

Addressing the Gender Gap in Computer Programming Through the Design and Development of Serious Games

Carlos Vaz de Carvalho¹, *Senior Member, IEEE*, Špela Cerar, Jože Rugelj, Hariklia Tsalapatias, *Member, IEEE*, and Olivier Heidmann

Abstract—The gap between male and female participation in computer science education and careers is a worldwide issue that must be addressed by introducing early methodological learning interventions that make computer science attractive to all, that is by answering the following issues: a) misperception among educators, learners, parents and youngsters on the suitability of computer science careers to girls and b) a wrong assumption of an insufficient preparedness to do it successfully. This article presents a European initiative - CODING4GIRLS - that proposes to teach coding through a game design and development process based on a design thinking methodological approach that is linked to creativity and human-centered solutions. In this methodology, students address increasingly complex real-life challenges by designing and developing awareness raising serious games for which they need to learn specific coding concepts.

Index Terms—Programming, coding, serious games, game design based learning, gender gap, girl coding.

I. INTRODUCTION

THE gender gap in computer science courses and careers is a worldwide issue, as acknowledged by UNESCO [1]. According to the National Science Foundation in the US, currently women only represent 19% of the university graduates in computer science [2]. In Europe the situation is comparable as reported by the European Commission: “Of 1,000 women with a Bachelors or other first degree, only 29 hold a degree in Information and Communication Technologies (ICT) as compared to 95 men, and only 4 in 1000 women will eventually work in the ICT sector” [3]. Not only that, the number of female Computer Science graduates was steadily decreasing since 2000 [1].

Manuscript received August 5, 2019; revised January 13, 2020; accepted March 18, 2020. Date of publication July 9, 2020; date of current version August 5, 2020. (Portuguese version received April 2, 2020; revised June 8, 2020; accepted June 11, 2020). This work was supported by the European Union through the Erasmus+ programme under Project 2018-1-SI01-KA201-047013. (*Corresponding author: Carlos Vaz de Carvalho.*)

Carlos Vaz de Carvalho is with the GILT R&D, Instituto Superior de Engenharia do Porto, 4200-072 Porto, Portugal (e-mail: cmc@isep.ipp.pt).

Špela Cerar and Jože Rugelj are with the Faculty of Education, University of Ljubljana, 1000 Ljubljana, Slovenia (e-mail: spela.cerar@pef.uni-lj.si; joze.rugelj@pef.uni-lj.si).

Hariklia Tsalapatias and Olivier Heidmann are with the Department of Computer Engineering, University of Thessaly, 383 33 Volos, Greece (e-mail: htalapa@inf.uth.gr; olivier.heidmann@gmail.com).

There exists a Portuguese version of this article available at <http://rita.det.uvigo.es/VAEPRITA/V8N3/A15.pdf>

Digital Object Identifier 10.1109/RITA.2020.3008127

The disparity between male and female participation continues at the professional level: 80% of top-level employees in Silicon Valley are male and only 10% of the technical positions are occupied by female [4]. In Europe, the EC report indicates that only 30% of the around 7 million people working in the ICT sector are women and they are under-represented at all levels, especially in decision-making positions [2]. Even more disturbing is the fact that only 0.4% of 15-year-old girls in OECD countries want to be ICT professionals [5].

These statistics are in striking contrast with the increasing demand for computer science graduates that are necessary to fuel the exploding software industries that design and deliver solutions for a wide range of needs in today’s and tomorrow’s technology immersed society. The demand for skilled personnel in digital innovation-related sectors is expected to grow over the next years according to the New Skills for New Jobs Initiative [6].

The current lack of personnel with digital competences while there is an equally high youth unemployment, demonstrates a gap between the skills built through formal education and the ones needed by industry: “Due to differences in demands and skills – and despite soaring unemployment – there may be a lack of 900 000 skilled ICT workers in 2020.” [7]. This gap is not specific to a particular country; rather, it is evident at the global level calling for interventions to address it in order to ensure sustainable economic growth through the development of human capital. There is a need for girls as well as boys to develop skills to effectively engage in computer science careers and transition into the professional world.

The reasons for which computer science, and generally STEM (Science, Technology, Engineering, Mathematics) education, does not draw more girls seem to be related to a) a misperception among educators, learners, parents and youngsters on the suitability of computer science careers for girls [8]; and b) a wrong assumption of an insufficient preparedness to complete it successfully [9]–[11]. Therefore the gap between male and female participation in computer science education and careers must be addressed by introducing early methodological learning interventions, in the late years of basic education and on the early years of the secondary school, the age at which many girls lose interest in STEM and computer science [12]. These interventions should attract

girls by raising their awareness on the rich possibilities for professional and personal growth that computer science offers but also by offering adjusted methodologies to their interested and motivations.

In the past few years there have been several initiatives that tried to attract more students to the computer science field, like CodeWeek and Code.org's "Hour of Code", just to name the most well-known ones. Some of these initiatives targeted especially girls, e.g. Made With Code, Girls Who Code, Black Girls Code, Rails Girls, Ladies Learning Code, Girls Teaching Girls To Code [13], [14]. These initiatives were supplementary to the standard school curriculum, therefore trying to compensate "from the outside" the existing gap. Although relevant and with some success, these initiatives were still not able to turn the table on the gender gap issue in ICT and STEM.

This article presents the European initiative CODING4GIRLS that tries to overcome some of the identified limitations of the previous initiatives. In short, the activities in the scope of CODING4GIRLS are organized around the following stages:

1. The development of a theoretical learning framework for building programming skills among girls to ensure that outcomes meet the needs of the stakeholder secondary education community. The learning framework is based on design thinking and learning by designing and developing games;
2. The design and development of tools and resources to support the implementation of the framework including instructor support tools;
3. Proof-of-concept technical implementation of the proposed learning framework in an enlarged set of schools;
4. Improvement of the learning framework and corresponding tools and resources according to the obtained feedback;
5. Replication and wide multiplication of the framework implementation in a large number of schools.

II. LEARNING TO CODE USING SERIOUS GAMES

The existing imbalance between men and women in ICT careers has been a topic of discussion partly due to the fact that everyone should have equal access to the same chances and opportunities but also due to the acknowledgement that diversity in the workforce is connected with better economic performance [15]. Like mentioned before, women and girls participation in STEM and ICT courses and careers is still significantly low. Any approach to teaching computer science to girls needs to dispel negative socio-economic and cultural stereotypes in order to make this field appealing and "cool" to them [16]. Girls compare themselves and who they are to cultural stereotypes related to the STEM fields and decide that they don't fit in. There is a lack of female role models in the public eye to showcase that women are indeed fit for these fields [15].

According to a study conducted by Alserri *et al.* [10], who analyzed secondary schools data on the subject, girls who

were heavy gamers at 13-14 years old seemed to be more likely to pursue a STEM degree. Therefore he suggested that games could be a way to motivate girls into computer science and STEM degrees in general. For this strategy to succeed a clear understanding of the gaming preferences of girls and how they differ from boys is required. Men tend to prefer strategy, sports, action/adventure, and shooter games, while women tend to enjoy more action/adventure, puzzle, strategy and casual games [17]. Women tend to dislike direct competition and prefer problem resolution or collaboration [18]. There is also a preference for social games (with a rewarding system), collaborative and exploration games, and virtual life and party games. In regard to adventure games, there is a preference for observing first and playing after the act of observation [17].

A. Serious Games

Since their apparition on the consumer market about 50 years ago, computer games have slowly but steadily become a crucial element in our social and cultural environment [19], [20]. But the use of video games extended beyond their traditional entertainment scope as they are now used in many different contexts or purposes, like the medical field, for awareness raising of social issues, training purposes, language learning or even in the military sphere [21], [22]. One of the most common and ever increasing use of video games happens in education as they foster motivation and engagement, while at the same time strengthen psychomotor skills, enhance knowledge retention and decision-making skills, and provide opportunities for repeated practice and immediate feedback [23]. Educational games are serious games, defined as games "...that do not have entertainment, enjoyment or fun as their primary purpose" [24]. Another definition for serious games is that they are "a mental contest, played with a computer in accordance with specific rules, that uses entertainment to further government or corporate training, education, health, public policy, and strategic communication objectives" [25]. Serious games aim at using the mechanics, dynamics and components that make gaming appealing, and combine these with the desired information to be transmitted to the user, creating an interactive source for learning that motivates each user to extend their own knowledge and deepen their study in a challenging but fun approach [26]–[28]. As a study by Freitas and Jarvis [29] concluded, serious games are an advantageous tool in a teacher's toolbox in the sense that they allow for the creation of content for the purpose of adapted, specific and tailored learning without taking any of the fun out of it.

B. Examples of Serious Games for Learning How to Code

There is a wide variety of approaches for teaching programming to novice coders. Computer games can be used to teach coding, either by encouraging students to develop and create their own video games or by allowing them to play serious games whose goals encompass learning outcomes related to programming.

The idea behind using serious games to teach programming comes from the fact that, by being more engaging and motivational, they allow students to learn computational thinking and programming skills in entertaining and familiar environments, before transferring those skills to learning a programming language [30]. In the same article, Kazimoglu *et al.* proposed that serious games for learning programming and computational thinking should make a clear distinction between different programming constructs and encourage good programming practices [30].

This can be done, for example, through visual programming languages which are especially user friendly as they are both easy to operate (on a drag-and-drop basis) and present content and results in a very intuitive fashion and in a very immediate way. Scratch [31]–[34], Snap! [35], or Alice [36]–[38], among others, are famous and successful visual programming languages. These environments are useful for learning to code, as they avoid errors regarding syntax and logic. In a study that compared Scratch (visual based programming) and Pascal (command line or GUI based programming), it was found that learners were more motivated to continue their studies in computer sciences when using Scratch [32]. The same study also found that the creation of games and stories makes learners more creative and autonomous while learning programming and these environments allow students to create their own basic games.

Serious games can also be useful for motivating students for programming by creating interesting experiences: Shabanah *et al.* [39] created an Algorithm Game Designer aimed at improving engagement in algorithm visualization systems; Grivokostopoulou *et al.* [40] developed a game for learning AI algorithms, based on the Pacman game, so that through algorithm visualizations and animations students could be helped in their learning process, by demonstrating the way an algorithm operates, its functions and how to make proper decisions based on specific parameters.

Online game-based platforms dedicated to learn how to code have also appeared. Combéfis *et al.* [41] reviewed the main online game-platforms used for teaching programming skills and concluded that the following motivating factors were fundamental: 1) a feedback and assessment process; 2) aesthetics; 3) collaboration and multiplayer aspects; 4) guidance through the game; 5) no negative consequences; 6) music; 7) a medium-level of challenge to keep the interest of the players. The question of collaborative vs competitive games for programming was also highlighted by Miljanovic and Bradbury [42].

In the same article, Miljanovic *et al.* reviewed a large number of serious games for the development of programming skills and found out that they mainly focused on problem solving and fundamental programming concepts, with few games focusing on data structures, development methods and software design.

There are also other games that, while not directly targeting programming skills, develop relevant computational thinking skills, like abstraction (e.g. Tetris), decomposition (e.g. Sudoku), algorithmic thinking (e.g. Puzzle games),

evaluation (e.g. Memory Games), and generalization (e.g. Pattern Games) [43].

While adopting a constructionist educational approach it is important to not only focus on providing the game to students, but also providing them with the means and knowledge to develop their own games [44]. A study conducted by Carmichael [16] showcased the beneficial outcomes of combining computer science concepts with video games specifically to a young female target-group. When choosing the coding creation software the educators should consider the following series of requirements: the academic year of the students, the familiarity of the instructor(s) with coding projects, the capability of the student in creating a coding project, and the amount of time students will be spending learning the know-how to do it Alserri *et al.* [10] have developed a conceptual model for gender-based engagement through the use of games, consisting of five elements:

1) Learning elements: these are the elements that distinguish entertainment games from educational games;

2) Female preferences for digital games: these are the game mechanics that girls prefer, and have to be incorporated into the design in order to motivate and engage them, namely, exploration, character customization, storyline, social interaction, collaboration, challenges, fun, control and feedback;

3) Flow: the game elements that keep the player immersed and focused should also be incorporated in order to obtain engagement and motivation: challenges, fun, control, feedback, concentration, clear goals, skill and immersion;

4) Female game types and genres like fantasy and role-playing games;

5) Social gender factors: parental, peers and teacher influence.

Next we present in more detail some of the available platforms, games and programming environments that can be used for teaching programming skills.

1) *Scratch*: Scratch [31]–[34] can be used as an introduction to programming tool, especially among younger students, but also as a means to facilitate the transition to other programming environments. Scratch is a block-based programming environment – visual programming language with blocks that works as puzzle pieces, which allows for learners to assemble programs by snapping together the blocks and to receive visual feedback. One of the benefits of Scratch is that users worldwide can share their projects, communicate with other developers, help others to improve projects or remix shared projects. It also has an offline editor that allows for the program to be downloaded to a computer and run without an internet connection. According to a research study conducted by Meerbaum-Salant *et al.* [34], most students can achieve a reasonable level of knowledge in Computer Science through Scratch. However, difficulties arise when teaching specific topics that require a greater level of abstraction. This can, however, be solved if students are supported in the process by the teacher. Otherwise, the study concluded, most students will only use it as a tool to create media and not as a way to learn programming as originally planned.

2) *Snap!*: Snap! [35] is another visual block-based programming language whose design is based on Scratch but with some additional features. Snap! similarly to Scratch, is web-based, but also has the possibility to be run offline through a browser.

Snap! is more suitable for older audiences than Scratch and can be used as an introduction to computer science tool. A study with high school students using Snap! and Java was conducted by Weintrop and Wilensky [35]. The students found the blocks-based approach to be easier than Java – which is in accordance with the view that blocks-based programming is more accessible to novice programmers. According to the study findings, this was due to the fact that blocks are easier to read, due to the visual nature of the blocks that provide cues on how they can be used, they are easier to compose, and serve as memory aids.

3) *Alice*: Alice [36] is a block-based programming environment that aims to motivate learners to program by promoting their creativity, through the creation of animations, interactive narratives or simple 3D games. Like the other programming environments it removes syntax errors by allowing learners to construct a script by dragging and dropping code elements, and linking them in desired sequence. A study that focused on Storytelling Alice (a programming environment based on Alice 2) which facilitates the creation of stories by students (with high-level animations, a storytelling gallery, and a story-based tutorial) found that although both writing code with Generic Alice and Storytelling Alice were equally successful at teaching programming concepts to girls, with Storytelling Alice, girls were more motivated and increased their time programming, although they found both programs equally entertaining [37]. Motivation and time spent programming have a positive impact in programming performance – thus making programming more motivating to girls and maximizing the time they spent on it, so this may be a way to increase women's participation in computer science.

Alice allows for the creation of virtual worlds with a Virtual World Editor where learners can add 3D objects and add functions and methods to existing 3D objects. Once the Virtual World is created the learner can write code to develop the logic of the game/narrative/animation [37].

Alice is a good programming language for learners with no previous experience, as it allows them to see how the animated programs run as they write their code [38].

4) *Tynker*: Tynker is an educational programming platform based on a drag-and-drop visual programming language. Unlike all the other programming environments presented so far, this one is a commercial product. The main difference from the others is that it approaches coding as a game, in which the users are required to create a program in order to make the screen characters move, interact, and achieve different tasks. It presents problems the players need to solve, making it easy for children to begin recognizing patterns. Another addition is the possibility to see the code-equivalent in JavaScript. Tynker also provides a built-in tutor to give step-by-step instructions, so that the learner can learn how to apply coding concepts. The learner is also able to switch between the blocks and

JavaScript code, enabling them to understand the block in a text-based programming language [42].

5) *Code Combat*: Code Combat is a web-based adventure game with learning plans both for individual students and classes and resources for teachers to use during classes. It has more than 100 free-to-play and subscriber-only levels. Learners are required to write their own code (either in JavaScript or Python) in order to make the characters move, interact, and achieve different tasks. It is not block-based, requiring learners to write their own code. It provides hints along the way and introduces concepts gradually. The game is divided in 5 different worlds, introducing different concepts as the player progresses through the game: 1) Kithgard Dungeon – syntax, methods, parameters, strings, loops and variables; 2) Backwoods Forest – If/else, Boolean logic, relational operators, functions, object properties, event handling, input handling; 3) Sarven Desert – Arithmetic, counters, while-loops, break, continue, arrays, string comparison, finding min/max; 4) Cloudrip Mountain – Object literals, remote method, invocation, for-loops, complex functions, drawing, modulo; 5) Kelvintaph Glacier – Advanced Techniques.

According to Grivokostopoulou *et al.* [40], the game's educational content covers aspects like Algorithms Design, Fundamental Programming Concepts (such as Syntax & Semantics, Variables & Primitive Data Types; Expressions & Assignments, Input & Output, Conditionals & Iteratives, Functions & Parameters, and Recursion) and Fundamental Data Structures (such as Arrays, Heterogeneous Aggregates, and Abstract Data Types).

6) *Human Resource Machine*: Human Resource Machine is a commercial puzzle game based on a visual programming language. In this game, the player has to automate a task by programming an office worker to do his task. The worker who successfully finishes the tasks is promoted to different levels and progresses through different puzzles. The player must create a sequence of moves, by dragging and dropping instructions, with new commands being gradually introduced to accomplish more complex operations. Through a visual programming language based in blocks, it teaches students fundamental programming concepts, like input & output and conditionals & iteratives, as well as teaching algorithm comparison. It is mostly useful for learning how to write assembly-code [42].

7) *LightBot*: LightBot is a puzzle game, with mechanics similar to those of Human Resource Machine, requiring programming logic in order to progress through the levels. The player must guide a robot in order to light up tiles and solve levels, featuring 40 levels and 20 challenge stars. The commands used in Lightbot appear as icons.

In Lightbot the players develop problem solving skills while learning different fundamental programming concepts, such as Conditionals, Iterations and Recursion, as well as debugging strategies. As the space for the tiles is limited, the learner must also consider code size.

A study conducted by Mathrani *et al.* [45] using Lightbot with students found that they enjoyed playing the games and that this was an effective way of learning programming

concepts like functions, procedures, conditionals and recursions. Most participating students agreed that the approach was an effective way for them to understand concepts that would otherwise be more difficult.

8) *May's Journey*: *May's Journey* [46] is the only game from this list that is directly targeted toward middle school girls. It is a 3D puzzle game in which the players solve a maze through the means of the game's custom programming language, which is inspired by Java. The game was designed with middle school girls' interests in mind, hoping to attract them to Computer Science by teaching the basics of programming. The developer employed pseudo-code in order to facilitate the transition between visual programming and general-purpose programming languages. The teaching content is: basic instructions and sequence logic, loops, variables, if statements, comparators and Boolean logic, operations on integers, operations on strings.

In the story, the hero is a girl who lives in a world that is falling apart and is separated from her friend, needing to fix the game world and solve mysteries. Each part of the mystery is revealed gradually, motivating the players to play more. As pre-teens are the main target group, they made the main character look like a middle school girl, so that the player can identify herself with the girl. The design also had in mind girls' preferences towards design (high lightness, warm color scheme and less aggressive illustrations).

9) *No Bug's Snack Bar*: *No Bug's Snack Bar* [47] is a research serious game with a drag and drop block-based approach to make the game independent from learning the syntax of the computer programming language, focusing on problem solving. Its learning outcomes are: Variable Manipulation, Sequence of Actions, Conditional & Iterativeness, and Debugging Strategies.

In the game, the main character has to work at a Snack Bar, passing through different missions using code. One innovation included in this game is the fact that there's a tool incorporated for teachers to monitor how students are progressing through the game. There's also a gamification approach, with the student gaining points as they progress through the games and find ways to better their solutions to the problems presented in order to maximize the points gained.

10) *Robot ON!*: *Robot ON!* [48] is a research puzzle-type game aimed at undergraduate students who are learning C++. In this game, the player is a scientist that has to activate a 'Mech Suit', doing so through a series of tasks. Once the player finishes a level, he activates a new robot system. It also has a framework in which instructors can create their own levels using other programming languages. The player has different tools, which are color coded, with the color on the code corresponding to the different tools. The player is introduced to different tools one at a time, accompanied by tutorials.

Instead of focusing on writing code, this game focuses on programming comprehension, in order to teach debugging skills and understand code written by others. As the game progresses, the player is given the following tools: 1) activator tool (to learn control flow); 2) commenter and un-commenter

tools (to learn code behavior); 3) namer tool (to learn variable purpose); and 4) checker tool (to learn data flow) [43].

11) *CMX*: *CMX* [49] is a Massive Multiplayer Online Role Playing Game (MMORPG) for learning programming. In the game there are two teams — crackers and hackers — who compete against each other to find the passwords on a global toxic waste factory. They are assisted by Senseis, that assist them in learning the programming language C. The game includes three levels with Senseis, with each one holding one password and unlocking another Sensei. The students seemed to increase their comprehension level of C — they could not only answer theory questions, but also lay executable programs by dragging and dropping tools, as well as writing programs in C.

12) *Run Marco!*: *Run Marco* [50] is a puzzle game with a drag and drop block-based approach. In the game a player has to help a young explorer, either boy or a girl, to find the right way. The game is available in more than 30 different languages, making it easier for students to understand the code as it is in their own language. Students learn about sequencing the commands, iteration, and conditions. In every chapter the player's solution is run in order to check its correctness. The program runs step-by-step, making it easier for the player to track and trace the errors in the program [50].

III. CODING4GIRLS INITIATIVE

CODING4GIRLS is an ongoing international project lead by a consortium that gathers educational experts, computer science educators, computer scientists, teachers and practitioners from different European universities, software development companies and educational organizations. The overall goal is to increase the number of women that participate in and lead computer science business projects resulting in a more diverse workforce from the promotion of better opportunities for girls. To achieve that goal, the project created a learning methodology to encourage and motivate girls to consider computer science education and careers. As such, the main target groups of the project are:

- Students from the last years of basic education and the earlier years of secondary education taking in particular attention girls;
- Teachers of basic and secondary education.

The CODING4GIRLS initiative's pedagogical approach has the following educational objectives:

1. To prepare young learners to enter computer science careers by building programming skills;
2. To enable learners to apply the newly developed programming knowledge in wider learning contexts;
3. To build transversal competencies related to programming, such as analytical and critical thinking;
4. To foster positive attitudes towards computer science among girls and boys with the objective of promoting the uptake of related educational and career paths;
5. To raise awareness on the links between ICT and the real-world through learning scenarios that demonstrate how ICT solutions can enhance quality of life and address common needs;

6. To empower learners to think entrepreneurially by introducing solutions to real-world problems through design thinking mind sets.

The above learning objectives address directly the reasons that lead learners not to choose careers in computer science, namely: a) their skill unpreparedness; and b) their misperception in relation to the benefits and suitability of computer science careers for all, but particularly for girls.

The project's pedagogical approach is heavily linked to creativity and human-centered solutions to demonstrate the relation of the field with real life by encouraging learners to design and code games that address specific world needs or issues.

Recognizing the importance that teachers and the social environment of learners play in the choice of academic and career choices, CODING4GIRLS further builds teacher capabilities and positive perceptions and attitudes in relation to the participation of girls in computer science by developing educator support content for integrating computer science related activities into their instructional practices.

The following are the expected concrete results of the implementation of the CODING4GIRLS initiative:

1. A methodological learning framework based on design thinking approaches for building programming skills among young girls and promoting the uptake of computer science careers through positive attitudes and preparedness for entering the world of work;

2. A proof-of-concept approach for building programming skills among young girls in basic and secondary education through the design and development of serious games.

3. Instructional support content in the form of learning sheets and best practice videos that will facilitate the integration of the proposed design thinking methodologies and the serious game approach into existing school practices, enriching learning for the benefit of the ultimate end-users, i.e. learners and teachers. The learning sheets will facilitate an easier integration of the proposed game-based learning framework into existing instructional practices;

4. An extended set of end-to-end and modular learning activities based on the proposed serious game approach for building programming skills among girls;

5. A benchmarking strategy that will provide a comprehensive guide on how to analyze the relevance, acceptance, effectiveness, and quality of the approach and instructional support content that will be established through quantitative and qualitative evaluation methodologies. The strategy will provide insight to external interested parties, including teachers and policy makers, on how to evaluate the proposed learning frameworks and software tools in relation to desired and expected learning outcomes on programming.

A. Design of CODING4GIRLS Framework

Work started with the establishment of the theoretical background and framework of the project which is a necessary first step before any methodological learning solutions and supporting technical solutions, namely software, may be proposed. The background was established by first analyzing the

current situation which involved collecting the documentation of current practices in schools in several countries, existing activities beyond the formal school system, related software and services developed by others, national and global level strategies and policies for the development of programming, and more generally STEM skills, links of education to industry needs, and more. It was also important to keep in mind that the project was addressing real and present needs in school education that were not totally met in previous initiatives. Those were:

- the need to attract talented girls into computer science careers for their own benefit, as these careers are highly rewarding personally and professionally, and for the benefit of society through the increase of available staff with relevant personal and technical skills;
- the need to address the gap in the participation of girls in computer science education, promoting equal opportunities in education and employment for girls and boys.

As such, the learning framework was based on the analysis of the status quo and the state-of-the-art on how technology-enhanced learning can promote the development of programming skills. This analysis led the proposal of the learning framework and the design and development of the necessary tools, which is the current stage of the project.

The CODING4GIRLS methodological learning framework for building programming skills among young girls (and boys) in basic and secondary education:

- Identified the stakeholders in secondary education that stand to gain from the CODING4GIRLS proposed learning methodologies and tools and documents their needs and characteristics. The work analyzed the needs of learners, teachers, and others;
- Identified the current situation in school education on the subject of teaching programming; documents current practices in schools on teaching programming in countries in which the CODING4GIRLS consortium has partners;
- Documented informal and non-formal practices on programming skill development inside and outside of school curricula that may exist in countries in which the CODING4GIRLS consortium has partners;
- Documented the level of deployment of serious games in the context of digital skills development in the countries in which the CODING4GIRLS consortium has partners;
- Documented other tools and services that may be related to the development of programming skills;
- Documented and analyzed learning requirements of school learners in secondary education in relation to building programming skills towards up-taking careers in ICT by both girls and boys;
- Analyzed the skill building needs for teachers in basic and secondary education on promoting programming skills and positive attitudes among girls and boys on the rich perspective of ICT careers;
- Created a design thinking game-based learning methodological framework that can enhance existing practices on programming skills development. Design thinking is rooted in the conviction that product and service design

must take into account wider world needs, including environmental ones. In ICT design thinking fits well because it provides a human-centered approach to innovation ensuring that digital services and tools address everyday needs and community interests, thus demonstrating the link between ICT and real-life. Demonstrating this link is important for attracting both girls and boys into long-term, satisfying, and engaging careers in computer science.

The framework exploits “design thinking” approaches that encourage learners to think innovatively and critically and to take into account user needs. Through the proposed design thinking learning methodology learners are exposed at an early age to processes applied by the ICT sector. This learning methodology is very well suited for the development of programming skills for the ICT sector among young boys and girls also particularly because students learn to code through coding games. Through the proposed design thinking learning methodology learners will be exposed at an early age to processes applied by the ICT sector.

B. Implementation

Implementation is based on a collaborative learning platform. Teachers are able to define a course starting by defining overarching issues, creating challenges (mini-games, explanations, and exercises) and distributing the course material for students to solve. They also have an overview of the course and of students’ activities. Students are able to participate in their teacher’s courses, communicate with each other, and participate in brainstormings. Students solve the challenges which consist of designing and developing minigames that require certain programming concepts. After the challenge they solve the overarching programming problem and see other students’ solutions of the same problem.

A “low entry high ceiling approach” is used because it allows learners to start with easy problems encouraging continued participation through more challenging tasks. Learners are going to be presented with “half-baked” scenarios in which a solution to an actual problem is partially ready challenging learners to finish the implementation by developing small and manageable modules.

Snap! was chosen as the programming environment most appropriate for our target group as it was considered that students could come in contact with a wider range of programming concepts than those found in other visual programming languages.

A set of basic and secondary schools will be the core for pilots of the project and their teachers and students will be directly involved from the first moment throughout the different stages of the project, by:

- Providing feedback on the specification and design of the methodology and tools;
- Being involved in the development, and doing beta testing of the evolving products;
- Participating in the piloting stage, collaborating with teachers and students from other countries, contributing new

problems and challenges, as well as evaluating the tools and methodology;

About 100 teachers and 500 girls are expected to be directly involved in the project at this stage. Nevertheless CODING4GIRLS is not a girls-only club! Boys will actively participate in the project activities because the idea is to promote equality of gender and assuming a girls-only attitude would generate a strong bias in terms of activities and participation. Therefore students (and teachers) involved in the activities are from both genders.

C. Instructional Support for Teachers

Instructional support contents target educators and aims at providing the necessary how-to good practice instructions that allows them to integrate the proposed learning methodologies and tools into their already well developed instructional practices. More specifically, this content:

- Provides teachers with the information they need for enhancing their teaching on programming through the proposed serious games approach and design thinking learning methodologies;
- Helps build teacher skills, confidence, and motivation in the deployment of emerging ICT, and specifically serious games, as a complementary learning tool;
- Builds the skills of teachers on the integration of ICT into instructional practices through supporting content;
- Facilitates a more effective adoption of project outcomes on developing programming skills among girls through serious games into wider, blended learning school practices.

The instructional support contents have the form of:

- Learning sheets that describe end-to-end blended learning activities that deploy the CODING4GIRLS serious game and design thinking approaches;
- How-to videos that act as a visual aid on how to use the features and the functionality of the CODING4GIRLS serious game;
- A user guide on the CODING4GIRLS serious game approach for building programming skills among girls through design thinking approaches.

D. Evaluation

The proposed methodologies and proof-of-concept tools will be evaluated in real-life contexts through the engagement of external groups of learners and teachers in several countries. To ensure impact, results will be widely disseminated to stakeholders, including learners, educators, policy makers, and the general public.

With the validation activities it is aimed to ensure that the proposed framework meets the needs of the identified target groups in terms of relevance to learning needs, acceptance, usability, and effectiveness. These high level objectives will be pursued through on-going internal and external validation that started early on in the project implementation period and engages different stakeholders, students, teachers, experts and national education institutions. The continuous nature of

the evaluation ensures that the input of all the stakeholder representatives are going to be taken into account during the design and implementation process of the proposed tools and methodologies thus ensuring that the final result addresses real educational needs in terms of building programming skills among girls and boys.

The evaluation takes place at three levels:

- Internally, engaging staff from the project partners;
- Externally, engaging external groups of users that are representative of stakeholders, and specifically learners and teachers;
- Through externals expert who will provide a professional opinion on the completion, quality, and effectiveness of project outputs.

The consortium will explore both “qualitative evaluation” models that aim to establish opinions, attitudes and perceptions and “quantitative evaluation” models, which are based on the analysis of statistics will also be explored.

E. Impact

The project will have impacts on different levels. For teachers, learners and education providers the impact is expected to be:

- Better understanding of the opportunities to engage in rewarding computer science careers in today’s technology and innovation-driven economy;
- Better understanding of the skills that learners need to build in order to be prepared to enter a computer science careers;
- Better understanding of the need to instill positive perceptions towards computer science careers among girls, boys, and parents;
- Empowering learners to think of innovative ICT solutions to everyday problems taking also into account the bigger pictures of real-world community interests;
- Long-term integration of proposed services and tools into organizational practices related to enhancing school education and quality of learning processes;
- Enhanced capacity of teachers to integrate innovative ICT and emerging pedagogical frameworks into existing lifelong learning offerings targeting young learners and an enhanced capacity of teachers to link school activities to the needs of the world of work.

IV. CONCLUSION

The gap between male and female participation in computer science education and careers is a worldwide issue that contributes to the current lack of qualified staff for digital innovation companies. This gap must be addressed by introducing early methodological learning interventions that make computer science attractive to all by answering the following issues: a) misperception among educators, learners, parents, and youngsters on the suitability of computer science careers to girls and b) a wrong assumption of their insufficient preparedness to do it successfully.

CODING4GIRLS is a European initiative meant to raise awareness on the rich possibilities for professional and personal growth computer science offers and will build computer science skills for preparing girls (and boys) for future engagement in computer science in post-secondary education, vocational, or professional careers. Recognizing the importance that teachers and the social environment of learners can play in the choice of a learner’s academic and career decisions. CODING4GIRLS further aims at building teacher capabilities and positive parent perceptions and attitudes in relation to the participation of girls in computer science. Objectives are directly related to employability of learners through the development at school age of competencies related to work thus address ET 2020 objectives [52] and the objectives of the New Skills for New Jobs initiative. Objectives are also directly linked to PISA 2021 and ET 2020 objectives in relation to building transversal competencies that help learners excel in academics and at work independently of subject area [53].

At a global level the desired impact of the CODING4GIRLS initiative is to attract and engage all talented students, including both girls and boys, in computer science education and careers:

- To promote inclusive, sustainable growth through the effective preparedness of young individuals to enter professional sectors with high growth, such as computer science;
- To promote openness and innovation in education through the development of educational software applications that will be open for use by everyone;
- To promote strategic deployment of ICT in education through the promotion of the development of proof-of-concept ICT-based educational applications for open use in secondary education;
- To promote high quality education for all embedded in the digital era through initiatives that build programming skills;
- To link education to the needs of the world of work by building skills required in emerging innovation sectors, including computer science;
- To fight unemployment through focused educational approaches that build skill sets that are required in industry;
- To promote active citizenship as a result of social inclusion and enhanced confidence of individuals to lead productive and satisfying careers and lives as contributing members of their communities;
- To promote the availability of ICT tools for educational purposes;
- To promote collaboration among educators in different countries on enhancing educational policies, practices, and activities;
- To promote the development of transversal problem-solving and critical thinking skills; notably teachers in the field point to the lack of available solutions for school education, especially in digital form, for addressing these skills.

Work started with the establishment of the theoretical background and framework of the approach. This background was developed by first analyzing the current situation on programming skills development, which is a necessary first step before any methodological learning solutions and supporting technical solutions, namely software, may be proposed. The analysis of the current situation involved the documentation of current practices in schools in several countries, existing activities beyond the formal school system, related software and services developed by others, national and global level strategies and policies for the development of programming, and more generally STEM skills, links of education to industry needs, and more.

It is expected that CODING4GIRLS approach will be applied and evaluated during the next year and will bring positive results to overcome the gap between the number of girls and boys in ICT courses and careers.

REFERENCES

- [1] *UNESCO Science Report?: Towards 2030*, UNESCO, Paris, France, 2015.
- [2] *Women, Minorities, and Persons with Disabilities in Science and Engineering Statistics: 2019*, National Science Foundation National Center for Science and Engineering, Alexandria, VA, USA, 2019.
- [3] EC. (2013). *Women Active in the ICT Sector*. [Online]. Available: <https://op.europa.eu/en/publication-detail/-/publication/9153e169-bd6e-4cf4-8638-79e2e982b0a3/language-en>, doi: 10.2759/27822.
- [4] EEOC. (2016). *Diversity in High Tech*. [Online]. Available: <https://www.eeoc.gov/eeoc/statistics/reports/hightech/>
- [5] *Global Education Monitoring Report—Gender Report: Building Bridges for Gender Equality*, UNESCO, Paris, France, 2019.
- [6] EC. (2010). *New Skills for New Jobs*. [Online]. Available: <https://ec.europa.eu/social/BlobServlet?docId=4508&langId=en>
- [7] EU. (2013). *Women Active in the ICT Sector*. [Online]. Available: <https://op.europa.eu/en/publication-detail/-/publication/9153e169-bd6e-4cf4-8638-79e2e982b0a3/language-en/format-PDF/source-112575860>
- [8] B. E. G. Ramírez, C. A. Collazos, and C. González, “Gender differences in computing programs: Colombian case study,” in *Proc. 17th Int. Conf. Hum. Comput. Interact.*, Sep. 2016, pp. 1–3, doi: 10.1145/2998626.2998670.
- [9] A. Hosein, “Girls’ video gaming behaviour and undergraduate degree selection: A secondary data analysis approach,” *Comput. Hum. Behav.*, vol. 91, pp. 226–235, Feb. 2019, doi: 10.1016/j.chb.2018.10.001.
- [10] S. A. Alserri, “Gender-based engagement model for serious games,” *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 8, no. 4, pp. 1350–1357, 2018, doi: 10.18517/ijaseit.8.4.6490.
- [11] S. O. Akinola, “Computer programming skill and gender difference: An empirical study,” *Amer. J. Sci. Ind. Res.*, vol. 7, no. 1, pp. 1–9, 2015.
- [12] *Cracking the Code: Girls’ and Women’s Education in Science, Technology, Engineering and Mathematics (STEM)*, UNESCO, Paris, France, 2017.
- [13] L. J. Sax *et al.*, “Anatomy of an enduring gender gap: The evolution of women’s participation in computer science,” *J. Higher Educ.*, vol. 88, no. 2, pp. 258–293, 2017.
- [14] R. Rousi, “Evening the odds—educational computer games and 21st century coding literacy are girls’ things too,” in *Proc. EdMedia+ Innovate Learn.*, 2014, pp. 35–43.
- [15] A. Bartilla and C. Köppe, “Awareness seeds for more gender diversity in computer science education,” presented at the 20th Eur. Conf. Pattern Lang. Programs, Kaufbeuren, Germany, 2015.
- [16] G. Carmichael, “Girls, computer science, and games,” *ACM SIGCSE Bull.*, vol. 40, no. 4, pp. 107–110, Nov. 2008, doi: 10.1145/1473195.1473233.
- [17] G. Osborn. (2017). *Male and Female Gamers: How Their Similarities and Differences Shape the Games Market*. [Online]. Available: <https://newzoo.com/insights/articles/male-and-female-gamers-how-their-similarities-and-differences-shape-the-games-market/>
- [18] L. Vermeulen, “Girls will be girls: A study into differences in game design preferences across gender and player types,” presented at the Under Mask, Perspect. Gamer, Luton, U.K., 2011. [Online]. Available: <http://hdl.handle.net/1854/LU-1886961>
- [19] D. G. Oblinger, “The next generation of educational engagement,” *J. Interact. Media Educ.*, vol. 8, pp. 1–18, Aug. 2004, doi: 10.5334/2004-8-oblinger.
- [20] D. G. Oblinger. (2006). *Simulations, Games, and Learning*. [Online]. Available: <http://net.educause.edu/ir/library/pdf/ELI3004.pdf>
- [21] P. Escudeiro and C. V. de Carvalho, “Game-based language learning,” *Int. J. Inf. Educ. Technol.*, vol. 3, no. 6, p. 643, 2013.
- [22] C. V. Carvalho *et al.*, “Lean learning academy: An innovative framework for lean manufacturing training,” in *Proc. 1st Int. Conf. Portuguese Soc. Eng. Edu. (CISPPE)*, Oct. 2013, pp. 1–5.
- [23] J. S. Tashiro and D. Dunlap, “The impact of realism on learning engagement in educational games,” presented at the Conf. Future Play, Toronto, ON, Canada, 2007.
- [24] D. R. Michael and S. L. Chen, *Serious Games: Games That Educate, Train, and Inform*. Oshawa, ON, Canada: Thomson Course Technology PTR, 2006.
- [25] M. Zyda, “From visual simulation to virtual reality to games,” *J. Comput.*, vol. 38, no. 9, pp. 25–32, 2005, doi: 10.1109/mc.2005.297.
- [26] M. Prensky, “Digital game-based learning,” *Comput. Entertainment*, vol. 1, no. 1, p. 21, Oct. 2003, doi: 10.1145/950566.950596.
- [27] C. V. de Carvalho, R. Costa, P. Bessa, L. Monterrubio, and J. Santos, “Developing entrepreneurship skills with a serious game,” in *Proc. Int. Conf. Hum.-Comput. Interact.*, 2019, pp. 351–363.
- [28] C. V. de Carvalho, P. Bessa, R. Costa, A. Barata, and A. Costa, “LearnIt: A serious game to support study methods in engineering education,” in *Proc. IEEE Global Eng. Edu. Conf. (EDUCON)*, Apr. 2019, pp. 959–967.
- [29] S. de Freitas and S. Jarvis, “Serious games—engaging training solutions: A research and development project for supporting training needs,” *Brit. J. Educ. Technol.*, vol. 38, no. 3, pp. 523–525, May 2007, doi: 10.1111/j.1467-8535.2007.00716.x.
- [30] C. Kazimoglu, M. Kiernan, L. Bacon, and L. Mackinnon, “A serious game for developing computational thinking and learning introductory computer programming,” *Procedia-Social Behav. Sci.*, vol. 47, pp. 1991–1999, Jan. 2012, doi: 10.1016/j.sbspro.2012.06.938.
- [31] M. Resnick, “Scratch: Programming for all,” *Commun. ACM*, vol. 52, no. 11, pp. 60–68, 2009, doi: 10.1145/1592761.1592779.
- [32] I. Ouahbi, F. Kaddari, H. Darhmaoui, A. Elachqar, and S. Lahmine, “Learning basic programming concepts by creating games with scratch programming environment,” *Procedia-Social Behav. Sci.*, vol. 191, pp. 1479–1482, Jun. 2015, doi: 10.1016/j.sbspro.2015.04.224.
- [33] U. Wolz, J. Maloney, and S. M. Pulimood, “‘scratch’ your way to introductory cs,” *ACM SIGCSE Bull.*, vol. 40, no. 1, pp. 298–299, Feb. 2008.
- [34] O. Meerbaum-Salant, M. Armoni, and M. Ben-Ari, “Learning computer science concepts with scratch,” *Comput. Sci. Edu.*, vol. 23, no. 3, pp. 239–264, Sep. 2013, doi: 10.1080/08993408.2013.832022.
- [35] D. Weintrop and U. Wilensky, “To block or not to block, that is the question: Students’ perceptions of blocks-based programming,” presented at the 14th Int. Conf. Interact. Design Children, Boston, MA, USA, 2015.
- [36] C. Kelleher, R. Pausch, and S. Kiesler, “Storytelling alice motivates middle school girls to learn computer programming,” presented at the SIGCHI Conf. Hum. Factors Comput. Syst., San Jose, CA, USA, 2007.
- [37] E. R. Sykes, “Determining the effectiveness of the 3D alice programming environment at the computer science I level,” *J. Educ. Comput. Res.*, vol. 36, no. 2, pp. 223–244, Mar. 2007, doi: 10.2190/J175-Q735-1345-270M.
- [38] S. Cooper, W. Dann, and R. Pausch, “Alice: A 3-D tool for introductory programming concepts,” *J. Comput. Sci. Colleges*, vol. 15, no. 5, pp. 107–116, 2000.
- [39] S. S. Shabanah, J. X. Chen, H. Wechsler, D. Carr, and E. Wegman, “Designing computer games to teach algorithms,” presented at the 7th Int. Conf. Inf. Technol., New Generations, Las Vegas, NV, USA, 2010.
- [40] F. Grivokostopoulou, I. Perikos, and I. Hatzilygeroudis, “An educational game for teaching search algorithms,” in *Proc. 8th Int. Conf. Comput. Supported Edu.*, 2016, pp. 129–136, doi: 10.5220/0005864601290136.
- [41] S. Combéfis *et al.*, “Learning programming through games and contests: Overview, characterisation and discussion,” *Olympiads Inform.*, vol. 10, no. 1, pp. 39–60, Jul. 2016, doi: 10.15388/oi.2016.03.

- [42] M. A. Miljanovic and J. S. Bradbury, "A review of serious games for programming," presented at the 4th Joint Int. Conf. Serious Games, Darmstadt, Germany, Nov. 2018.
- [43] I. Frankovic, N. Hoic-Bozic, and L. N. Prskalo, "Serious games for learning programming concepts," presented at the 8th Int. Conf. Future Educ., Florence, Italy, 2018.
- [44] Y. B. Kafai, "Playing and making games for learning: Instructionist and constructionist perspectives for game studies," *Games Culture*, vol. 1, no. 1, pp. 36–40, Jan. 2006, doi: [10.1177/1555412005281767](https://doi.org/10.1177/1555412005281767).
- [45] A. Mathrani, S. Christian, and A. Ponder-Sutton, "PlayIT: Game based learning approach for teaching programming concepts," *Educ. Technol. Soc.*, vol. 19, no. 2, pp. 5–17, 2016.
- [46] C. Jemmali, *May's Journey: A Serious Game to Teach Middle and High School Girls Programming*. Worcester, MA, USA: Worcester Polytechnic Institute, 2016. [Online]. Available: <https://digitalcommons.wpi.edu/etd-theses/455>
- [47] C. Johnson *et al.*, "Game development for computer science education," presented at the ITiCSE Working Group Rep., Arequipa, Peru, 2016.
- [48] M. A. Miljanovic and J. S. Bradbury, "Robot on! A serious game for improving programming comprehension," presented at the 5th Int. Workshop Games Softw. Eng., Austin, TX, USA, 2016.
- [49] C. Malliarakis, "CMX: Implementing an MMORPG for learning programming," presented at the 8th Eur. Conf. Games Based Learn., Berlin, Germany, 2014.
- [50] D. Kumar, "Digital playgrounds for early computing education," *ACM Inroads*, vol. 5, no. 1, pp. 20–21, Mar. 2014, doi: [10.1145/2568195.2568200](https://doi.org/10.1145/2568195.2568200).
- [51] A. Giannakoulas and S. Xinogalos, "Current trends in on-line games for teaching programming concepts to primary school students," in *Proc. Int. Conf. Technol. Innov. Learn., Teach. Educ.*, vol. 993, 2019, pp. 62–78.
- [52] *European Commission, European Policy Cooperation (ET 2020 Framework)*. Accessed: Feb. 15, 2018. [Online]. Available: https://ec.europa.eu/education/policies/european-policy-cooperation/et2020-framework_en
- [53] OECD. (2019). *PISA 2021 ICT Framework*. [Online]. Available: <http://www.oecd.org/pisa/sitedocument/PISA-2021-ICT-framework.pdf>