSN 1617–5468.
48. Witt, A.-K.; Cuesta, M. How Gender Conscious Pedagogy in Higher Education Can Stimulate Actions of Social Justice in Society. *Soc. Incl.* **2014**, *2*, 12–23. [CrossRef]
49. Wang, M.-T.; Eccles, J.S.; Kenny, S. Not Lack of Ability but More Choice: Individual and Gender Differences in Choice of Careers in Science, Technology, Engineering, and Mathematics. *Psychol. Sci.* **2013**, *24*, 770–775. [CrossRef]
50. Sentance, S.; Csizmadia, A. Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Educ. Inf. Technol.* **2017**, *22*, 469–495. [CrossRef]
51. Alvarado, C.; Cao, Y.; Minnes, M. Gender Differences in Students' Behaviors in CS Classes Throughout the CS Major. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, Seattle, WA, USA, 8–11 March 2017; pp. 27–32. [CrossRef]
52. Külling, C.; Waller, G.; Suter, L.; Willemse, I.; Bernath, J.; Skirgaila, P.; Streule, P.; Süss, D. *JAMES-Studie 2022: Jugend, Aktivitäten, Medien—Erhebung Schweiz. Ergebnisbericht zur JAMES-Studie 2022*; ZHAW Zürcher Hochschule für Angewandte Wissenschaften: Zürich, Switzerland, 2022; Available online: https://www.zhaw.ch/storage/psychologie/upload/forschung/medienpsychologie/james/2018/Bericht_JAMES_2022_de.pdf (accessed on 21 April 2025).
53. Sharma, T.G.; Hamari, J.; Kesharwani, A.; Tak, P. Understanding Continuance Intention to Play Online Games: Roles of Self-Expressiveness, Self-Congruity, Self-Efficacy, and Perceived Risk. *Behav. Inf. Technol.* **2022**, *41*, 348–364. [CrossRef]
54. Schell, J. *Die Kunst des Game Designs: Bessere Games Konzipieren und Entwickeln*; BoD–Books on Demand: Norderstedt, Germany, 2020.
55. Brathwaite, B.; Schreiber, I. *Challenges for Game Designers*; Charles River Media Inc.: Rockland, MA, USA, 2008.
56. Tekinbas, K.S.; Zimmerman, E. *Rules of Play: Game Design Fundamentals*; MIT Press: Cambridge, MA, USA, 2003.
57. Hunicke, R.; LeBlanc, M.; Zubek, R. MDA: A Formal Approach to Game Design and Game Research. In Proceedings of the AAAI Workshop on Challenges in Game AI, San Jose, CA, USA, 25–29 July 2004; Volume 4, pp. 1–7.
58. Bassenge, L. *Making Room for Play: A Play-Centric Workshop Proposal for Increasing Girls' Game Literacy and Access to Games*; Uppsala Universitet: Uppsala, Sweden, 2021.
59. Games, A.; Kane, L. Exploring Adolescent's STEM Learning Through Scaffolded Game Design. In Proceedings of the 6th International Conference on Foundations of Digital Games, Bordeaux, France, 29 June–1 July 2011; pp. 1–8. [CrossRef]

60. Annis, B.; Nesbitt, R. *Results at the Top: Using Gender Intelligence to Create Breakthrough Growth*; John Wiley & Sons: Hoboken, NJ, USA, 2017.
61. Papert, S. *The Children's Machine: Rethinking School in the Age of the Computer*; BasicBooks: New York, NY, USA, 1993.
62. Papert, S.; Harel, I. Preface, Situating Constructionism. In *Constructionism: Research Reports and Essays*; Ablex Publishing Corporation: Norwood, NJ, USA, 1991.
63. Kafai, Y.; Vasudevan, V. Hi-Lo Tech Games: Crafting, Coding and Collaboration of Augmented Board Games by High School Youth. In Proceedings of the 14th International Conference on Interaction Design and Children, Medford, MA, USA, 21–25 June 2015; pp. 130–139. [CrossRef]
64. Harris, E.; Winterbottom, M.; Xanthoudaki, M.; Calcagnini, S.; de Pijer, I. *Tinkering: A Practitioner Guide for Developing and Implementing Tinkering Activities*; Contemporary Education for Innovators of Tomorrow Project; European Commission: Brussels, Belgium, 2016. Available online: https://www.researchgate.net/publication/306066132_A_PRACTITIONER_GUIDE_FOR_DEVELOPING_AND_IMPLEMENTING_TINKERING_ACTIVITIES (accessed on 21 April 2025).
65. Taub, R.; Ben-Ari, M.; Armoni, M. The Effect of CS Unplugged on Middle-School Students' Views of CS. *ACM Sigcse Bull.* **2009**, *41*, 99–103. [CrossRef]
66. Caena, F.; Redecker, C. Aligning Teacher Competence Frameworks to 21st Century Challenges: The Case for the European Digital Competence Framework for Educators (DigCompEdu). *Eur. J. Educ.* **2019**, *54*, 356–369. [CrossRef]
67. Schön, S.; Ebner, M.; Kumar, S. The Maker Movement. Implications of New Digital Gadgets, Fabrication Tools and Spaces for Creative Learning and Teaching. *eLearn. Pap.* **2014**, *39*, 14–25.
68. Schad, M.; Jones, W.M. The Maker Movement and Education: A Systematic Review of the Literature. *J. Res. Technol. Educ.* **2020**, *52*, 65–78. [CrossRef]
69. Becker, S.; Jacobsen, M. A Year at the Improv: The Evolution of Teacher and Student Identity in an Elementary School Makerspace. *Teach. Educ.* **2021**, *34*, 1–18. [CrossRef]
70. Hwang, J.-P.; Chang, M.-Y. Effects of Maker Education Integrating ARCS on Learners' Performance, Motivation, Self-Efficacy. *Int. J. Online Pedagog. Course Des.* **2022**, *12*, 1–9. [CrossRef]
71. Kafai, Y.B.; Peppler, K.A. Youth, Technology, and DIY: Developing Participatory Competencies in Creative Media Production. *Rev. Res. Educ.* **2011**, *35*, 89–119. [CrossRef]
72. Girvan, C. Constructionism, Creativity and Virtual Worlds. In *Constructionism and Creativity, Proceedings of the 3rd International Constructionism Conference 2014, Vienna, Austria, 19–23 August 2014*; Futschek, G., Kynigos, C., Eds.; Austrian Computer Society: Vienna, Austria, 2014; pp. 367–377.
73. Slany, W. Tinkering with Pocket Code, a Scratch-like programming app for your smartphone. In Proceedings of the Constructionism 2014, Vienna, Austria, 19–23 August 2014; Futschek, G., Kynigos, C., Eds.; Austrian Computer Society: Vienna, Austria, 2014.
74. Resnick, M.; Maloney, J.; Monroy-Hernández, A.; Rusk, N.; Eastmond, E.; Brennan, K.; Millner, A.; Rosenbaum, E.; Silver, J.; Silverman, B.; et al. Scratch: Programming for All. *Commun. ACM* **2009**, *52*, 60–67. [CrossRef]
75. Feurzeig, W. Toward a Culture of Creativity: A Personal Perspective on Logo's Early Years and Ongoing Potential. *Int. J. Comput. Math. Learn.* **2010**, *15*, 257–265. [CrossRef]
76. Hoyles, C.; Noss, R.; Adamson, R. Rethinking the Microworld Idea. *J. Educ. Comput. Res.* **2002**, *27*, 29–53. [CrossRef]
77. Barab, S.; Squire, K. Design-Based Research: Putting a Stake in the Ground. *J. Learn. Sci.* **2004**, *13*, 1–14. [CrossRef]
78. Brown, A. Design Experiments: Theoretical and Methodological Challenges in Creating Complex Interventions in Classroom Settings. *J. Learn. Sci.* **1992**, *2*, 141–178. [CrossRef]
79. Spieler, B.; Kemeny, F.; Landerl, K.; Binder, B.; Slany, W. The Learning Value of Game Design Activities: Association Between Computational Thinking and Cognitive Skills. In Proceedings of the 15th Workshop in Primary and Secondary Computing Education (WiPSCE), Virtual Event, 28–30 October 2020; Association for Computing Machinery: New York, NY, USA; pp. 1–4. [CrossRef]
80. Spieler, B. The Science Behind the Art of Engaging: Online Tutoring in Games and Coding. In Proceedings of the 15th European Conference on Games Based Learning (ECGBL), Online, 23–24 September 2021; Felicia, P., Ed.; Academic Conferences International: Reading, UK, 2021; pp. 691–981. Available online: https://www.researchgate.net/publication/357928561_The_Science_Behind_the_Art_of_Engaging_Support_in_Games_and_Coding (accessed on 21 April 2025).
81. Spieler, B. Development and Evaluation of Concepts and Tools to Reinforce Gender Equality by Engaging Female Teenagers in Coding. Ph.D. Thesis, Graz University of Technology, Styria, Austria, 2018. Available online: https://bernadette-spieler.com/wp-content/uploads/2018/10/Dissertation_Spieler.pdf (accessed on 21 April 2025).
82. Amiel, T.; Reeves, T.C. Design-Based Research and Educational Technology: Rethinking Technology and the Research Agenda. *Educ. Technol. Soc.* **2008**, *11*, 29–40.
83. Spieler, B.; Slany, W. Female Teenagers and Coding: Create Gender Sensitive and Creative Learning Environments. In Proceedings of the Constructionism 2018, Vilnius, Lithuania, 21–25 August 2018; Vilnius University: Vilnius, Lithuania, 2018; pp. 625–636.

84. Spieler, B.; Slany, W. Game Development-Based Learning Experience: Gender Differences in Game Design. In Proceedings of the 12th European Conference on Games Based Learning, Sophia Antipolis, France, 4–5 October 2018; pp. 616–625.
85. Spieler, B.; Slany, W. Luna&Cat: A Customised App to Attract Female Teenagers to Coding. In Proceedings of the Conference on Gender Research (ICGR 2019), Rome, Italy, 11–12 April 2019; pp. 583–591.
86. Spieler, B. Reinforcing Gender Equality by Analysing Female Teenagers' Performances in Coding Activities: A Lesson Learned. In Proceedings of the Conference on Gender IT 2018/GEWINN-Konferenz, Heilbronn, Germany, 24–35 May 2018; ACM: New York, NY, USA; pp. 1–8. [CrossRef]
87. Spieler, B.; Krnjic, V.; Slany, W. Girls Create Games: Lessons Learned. In Proceedings of the 13th European Conference on Games Based Learning, Odense, Denmark, 3–4 October 2019; pp. 675–684.
88. Spieler, B.; Kemeny, F. Design, Complexity, and Coding: A Framework to Evaluate Games. In Proceedings of the 14th International Conference on Game Based Learning (ECGBL), Brighton, UK, 24–25 September 2020; pp. 558–566.
89. Spieler, B.; Mikats, J.; Valentin, S.; Oates-Indruchová, L.; Slany, W. "RemoteMentor": Evaluation of Interactions Between Teenage Girls, Remote Tutors, and Coding Activities in School Lessons. In *Learning and Collaboration Technologies. Designing, Developing and Deploying Learning Experiences. HCII 2020. Lecture Notes in Computer Science*; Zaphiris, P., Ioannou, A., Eds.; Springer: Cham, Switzerland, 2020; Volume 12205, pp. 547–567.
90. Spieler, B.; Gursch, S.; Krnjic, V.; Horneck, K.; Slany, W. Designing, Coding and Embroidering: A Workflow for Gender-Sensitive and Interdisciplinary Teaching. *Int. J. Emerg. Technol. Learn.* **2024**, *19*, 79–104. [CrossRef]
91. Spieler, B.; Schifferle, T.; Dahinden, M. Making im Unterricht: Erfahrungen von Lehrpersonen aus dem Design-Based-Research-Projekt. Medienpädagogik. *Medien. Z. Für Theor. Und Prax. Medien.* **2023**, *56*, 331–363. Available online: https://www.researchgate.net/publication/378228149_Making_im_Unterricht_Erfahrungen_von_Lehrpersonen_aus_dem_Design-Based-Research-Projekt (accessed on 21 April 2025). [CrossRef]
92. Shaw, M.S.; Coleman, J.J.; Thomas, E.E.; Kafai, Y.B. Restorying a Black Girl's Future: Using Womanist Storytelling Methodologies to Reimagine Dominant Narratives in Computing Education. *J. Learn. Sci.* **2023**, *32*, 52–75. [CrossRef]
93. Spivak, G. The Post-Colonial Critique. In *Interviews, Strategies, Dialogues*; Routledge: London, UK; New York, NY, USA, 1990.
94. Grünberg, L. *From Gender Studies to Gender in Studies—Case Studies of Gender-Inclusive Curriculum in Higher Education*; UNESCO-CEPES (UNESCO European Centre for Higher Education): Bucharest, Romania, 2011.
95. Rose, D.H.; Meyer, A. *Teaching Every Student in the Digital Age: Universal Design for Learning*; Association for Supervision and Curriculum Development: Alexandria, VA, USA, 2002. [CrossRef]
96. Schwarzer, R.; Jerusalem, M. Generalized Self-Efficacy Scale. In *Measures in Health Psychology: A User's Portfolio. Causal and Control Beliefs*; Weinman, J., Wright, S., Johnston, M., Eds.; NFER-Nelson: Windsor, UK, 1995; pp. 35–37.
97. Krieger, S.; Allen, M.; Rawn, C. Are Females Disinclined to Tinker in Computer Science? In Proceedings of the 46th ACM Technical Symposium on Computer Science Education, Kansas City, MO, USA, 4–7 March 2015; pp. 102–107. [CrossRef]
98. Mayring, P. *Qualitative Content Analysis: Theoretical Foundation, Basic Procedures and Software Solution*; Philipp Mayring: Klagenfurt, Austria, 2014.
99. Denner, J.; Werner, L.; Bean, S.; Campe, S. The Girls Creating Games Program: Strategies for Engaging Middle School Girls in Information Technology. *Front. J. Women's Stud.* **2005**, *26*, 90–98. [CrossRef]
100. Süss, H.M.; Oberauer, K.; Wittmann, W.W.; Wilhelm, O.; Schulze, R. Working-Memory Capacity Explains Reasoning Ability—And a Little Bit More. *Intelligence* **2002**, *30*, 261–288. [CrossRef]
101. Runco, M.A. *Problem Finding, Problem Solving, and Creativity*; Greenwood Publishing Group; Ablex Publishing Corporation: New York, NY, USA, 1994.
102. Resnick, M. *Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play*; MIT Press: Cambridge, MA, USA, 2017. [CrossRef]
103. Schwartz, B. Self-Determination: The Tyranny of Freedom. *Am. Psychol.* **2000**, *55*, 79. [CrossRef]
104. Koehler, M.J.; Mishra, P.; Kereluik, K.; Shin, T.S.; Graham, C.R. The Technological Pedagogical Content Knowledge Framework. In *Handbook of Research on Educational Communications and Technology*, 4th ed.; Spector, J.M., Merrill, M.D., Elen, J., Eds.; Springer: New York, NY, USA, 2014; pp. 101–111. [CrossRef]