Co-funded by
the European Union

QSCAR
CODING CAMPS

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them

unibz Freie Universität Bozen
Libera Università di Bolzano
Università Liedia de Bulsan

Università degli Studi di Bari Aldo Moro

UJI Universitat Jaume I

Jyväskylän Yliopisto
University of Jyväskylä

Scuola Italiana Alto Adige

Autonome Provinz Bozen - Südtirol
Provincia Autonoma di Bolzano - Alto Adige
Provinzia Autonoma de Bulsan - Südtirol

MIM Ministero dell'Istruzione e del Merito

Diputació de Castelló

Handbook
1st release
(English version)

Project 101132432 – OSCAR

Promoting crOss-cutting digital Skills through Europe-wide non- Conventional leArning experiences

Table of contents

Handbook Purpose

The purpose of the handbook is to provide recommendations and implementation guidelines for organizing online/hybrid coding camps that replicate the face-to-face dynamics. The handbook takes a holistic approach to address various aspects, such as logistics, spaces, multilingualism, and inclusion.
The handbook will be structured according to the ADDIE model[1] an instructional design model for e-learning activities, which consists of five phases, as shown in Figure 1.

- **Analysis:** to identify the problems, objectives, environment, and previous learner's skills. Outcome: an analysis of training needs and plan of the coding camp.
- **Design:** to design a planned approach to address the learning challenges. Outcome: holistic overview of the coding camp design.
- **Development:** to build and assemble the designed content assets. Outcome: course content and material.
- **Implementation:** to transform the plan into action. Outcome: Detailed timeline with instructions.
- **Evaluation:** to check if the objectives are achieved by the instructional design and didactic content. Outcome: guide to evaluate the effectiveness of the coding camps and actionable changes for the future editions.

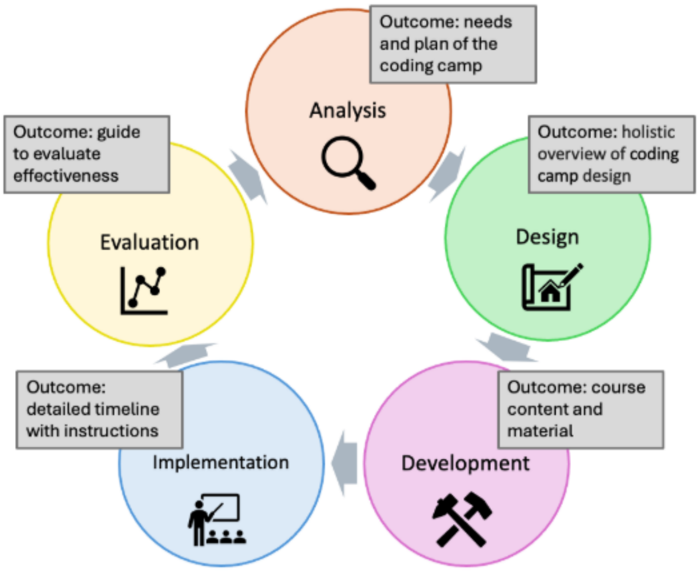


Figure 1. Structure of the handbook, based on the five phases of the ADDIE model.

[1] *Branch, R. (2010). Instructional design: the ADDIE approach. Boston, MA: Springer US.*

The activities in the analysis phase of the handbook will not be necessary for the reader, as they will be using the handbook to organize a coding camp that fulfils the training needs identified by the OSCAR project. Therefore, the analysis phase chapter will provide an overview of the coding camp objectives, target groups, and learning objectives. In the subsequent chapters of the handbook, the reader will receive guidance on how to organize the coding camp based on the results of the OSCAR project.

Chapter 1 - Coding Camp Analysis

The OSCAR project aims to promote cross-cutting digital skills among high school students through non-conventional learning experiences that are online or hybrid. The focus is on timely and motivating topics such as serious games, extended reality, and app development. This chapter describes the training needs based on three key resources (shown in Figure 2):

a. cross-cutting policies that underlie the OSCAR project as per the Grant Agreement,
b. the results of a roundtable discussion with project stakeholders, and
c. the collective experience and research findings of the consortium partners.

The intersection of these resources is summarized in the following figure:

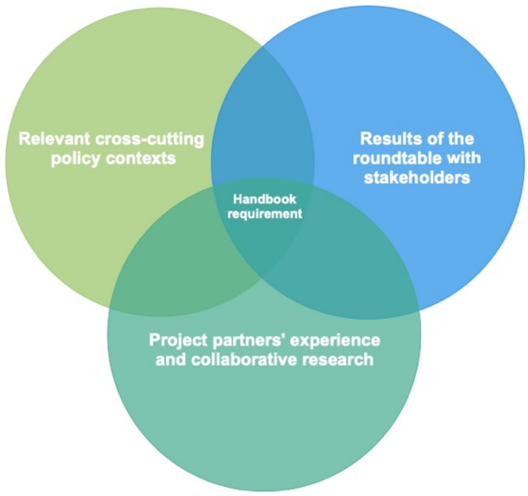


Figure 2. The handbook content requirements are the result of the intersection of three key resources

Learning objectives. At OSCAR's coding camps, participants will collaboratively develop innovative digital solutions to tackle sustainability challenges, with a focus on climate considerations across Europe. They will specifically concentrate on creating serious games, mobile apps, and XR solutions.

Skills. OSCAR's coding camps will promote cross-cutting skills in accordance with relevant policy context, using the reference frameworks created by the European Commission and the Council of Europe to support the conceptualisation of the Key competences and their key terminology, i.e., the DigComp framework (hereafter referred to as DigComp), the European Skills Agenda[2] (hereafter referred to as ESA),

[2] https://ec.europa.eu/social/main.jsp?catId=1223&langId=en

and cross- cutting competencies for sustainability[3] (hereafter referred to as SDG). As shown in Figure 3, OSCAR's coding camps will promote skills taking into consideration the stakeholder preferences expressed during the roundtable.

1. **Problem solving**, focusing on the following competences (DigComp):
    - Solving technical problem, i.e., identifying technical problems when operating devices and using digital environments, and solving them (from troubleshooting to solving more complex problems).
    - Identifying needs and technological responses, i.e., assessing needs and identifying, evaluating, selecting, and using digital tools and possible technological responses to solve them.
    - Creatively using digital technologies, i.e., using digital tools and technologies to create knowledge and to innovate processes and products. Engage individually and collectively in cognitive processing to understand and resolve conceptual problems and problem situations in digital environments.
2. **Communication and collaboration**, focusing on the following competences (DigComp):
    - Interacting through digital technologies, i.e., interacting through a variety of digital technologies and understanding appropriate digital communication means for a given context.
    - Sharing through digital technologies, i.e., sharing data, information, and digital content with others through appropriate digital technologies. Acting as an intermediary, knowing about referencing and attribution practices.
    - Engaging in citizenship through digital technologies, i.e., participating in society using public and private digital services. Seeking opportunities for self-empowerment and for participatory citizenship through appropriate digital technologies.
    - Collaborating through digital technologies i.e., using digital tools and technologies for collaborative processes, and for co-construction and co-creation of data, resources, and knowledge.
    - Promoting communication and collaboration skills will help develop peer-support skills, which were identified as a preference by stakeholders during the roundtable.
3. **Digital content creation**, focusing on the following competences (DigComp):

[3] UNESCO (2017) Education for Sustainable Development Goals: learning objectives. https://unesdoc.unesco.org/ark:/48223/pf0000247444.locale=en

- Developing digital content, i.e., creating and editing digital content in different formats, to express oneself through digital means.
- Integrating and re-elaborating digital content, i.e., modifying, refining, and integrating new information and content into an existing body of knowledge and resources to create new, original, and relevant content and knowledge.
- Programming, i.e., planning and developing a sequence of understandable instructions for a computing system to solve a given problem or to perform a specific task.

4. **Entrepreneurial skills**, i.e., acting upon opportunities and ideas and transforming them into value for others. The value that is created can be financial, cultural, or social[5]. Together with creatively using digital technologies (DigComp competence 5.3), entrepreneurial skills can help transform ideas into value for oneself and others (SDG). This can be obtained by learning how to create a marketing plan or preparing a pitch for a product
5. **Civic and environmental skills** [ESA] focusing on
   - Safety (protecting the environment) (DigComp), i.e., being aware of the environmental impact of digital technologies and their use.
   - Skills to support the green and digital transition (ESA) i.e., developing solutions for society with climate considerations

For each skill, the expected proficiency level will be defined using an approach that can be shared Europe-wide, e.g., using the proficiency levels described in the Digital Competence Framework.

Problem solving	5.1, 5.2 5.3	action 7	transversal
Communication and collaboration	2.1, 2.2 2.3, 2.5	action 7	transversal
Digital content creation	3.1, 3.2 3.4		
Entrepreneurial and transversal skills		action 7	
Civic and environmental competency	4.4	action 6 action 8	
	DigComp framework	European Skills Agenda	Education for SDGs

Figure 3. Skills promoted by the OSCAR coding camps and their relevance to policy contexts. Skills are sorted based on stakeholder preferences expressed during roundtable discussions (as outlined in deliverable D2.1 Roundtable with stakeholders).

Target group. OSCAR coding camps target high school students (aged 15-19) attending different schools (from non-vocational to computer science) with diversified disciplinary backgrounds and little or no previous knowledge of software development. The coding camps will promote equity and inclusivity by providing fair treatment and equal opportunities to all participants, including individuals who may have different educational needs or are socially at risk. In particular, the coding camps will provide opportunities for girls to learn and engage in Computer Science.

1.1 References

Branch, R. (2010). Instructional design: the ADDIE approach. Boston, MA: Springer US.

Vuorikari, R., Kluzer, S. and Punie, Y., DigComp 2.2: The Digital Competence Framework for Citizens - With new examples of knowledge, skills and attitudes, EUR 31006 EN, Publications Office of the European Union, Luxembourg, 2022, ISBN 978-92-76-48882-8, doi:10.2760/115376, JRC128415

Chapter 2 - Coding Camp Design

This chapter provides a holistic overview of a coding camp design, showcasing how the OSCAR platform can be used to achieve the desired outcome. The chapter is based on cross-cutting policies, the results of a roundtable discussion with project stakeholders, and the collective experience and research findings of the consortium partners. Throughout the chapter, **orange boxes labelled "OSCAR findings"** summarize key insights, data, and lessons learned from the coding camps organized by the project partners, offering you evidence-based recommendations briefly. **Blue boxes labelled "In the OSCAR platform"** show how the platform implements or supports the strategies described in the preceding paragraphs, helping you translate ideas into concrete tools and features on the platform. **Tables and figures** synthesize the content as ready-to-adapt tools when planning coding camps.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

OSCAR findings
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

In the OSCAR platform (first release)
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Icons presented will highlight content that is specifically relevant to **app development**, **eXtended Reality (XR) experiences**, and **serious game development**.

2.1 Strategies for online/hybrid non-conventional learning experiences

2.1.1 Schedule of the coding camp

Porras et al. (2019) analyzed 51 papers and found that most coding camps and hackathons (45%) last two days (48 hours, typical for hackathons). Additionally, 25% of the events are 24 hours or shorter, while 22% last between 3 to 5 days. In 2024, the analysis of 11 editions of the same coding camp showed that a duration of 20 hours, spread over 5 days, is empirically evaluated as an effective solution (Fronza and Corral, 2024). In the same research work, the authors suggested organizing a coding camp in the following three parts:

1. First part: explaining general concepts about problem understanding and problem- solving (4 hours);
2. Second part: teaching the fundamental programming concepts and the development tools (4 hours);
3. Third part: students work in teams to apply knowledge and skills with a hands-on approach. At the end, teams present their projects (12 hours).

Given their problem-based learning approach, coding camps are dynamic and require a **flexible schedule**. This flexibility allows facilitators to adjust session content based on the participants' needs, skill levels, and the challenges they encounter. Incorporating breaks and interactive activities enhances engagement and prevents cognitive overload. Games and fun are a cornerstone of a successful outcome in all their delivery formats (Fronza et al., 2022) and represent an opportunity to deliver key learning and increase engagement, emotion, and motivation (Sorathia and Servidio, 2012). Games can mimic the "real-world" professional context, while teaching several topics including requirements engineering (Kurkovsky et al., 2019) global software engineering (Sablis et al., 2019), cross-domain stakeholder-alignment (Köhlke et al., 2021), and testing (Lőrincz et al., 2021). Soft skills and professional awareness can be taught through games and similar experiences (Fronza and Corral, 2024).

OSCAR findings
A single intensive week with afternoon sessions was found to be successful and led to a completion rate of 76.8% and a high level of reported participants' satisfaction.

In the OSCAR platform (first release)
Timelines for scheduling activities, assignments, and collaboration sessions can have many different visualizations, from simple lists to tables, calendars, and dedicated progress trackers. Participants can track upcoming events, making the flexible structure of dynamic coding camps easily manageable.

2.1.2 Structure of each session

Each session should provide a balanced, flexible, and engaging learning experience in all delivery formats. Using a well-organized table for planning, like the one illustrated below, is highly effective.

Table 1. Template for planning each session of a coding camp

Session	Hours	Objective	Content	Skills	Assessment
Session number	Number of hours	Main objective of the session	Material that students will learn during the session	List of skills fostered in this session	Assessment strategy for each skill

OSCAR findings

In OSCAR, each session of a coding camp was designed by filling Table 1 with: (i) skills S1–S16 defined in chapter 1; (ii) detailed activities that directly connected content, tools, and skills; (iii) observable indicators for assessment.

In the OSCAR platform (first release)
Facilitators can upload learning materials, interactive tasks, and skill assessments directly into the provided session templates. These documents align with structured session planning tables, promoting clarity in skill acquisition and evaluation. Moreover, the OSCAR platform gathers all the necessary data for evaluating learning outcomes, measuring event impact, and identifying areas for future improvement. Participants receive recognition in the form of badges for completed projects, milestones, and skills, enhancing their sense of achievement. Completed projects can be published in an external repository and linked to individual profiles, or as answers to project tasks on the platform. Projects can be evaluated directly on the platform. Participants can also reflect on their performance through step-by-step tracking and feedback from coordinators recorded on the platform.

2.1.3 Teaching style of each session

The COVID-19 pandemic has significantly transformed traditional coding camps, moving many from onsite formats to online settings. This shift aligns with businesses adopting hybrid and remote work models, making it essential for students to develop relevant skills in these areas. Coding camps, as flexible learning environments, can now be conducted in various formats, including online, onsite, and hybrid setups. All three delivery formats can be equally effective, provided that the instructional design, course content, and engagement activities are thoughtfully planned to leverage the strengths of both physical and virtual delivery methods (Porras and Khakurel, 2021). In particular, the hybrid format emerges as an excellent option for combining the advantages of online and onsite formats: in-person participation offers immediacy and social presence, while various communication and engagement activities can extend these advantages to the participants online.

The teaching style of online and hybrid coding camps addresses several challenges that are typical of these formats. The **online** delivery method often results in poor communication (Herbsleb and Moitra, 2001), a sense of isolation among participants (Mooney and Becker, 2021), reduced engagement (Powell et al., 2021), and fatigue from extended computer use (Yousof et al., 2021). Moreover, facilitators need to replicate the dynamic and interactive nature of **onsite** learning that dedicates little time to explaining fundamental principles in favor of example-centric and copy-paste programming (Fronza et al., 2022). In **hybrid** coding camps, it is essential to create an effective and engaging experience for all participants to minimize inequality between those who are physically present and those who are participating online (Fronza et al., 2024).

To effectively choose the most suitable teaching format - online, hybrid, or onsite - it is essential to carefully evaluate the key characteristics and objectives of each part of a coding camp. In the **first part**, holding the first session onsite helps enhance interaction and collaboration, fostering a sense of community, and establishing clear objectives from the beginning. If some participants are unable to attend the classroom in person, it is essential to implement ad-hoc strategies that ensure the same goals are achieved, even in a hybrid setting. Then, an online or hybrid teaching format can be effective for explaining general concepts related to problem understanding and problem-solving. If the session involves addressing questions or conducting guided exercises, a synchronous format is recommended. Otherwise, if the content is mainly expository, an asynchronous approach using pre-recorded videos and discussion forums offers greater flexibility for participants.

In the **second part**, the introduction of fundamental programming concepts and development tools should preferably be introduced in an onsite or hybrid format. The same applies to the **third part**, where participants work in teams, and apply the acquired knowledge and skills to develop a project. These activities require hands-on experimentation with tools and code, which benefits from real-time guidance from instructors or tutors. A hybrid approach enables in-class participants to receive direct assistance, which they can then share with team members participating remotely. To ensure effective communication and real-time decision-making, it is essential to support virtual collaboration using appropriate tools (such as GitHub, Trello, or Google Docs). Hybrid delivery models benefit from clearly defined online interaction protocols. Theoretical frameworks on hybrid collaboration emphasize the importance of pre-defined communication norms, role rotation among remote and onsite members, and explicit guidance on camera and audio engagement to sustain social presence and group cohesion. Table 2 summarizes the suggested delivery formats for each activity in the three parts of a coding camp.

Table 2. Suggested teaching style for each part of a coding camp.

Part of the coding camp	Focus	In-person	Online synchronous	Online asynchronous	Hybrid
First	Team bonding				
	General concepts about problem understanding and problem- solving			if expository	
Second	Fundamental programming concepts and the development tools				

Third	Project work in teams				
	Presentation				

OSCAR findings
In OSCAR, it has proven effective to hold an onsite first session focused on team bonding, goal alignment, tools, and the code of conduct. Following this initial session, subsequent meetings can allow for hybrid participation from teams. It is beneficial to organize teams according to a limited set of predefined patterns, which enable the rotation of onsite and remote roles across different days. These patterns should be communicated to staff and participants in advance. Each team should have a dedicated online space (e.g., a breakout room) that remains stable throughout the coding camp to support seamless collaboration whether members are onsite or remote.
To bridge the gap between onsite and online participants, it is important to clearly outline expectations regarding camera use and digital etiquette in the invitation and revisit these guidelines at the beginning of the camp. Activities in a show-and-tell format or short games that encourage participants to share something visually can help make using cameras feel more natural and enhance social presence.
It is suggested to have a facilitator dedicated to online participants in addition to the facilitator in the physical room. The online facilitator monitors breakout rooms, manages technical issues, and mediates between the physical classroom and the online environment.

In the OSCAR platform (first release)
Collaboration between remote and on-site participants is supported by allowing them to monitor their progress through different project phases, thereby increasing their engagement across various teaching formats (on-site, hybrid, online).

2.1.4 Facilitation strategies

Strategies for facilitating a coding camp should be identified to promote the skills emphasized by the coding camp. This section outlines a set of strategies for enhancing the cross-cutting skills fostered by the OSCAR coding camps (Figure 4).

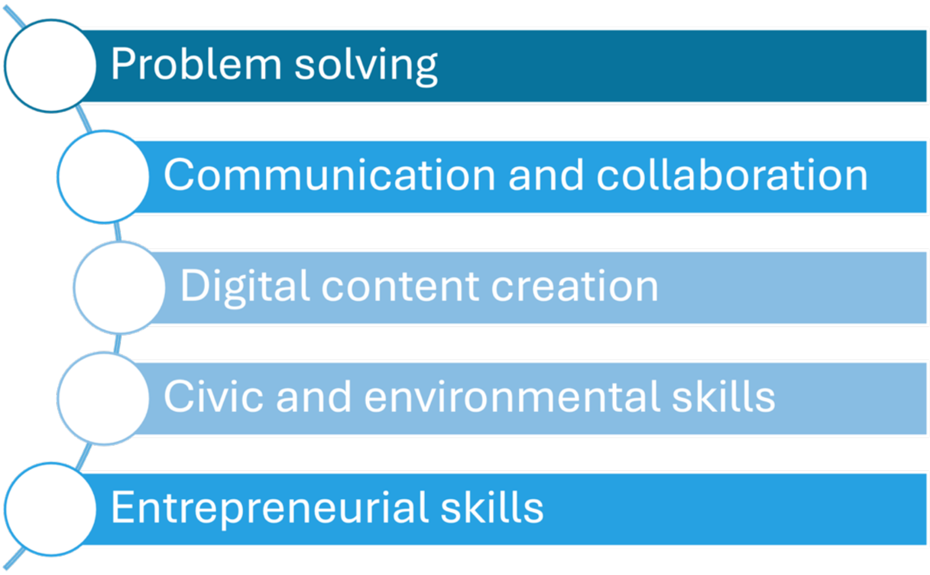


Figure 4. Cross-cutting skills promoted by the OSCAR coding camps, as outlined in Chapter 1.

2.1.4.1 Supporting online/hybrid collaboration

The shift to online formats raises challenges, including a lack of belonging and fatigue due to prolonged computer use. Vicinity of participants, face-to-face interactions, and fun are key characteristics that help participants advance their technical work, share best practices, and grow individually and collectively in expertise (Porras et al., 2019). The hybrid format recently emerged as an excellent solution for addressing these challenges by leveraging the advantages of online and onsite formats. Moreover, the hybrid setting features an authentic experience during the coding camp. Indeed, many companies transitioned toward hybrid work as a "new normal" way of working after the COVID-19 pandemic. Finally, working in remote and hybrid teams improves several dimensions of communication and collaboration skills, including interacting, sharing, and collaborating through digital technologies. A systematic review by Happonen et al. (2021) revealed limited literature on strategies for online or hybrid events, underscoring the need for further research and innovation. In 2024, the analysis of 11 editions of the same coding camp, delivered in various formats (onsite, online, hybrid), provided a set of recommendations for facilitators based on insights gained during these editions (Fronza and Corral, 2024).

2.1.4.2 Solve the problem first, then write the code

It is essential to emphasize that creating a human solution is fundamental for producing a good software solution. By embracing the philosophy of "solving the problem first, then writing the code", students learn the importance of developing their skills as problem solvers and designers of solutions. Once a solution is established, they can then prototype it using a software development tool. This approach directly contributes to enhancing their problem-solving skills.

2.1.4.3 Games and fun activities

Games and fun play a vital role in the success of coding camps, regardless of their delivery format—onsite, online, or hybrid (Fronza et al., 2022). Games and engaging activities can serve as active breaks, encouraging off-screen interactions to alleviate digital fatigue. Furthermore, facilitators can use games to deliver key learning beyond the technical aspects (Fronza et al., 2020; Fronza et al., 2022) while simultaneously increasing engagement, emotion, and motivation (Sorathia and Servidio, 2012). Games help develop fundamental soft skills and professional awareness, such as communication, collaboration, empathy, resourcefulness, listening skills, and inclusion. Moreover, a joyful and comfortable environment also promotes creativity (Boos, 1971), which is a fundamental component of entrepreneurial skills. Finally, well-chosen games can energize the class and deliver key messages. A brief discussion after the game helps to share the lessons learned during the game and sets the stage to continue the learning journey with new topics (Fronza and Corral, 2024). Engagement and participation in the coding camp can be improved through gamification by incorporating elements like reward systems, exciting challenges, and competitive leaderboards. Research on motivational design in game-based learning suggests that providing thematic freedom and narrative ownership increases learners' sense of autonomy and intrinsic motivation. Aligning game topics with socially meaningful issues—such as sustainability or digital citizenship—enhances engagement and perceived relevance, without compromising the coding focus.

Learner engagement in coding camps depends on both pedagogical and organizational variables. From a pedagogical perspective, Self-Determination Theory highlights that autonomy, competence, and relatedness sustain intrinsic motivation (Ryan et al, 2000). Providing learners with meaningful choice—e.g., selecting project themes or rotating creative roles—enhances ownership and persistence. From an organizational design perspective, early communication of objectives, realistic pacing, and avoiding periods of academic overload support focus and energy. Effective facilitation therefore combines motivational design with logistical awareness, keeping participants cognitively and emotionally invested throughout the coding camp.

Playful activities are conceived as structured opportunities to work on transversal skills, not as "extra" or purely recreational moments (Fronza et al., 2022). For example, short games can be designed as "skill containers" that intentionally target competences such as getting-to-know-you and personal motivation, rapid prototyping and resource management, hybrid collaboration, and visual or verbal communication. For each activity, the facilitators can identify observable indicators (e.g., how teams organize tasks, how they communicate, how they handle time constraints) and map them to specific skills within the OSCAR framework. These observations can later be translated into badges or qualitative feedback, turning games into high-impact learning moments that feed directly into formative assessment.

2.1.4.4 Teamwork

Through teamwork, participants engage in a creative and practical process that is rarely done alone: software development is typically carried out by teams working closely together across or within different development phases. In a coding camp where participants collaborate in teams, they get exposure to working with other participants of different backgrounds, priorities, perspectives, and work approaches. The experience of remote and hybrid work also gives a taste of contemporary professional environments. Moreover, working in remote and hybrid teams contributes directly to improving several dimensions of communication and collaboration skills, including interacting, sharing, and collaborating through digital technologies. Consequently, encouraging communication and collaboration promotes innovation (Doğru 2021) as part of entrepreneurial skills. To ease communication and collaboration in online/hybrid teams, facilitators should encourage camera usage and explain why they are doing so at the beginning at the coding camp (Castelli and Sarvary, 2021) so that participants can prepare to be comfortable in front of the camera (e.g., room, clothes, etc.). Moreover, show-and-tell games should be used to encourage and motivate camera usage (Fronza et al., 2022). Section 1.6 provides a list of strategies for the creation of teams.

OSCAR findings
In OSCAR, teamwork has proven to be an essential strategy. Depending on the focus of the coding camp, role differentiation in teams may foster creative ownership, build interdependence, and leverage individual strengths. The physical layout of the room and the configuration of digital tools play a crucial role in supporting teamwork (see Section 2.1.5.3) (Fisher 2021, Zyda 2025).

2.1.4.5 Peer tutors

Peer tutoring consists in having "people from similar social backgrounds who are not professional teachers helping each other to learn and learning themselves by teaching." This definition emphasizes the potential of peer tutoring to support both tutees and tutors (Altintas et al., 2016; Topping, 1996). For tutors, activities like explaining, rephrasing, and questioning provide many opportunities to engage in reflective knowledge-building (Roscoe and Chi, 2007). Tutees achieve the greatest learning effectiveness when interacting with a reciprocal (Chen et al., 2020). Moreover, peer tutoring positively impacts motivation, studying behavior, and exam results (Hardt et al., 2022). Peer tutoring promotes problem solving (Schleyer et al., 2005), collaboration, and reduces knowledge gaps, while increasing motivation, trust, and communication among peers (Fronza et al., 2021). Furthermore, peer tutoring in a coding camp can create a seed effect in participants, enhance their motivation to learn, and ensure the camp's sustainability by continuously training student tutors (Fronza et al., 2021).

OSCAR findings
In OSCAR coding camps, peer tutors play a crucial role in enhancing the learning experience and increasing the scalability of the camps. It is recommended to select tutors from students who have either previously attended a similar coding camp or have demonstrated a strong mastery of the relevant tools and concepts. Selection criteria should include technical competence, communication skills, and a collaborative attitude. Before the camp begins, peer tutors should undergo a brief onboarding process that clarifies their roles, responsibilities, and expectations, as well as provides training on the digital tools and platforms that will be used. During the camp, tutors serve as the first line of support for teams, assisting peers with debugging, interpreting instructions, and navigating the platform. They also observe behaviors related to the skill framework and contribute to the learning assessment.

2.1.4.6 Create a space for experimentation

To foster innovation (i.e., entrepreneurial skills), participants should work in an environment that encourages experimentation and a culture of continuous improvement, where they can experiment without fear of judgment and view failures as opportunities to learn and progress (Doğru 2021). To this end, Agile methods are an effective choice, as they support a work that is opportunistic, incremental, and by trial-and-error phases (Burnett and Myers, 2014; Fronza et al., 2020).

A safe space for experimentation should be made explicit through a code of conduct and through concrete interaction patterns. Participants need to know that mistakes, incomplete solutions, and failed attempts are treated as normal and valuable parts of the learning process.

OSCAR Findings

OSCAR coding camps focused on app development adopt an agile development approach to structure both learning and production of the apps. Specifically, Agile methods are used through the following principles:

- **Incremental development**: apps are built step by step, adding and refining features over time rather than aiming immediately for a complete product.
- **Trial-and-error**: participants are encouraged to test ideas quickly, observe what works or fails, and adjust their solutions accordingly.
- **Frequent releases**: teams regularly present intermediate versions of their apps ("releases") for feedback, reflection, and re-planning.
- **Continuous feedback**: facilitators and peer tutors provide formative feedback based on frequent releases and presentations.

In practical terms, the coding camp primarily focuses on iterative teamwork for app development. Each iteration concludes with a brief presentation of the current release, followed by a feedback session. Learning goals are adjusted based on the progress observed in these releases, allowing for personalized and inclusive pathways.

2.1.4.7 Promote diversity and inclusion

A coding camp that fosters diverse perspectives, ideas, and experiences creates a valuable environment for promoting innovation (i.e., entrepreneurial skills) (Doğru, 2021). This handbook focuses specifically on strategies for encouraging greater participation from girls and embracing diversity.

2.1.4.8 "We are here to help"

To promote creativity and innovation, coding camp facilitators should act more as a "guide by the side" rather than a "sage on the stage" (Xhomara and Uka, 2023). This approach is supported by the strategy "We are here to help" (Fronza et al., 2020): participants can ask for support by first showing their intermediate product and describing their issue with the solutions they have already tested. This supports the creation of a space for experimentation with a culture of continuous improvement.

> OSCAR Findings
> In the OSCAR coding camps, an effective way to implement the principle of "We are here to help" is to guide students to follow a simple protocol before seeking support: (i) show their current output (code or app), (ii) describe the problem in their own words, and (iii) list what they have already tried.

2.1.4.9 Problem-based learning

An effective strategy to promote entrepreneurship is enabling participants to acquire knowledge through the study and resolution of sample problems (Wahid et al., 2024).

2.1.4.10 Working on socially significant projects

Engaging students in solving real-world problems by creating solutions with a social value serves as a great motivator. Additionally, it fosters positive emotions that promote inclusion and diversity and, consequently, encourages innovation (Doğru, 2021). To enhance entrepreneurial skills, students should create a marketing plan and organize a pitch for their product to communicate technical aspects of their work, like design and implementation decisions. As a result, this strategy will promote social entrepreneurship (Ip 2024). Specifically, asking students to create solutions toward solving an environmental problem will help incorporate environmental and climate considerations into the coding camp, thus fostering civic and environmental skills, particularly in terms of safety.

Table 3 shows the alignment of each teaching strategy with the skills fostered by the OSCAR coding camps.

Table 3. Alignment of teaching strategies with skills fostered by the coding camp.

Strategy	Problem solving	Communication and collaboration	Digital content creation	Entrepreneurial skills	Civic and environmental skills
Supporting online/hybrid collaboration		■			
Solve the problem first, then write the code	■				
Games and fun activities		■		■	
Teamwork		■		■	
Students as teaching partners	■	■	■		
Create a space for experimentation				■	
Promote diversity and inclusion				■	
"We are here to help"				■	
Problem-based learning				■	
Working on socially significant projects – focusing on environmental problems				■	■

As shown in Table 4, the teaching strategies used in the OSCAR coding camps are based on the four P's of creative learning: projects, passion, peers, and play. These strategies align closely with the Constructionist approach to education, which emphasizes the value of learners playfully creating personally meaningful projects in collaboration with peers (Resnick, 2014).

Table 4. Alignment of teaching strategies with the four P's of creative learning (Resnick, 2014).

Strategy	Projects	Passion	Peers	Play
Games and fun activities				
Students as teaching partners				
Problem-based learning				
Working on socially significant projects – focusing on environmental problems				

In the OSCAR platform (first release)

The OSCAR platform supports the above-mentioned facilitation strategies as follows:

- **Games and Fun**: Games are played with the support of the platform, and gamification elements, such as badges, are incorporated.
- **Teamwork**: Participants can use the platform to communicate and collaborate with their team, in a private space.
- **Peer Tutors**: Peer-tutors use the platform to assist the teams they are assigned to. They are granted access to the private spaces of each team to communicate and collaborate effectively. Additionally, peer-tutors complete observation surveys for each team assigned to them within the platform. They can report any critical issues to the facilitators. They have access to private space on the platform to communicate with other peer-tutors and facilitators.
- **Diversity and Inclusion**: The platform supports visibility of diverse backgrounds through customizable participant profiles.
- **"We are here to help"**: Integrated FAQs, messaging, and gamified tracking enable a "we are here to help" culture.

2.1.5 Technology

In online and hybrid coding camps, having a well-structured technological setup is essential to ensure seamless communication and collaboration. In hybrid coding camps, the challenge lies in creating inclusive educational experiences that transcend the limitations of remote and physical attendance while fostering meaningful interaction between both groups of participants. To achieve this, the physical presence of participants needs to be complemented with online tools that facilitate real-time collaboration and active participation. Digital platforms that allow for real-time interaction and collaborative tasks are key to ensuring that all participants feel involved and valued. Support mechanisms, such as dedicated online forums or live Q&A sessions, can help maintain a smooth flow of information and provide additional guidance to online learners.

Given that participants may have limited or no prior experience in software development, the technological infrastructure should be user-friendly, accessible, and support both learning and teamwork. Moreover, tools should be selected with accessibility in mind.

From an instructional-design standpoint, pre-camp technical rehearsals and a contingency plan for platform switching reduce cognitive load and disruption during synchronous sessions, especially in hybrid settings.

OSCAR Findings

In our experience, we found that a coherent technological setup is essential for a smooth hybrid camp experience. Each team should have access to at least one computer with a stable internet connection, and the use of headsets with microphones should be strongly encouraged to reduce background noise and improve the quality of online communication. The core toolset typically includes: a videoconferencing platform that supports plenary sessions and breakout rooms; a browser-based development environment (ideally block-based or low-configuration) to minimize installation and compatibility issues; a learning or collaboration platform for materials, communication, assessment, and badges; and optionally, live interaction tools for polls, quizzes, or quick feedback. Whenever possible, tools should be usable with free accounts and require minimal configuration, in order to reduce the cognitive load associated with technology and allow participants to focus on problem solving and creativity.

For serious game development, extended-reality (XR) environments—3D design, animation, and augmented reality—provide experiential contexts that couple spatial and visual reasoning with iterative making. Serious game and XR development align with constructivist/constructionist approaches by turning abstract computational ideas into tangible artefacts that learners design, test, and refine, strengthening creative problem-solving and embodied cognition.

2.1.5.1 Hardware

To ensure effective participation in the coding camp, all teams in the classroom must have access to at least one laptop or desktop computer that meets the minimum system requirements for the development of tools being used. This requirement also applies to participants attending remotely.

A stable internet connection is essential to support video conferencing, cloud environments, and real-time collaboration. Additionally, headphones with a microphone and webcams are necessary to allow communication and interaction during virtual sessions.

Additional hardware may be required based on the coding camp's focus. For instance, during coding camps focused on extended reality (XR), participants may need VR headsets available for testing immersive experiences. Whenever possible, emulators should be suggested to ensure that all participants, whether attending in person or remotely, have the same learning experience.

The use of optional peripherals, such as external monitors for multitasking, graphic tablets for UI/UX design, and mobile devices for app testing, can enhance the learning experience.

2.1.5.2 Software

Communication tools are essential for creating an interactive and efficient learning environment. These tools include a video conferencing platform, such as Zoom, Microsoft Teams, or Google Meet, which can be integrated with an instant messaging tool, like Slack or Discord, for quick communication. An important feature of the video conferencing platform is the ability to offer both plenary sessions and sessions where different teams can access a designated space for collaboration.

Collaborative work tools should enable participants to edit the same document at the same time (e.g., Google Docs, Notion, Trello), share resources and project files (Google Drive, OneDrive, Dropbox), and effectively brainstorm (e.g., Miro). Using integrated ecosystems (content hub + synchronous meetings + feedback channels), such as the OSCAR platform, streamlines coordination and mirrors real hybrid workflows, improving transparency, iteration speed, and team continuity.

To enhance efficiency and streamline collaboration, solutions that integrate communication and collaboration tools, like Microsoft Teams, should be prioritized to reduce the need for participants to switch between various applications. Moreover, to accommodate participants with little to no digital skills, the coding camp should adopt user-friendly and flexible tools.

As technical training, the selected development tools can determine the success of the learning journey (Fronza and Corral, 2024). To enhance remote participation and facilitate instructors in troubleshooting, it is recommended to avoid local installations of software development tools. Furthermore, pre-configured virtual machines or cloud environments can help eliminate compatibility issues. Block-based programming environments should be prioritized, as they allow for the structure and practice of a programming language, with an approach that is friendly for novices. Also in this case, it is recommended to avoid local installations using web-based solutions such as Thunkable, GitHub Codespaces, and CodeSandbox.

Overall, all software tools should be accessible with a free plan and easy to use. From an instructional design perspective, tool reliability, accessibility, and compliance to privacy legislation must be validated prior to implementation. Pre-camp technical rehearsals and contingency planning are recommended to minimize any disruptions during the sessions.

OSCAR Findings

The core toolset typically includes: a videoconferencing platform that supports plenary sessions and breakout rooms; a browser-based development environment (ideally block-based or low-configuration) to minimize installation and compatibility issues; a learning or collaboration platform for materials, communication, assessment, and badges; and optionally, live interaction tools for polls, quizzes, or quick feedback. Whenever possible, tools should be usable with free accounts and require minimal configuration, in order to reduce the cognitive load associated with technology and allow participants to focus on problem solving and creativity.

2.1.5.3 Space

The physical layout of spaces should align with the goals of the camp sessions. For example, in frontal lectures, rooms should be arranged in a flexible setup without desks to allow for dynamic movement when needed. In contrast, for sessions that involve intensive teamwork, classrooms should be organized with one table per team, ensuring sufficient distance between groups to facilitate easy movement for staff. It is also helpful to designate a specific area for online support staff. This area should be equipped with one or more screens to monitor online participants and breakout rooms.

OSCAR findings

For the first session, OSCAR coding camps implemented a flexible layout featuring movable chairs and no desks. This setup facilitated frontal lectures and provided enough space for participants to move around during ice-breaking activities, games, and group discussions. For the project development sessions, the classrooms were rearranged with one table per team, creating a clear "home base" for their work. Each table had convenient access to electrical outlets for laptops and other devices. The space between tables was intentionally generous to prevent overcrowding and allow staff to move around easily. The room included a dedicated side table with two screens for staff supporting remote participants.

2.1.5.4 Monitoring and data centralization

Organizers should monitor the implementation of coding camps to comprehensively assess learning outcomes. Centralizing all data would simplify analysis and visualization, making reporting activities and assessments more efficient. For OSCAR, this is the fundamental concept behind the "connected coding camps" (Fronza et al., 2025), which support the continuous development of skills from a broader perspective, encourage participants to develop their results further, and become building blocks for planning one's educational activities. The central tool to enable the connected coding camps is a tailored platform that provides options for using coding camp results and skills achieved as a collection of personal portfolios that support individual goals. To this end, the platform could provide options to publish the results of a coding camp for others to review, instructions for further development of the outcomes and the platform could also support the participants to link their results for a portfolio of their own. Moreover, the platform could possibly enable a development of new variations by outcomes of the previously organized coding camps, and it could link companies, educational institutions, and other stakeholders.

Figure 5 summarizes the features of the OSCAR platform. It is important to note that each feature should be based on the input from different coding camps, as represented by the various layers of each feature in the figure. This way, the platform would serve as a strategic hub of connectivity. Moreover, the fact that several coding camps can share the same infrastructure accumulates data about them on the platform. Some features in Figure 3 have a more significant impact than others. Because feedback is essential for any learning, and because being able to tell others what a participant has done is a prerequisite for many other features, we consider essential the features highlighted with a star in Figure 5, i.e., (3) providing a clear visualization of skills and their assessment and (8) providing the ability to share skills or other achievements. Similar features may already be available in modern learning management systems, so if that is the case, we encourage organizers of coding camps to promote their use.

Authentic assessment methods—such as project presentations and peer review—enable learners to demonstrate competence through narrative and creative expression. Embedding reflective components during the design phase fosters metacognitive awareness and strengthens the link between process and outcome evaluation.

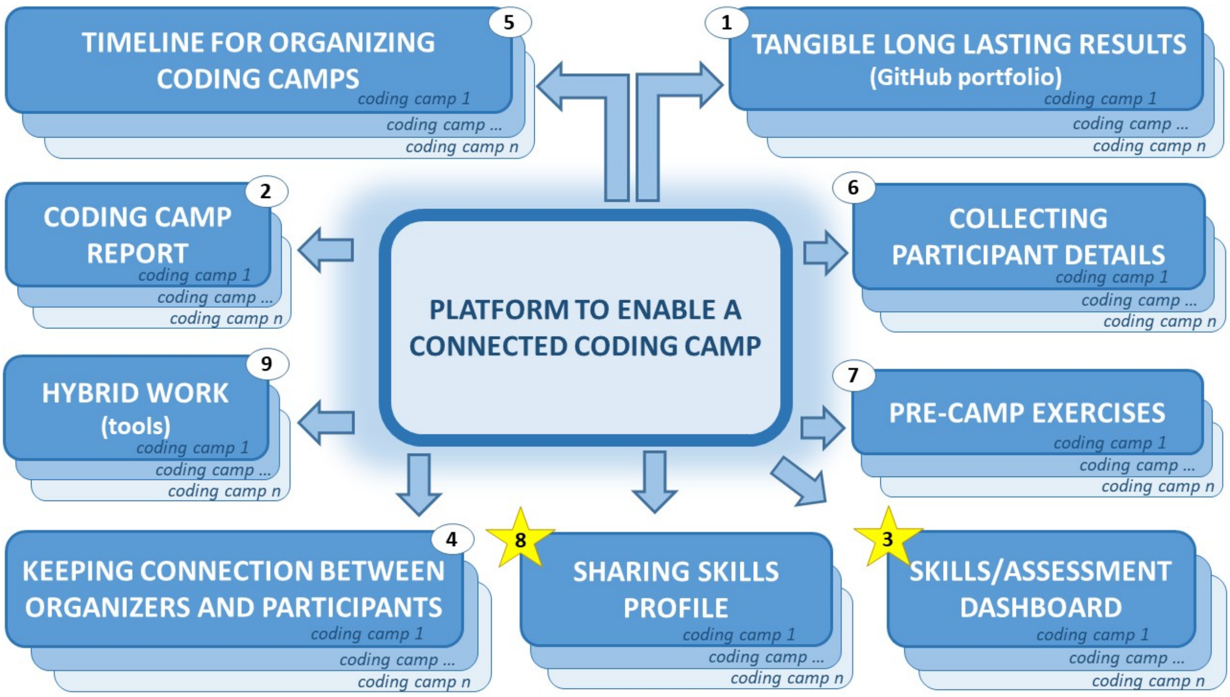


Figure 5. Main features of the platform that enable a connected coding camp. Figure from (Fronza et al., 2025).

In the OSCAR platform (first release)

In the OSCAR platform, access to and visibility of information regarding coding camp content and student information is governed via permissions assigned to users on a per document basis. Information is not accessible to users by default without the required level of permission, determined by the coding camp organizers. The platform may be self-hosted, in which case coding camp organizers are responsible for data security and data retention, and other regulations concerning user data. If the platform is hosted by an external partner, coding camp organizers should make sure the partner adheres to the relevant data regulations.

Centralized data collection captures the skills acquired (visualized in dashboards) and the projects completed (published in Project Results repository). This centralized model supports reflective practice, impact assessment, and continuous improvement of future coding camps.

OSCAR consolidates the technological needs into a single platform:

- **Communication**: Integrated messaging with document comments, dedicated chat rooms and team specific email lists.
- **Collaboration**: Shared document editing, feedback tools, and asynchronous task management.
- **Monitoring**: Progress tracking of project milestones and tasks to manage remote and hybrid settings seamlessly. Leaderboard to track the collected badges.

2.1.6 Strategies for the creation of teams

The formation of teams in coding camps is crucial to promote effective collaboration and embrace diverse perspectives. Research supports instructor-formed teams (Oakley et al., 2004). However, many participants are reluctant to accept that they cannot choose their teammates. To address this issue, Oakley et al. (2004) suggest explaining to them that in a professional environment, they will not be asked who they would like to work with.

For the formation of teams, the following criteria should be followed to promote effective collaboration.

2.1.6.1 Number of members

Even though there is no consensus in the literature on the optimal team size, Oakley et al. 2004) suggest that teams of three or four members are ideal. In two-person teams, there may not be a sufficient variety of ideas and skills, and conflict resolution may be problematic, advantaging the dominant partner. On the other hand, in teams with more than five members, one or more members may be relatively passive.

2.1.6.2 Composition

Heterogeneity is one of the main criteria for team formation. Specifically, ***ability of heterogeneity*** prevents strong students from grouping together and isolating weaker students. Instead, weak students benefit from seeing how good students approach assignments, while strong students may benefit from the experience of tutoring their peers (Oakley et al., 2004). Additionally, ***background heterogeneity*** ensures that team members possess a variety of backgrounds and ability because, for example, they attend different types of schools (Fronza et al., 2022). This encourages teams to share knowledge and support one another, creating an environment of mutual assistance.

Another criterion for team formation is to avoid isolating at-risk minority groups (Gammie and Matson, 2007). This helps prevent these individuals from adopting passive roles or being assigned to such roles, thereby losing many advantages of teamwork (Takeda and Homberg, 2014) and increasing the dropout risk. For example, since women constitute an at-risk minority in Computer Science and STEM in general, Oakley et al. (2004) suggest forming mixed teams that include most women (Fronza and Corral, 2024). Additionally, a student may belong to an at-risk minority because of several other reasons, including gender, socioeconomic status, language, or ethnicity. Team composition should aim for cognitive and demographic diversity, as heterogeneity fosters creativity, distributed problem-solving, and peer scaffolding.

When participant pools are homogeneous, role-rotation or cross-team collaboration can be used to simulate diverse perspectives and maintain balanced participation.

OSCAR findings

The findings from OSCAR indicate that team formation is most effective when it follows a clear procedure aligned with the educational objectives of the camp. Based on our experience, groups consisting of three or four participants provide a good balance between diverse perspectives and the opportunity for each member to actively participate. To create heterogeneous teams, we bring together participants from different schools, study tracks, and levels of experience to promote peer learning and minimize the risk of one student dominating the group's work. Additionally, when forming mixed-gender teams, we aim to include at least two girls whenever possible. We also consider anticipated patterns of remote participation, such as which students will be online on specific days, to ensure that each group can operate effectively within the chosen hybrid format. Clearly communicating the rationale behind team composition to participants has proven valuable, as it helps align expectations and reinforces connections to real world collaborative environments.

2.1.6.3 Hybrid work configuration

In hybrid teams, some members work from home, others from the classroom, and others from a combination of both. In these teams, participants use tools like video calls, shared online workspaces, and collaborative platforms to ensure fluid communication regardless of physical location. A key decision is choosing the hybrid work configuration by determining how many team members will work in the classroom and how many will work remotely.

OSCAR findings

Due to the lack of guidelines in the existing literature, we conducted a study during the OSCAR project to examine how different hybrid work configurations affect the final products developed by teams of three participants during coding camps (Fronza et al., 2024). The study found that having 1 or 2 team members online does not affect the quality of the final product. This indicates that a configuration with one person onsite and two online can be used when physical spaces can only accommodate one-third of the participants in person. In such cases, it is recommended to implement strategies that allow teams to focus more on programming logic while placing less emphasis on the user interface. In another study (Mikkonen et al., 2025) we considered hybrid work configurations for four-person teams, specifically scenarios with two members working online and two in the classroom, as well as three members online and one in the classroom. The results from all configurations were similar. However, when co-located, the participants were inclined to engage in discussions about the internals and behaviors of individual components. In contrast, remote and hybrid setups encouraged participants to divide tasks, with well-defined interfaces between different functions.

When participant pools are relatively homogeneous, role rotation (designer, programmer, tester, storyteller) and cross-team consultations can simulate diverse viewpoints, sustain creativity and balanced participation while preserve the benefits of instructor-formed teams.

In the OSCAR platform (first release)
The participants' information forms and the profile search tool allow organizers to create balanced, heterogeneous teams by considering participants' background, skill levels, and preferences. Participant's user profiles and team member roles enable thoughtful team allocation.

2.1.7 Selection and preparation of peer-tutors

Peer tutors play a key role in hybrid and online coding camps by acting as guides for other participants, facilitating doubt resolution, and promoting collaborative learning (Fronza and Corral, 2024). Their inclusion not only enhances the student experience but also strengthens the sense of community and active participation. To ensure their success, an appropriate selection and training process is essential (Bugaj et al., 2019; Bouthillette 2016; Herinek et al., 2024).

The **selection** of peer tutors should be based on several key criteria to ensure their effectiveness in the coding camp. First, they must demonstrate strong mastery of the content and skills covered in the camp. Additionally, they need to possess strong communication skills that allow them to explain concepts clearly to students with varying levels of experience. A collaborative attitude is also essential; this includes a willingness to help, patience, and the ability to create a positive learning environment. Furthermore, prior experience in coding camps or other mentorship activities is highly desirable, as it can enhance their effectiveness as tutors.

Coding camps have a seed effect by allowing participants to develop the necessary motivation to eventually serve as tutors. During their participation in a coding camp, perspective tutors distinguish in terms of ability to explain their (creative) ideas in multiple ways, enjoyment of teamwork and effort to encourage their peers to strive for the best, curiosity, and the number of questions asked to facilitators and peers to extend their knowledge (Fronza et al., 2021).

Before the start of the coding camp, peer tutors must receive **specific training** to effectively fulfill their role (Fronza et al., 2021). This preparation includes orientation sessions, where they are introduced to their responsibilities, expectations, and the impact of their role within the program. Additionally, they receive training in peer teaching methodologies, equipping them with techniques to foster collaboration, guides without providing direct answers, and encourage critical thinking among participants. They are also trained in the use of digital tools, ensuring proficiency in the platforms used in the coding camp to effectively assist their peers. Finally, they receive instruction in conflict resolution strategies and emotional support, helping them manage recurring questions, promote inclusion, and address potential frustrations among participants.

The **participation** dynamics of peer tutors include various activities that strengthen both learning and the overall experience of participants. Their role involves supporting practical sessions, supervising small groups, and reinforcing concepts in real-time (Fronza and Corral, 2024; Fronza et al., 2022). Additionally, they facilitate consultation spaces, making themselves available in discussion forums, chats, or synchronous meetings to address questions and guide students throughout their learning process. Moreover, peer tutors should participate in regular meetings with the organizers to share experiences, suggest adjustments, and optimize the coding camp's dynamics. This ensures an effective and collaborative learning experience, continuously improving the engagement and success of all participants.

Preparatory sessions should familiarize tutors not only with the technical platform but also with inclusive communication practices and adaptive support strategies derived from differentiated instruction models.

In the OSCAR platform (first release)
Peer tutors are registered within the platform and are assigned relevant access rights.

2.2 Strategies for differentiated teaching and inclusion

An effective strategy to attract more students is to integrate learning with their cultural backgrounds. For K–12 students, keeping them interested and engaged is a major challenge. Many informal programs use robotics or video games, but these can be hard to access and often appeal mostly to students already interested in STEM (Newton 2020), limiting diversity. An alternative is to use cultural elements that reflect students' backgrounds. For example, some programs use music genres like hip-hop to make coding more engaging and relatable (Krug et al 2021). In recent years, a notably inclusive teaching method has been Differentiated Instruction. However, this approach is long-term and often incompatible with the short-term, intensive nature of coding camps. To make Differentiated Instruction effective in this context, camps need digital tools that can quickly assess students' needs and connect them with schools and other informal learning opportunities.

2.3 Inclusion

Accessibility is shaped not only by the platform itself, but also by the user's operating system and assistive technologies (e.g., screen readers, alternative input devices, braille terminals). For this reason, the platform should be implemented in line with established standards and best practices such as WCAG and WAI-ARIA, and it should be validated through both expert review and feedback from real users who rely on accessibility tools. Practical requirements include full screen-reader compatibility, robust keyboard navigation, high-contrast visual options, and accessible multimedia through captions (and, where feasible, sign language interpretation). Inclusivity is further strengthened by multilingual support, optimization for low-bandwidth contexts, and the clear publication of an accessibility statement with channels for requesting accommodations. In this direction, the OSCAR platform contributes to inclusion by enabling personalized goals, visualizing skill development, adapting learning materials, supporting formative feedback, and ensuring equitable participation across onsite and remote settings. In its first release, OSCAR already integrates ARIA elements, keyboard navigation, customizable visual styles (including high-contrast themes), multilingual content through document translations and video subtitles, and a localized interface with additional languages planned.

Alongside accessibility, Differentiated Instruction is presented as an inclusive approach that improves learning outcomes by adapting teaching to individual learning profiles, using assessment data to guide decisions and leveraging the role of positive emotions in sustaining engagement. Differentiation requires educators to work across five curricular elements: content (what is learned), process (how learners make sense of ideas), product (how learning is demonstrated), learning environment (the climate and spaces for collaboration and autonomy), and assessment and feedback (formative support that builds motivation and self-efficacy). In intensive and hybrid camps, this translates into deliberate design choices: negotiated individual or team goals, frequent project releases, challenges offered at different levels of complexity, exercises available in multiple variants (from guided to open-ended), and feedback embedded in the rhythm of the camp through prototypes, reflection moments, and short written or oral comments. Digital support becomes essential to track skills over time, monitor progress in real time against planned trajectories, document learning journeys, and provide tools for fast communication and material sharing. In this perspective, a dedicated platform can reduce friction compared to disconnected generic tools and can operationalize differentiation through integrated features such as ePortfolios, product review and assessment tools, skills dashboards and growth charts, collaboration spaces, and—where appropriate—AI-based supports for ideation and project design.

2.3.1 Accessibility

Modern accessibility features, like screen readers, are usually provided by the operating system or additional software installed on the user's device. Support for accessibility peripherals, such as braille terminals, is also dependent on the operating system. The platform can support accessibility mainly by adhering to the WCAG and WAI-ARIA standards and recommended practices (see https://www.w3.org/WAI/standards-guidelines/wcag/ and https://www.w3.org/WAI/standards-guidelines/aria/ respectively) in its implementation. Modern popular web frameworks largely implement these. Nevertheless, the platform's accessibility should be reviewed, preferably, both by an accessibility expert and genuine users of accessibility features or tools.

To ensure that all participants can fully engage with the coding camp activities, the platform must be designed with accessibility at its core. This includes providing full compatibility with screen readers, offering keyboard navigation for users who cannot use a mouse, and ensuring high-contrast visual options for participants with visual impairments. All video content should include captions and, when possible, sign language interpretation. Additionally, the platform should support multiple languages and be optimized for low-bandwidth connections, enabling access from diverse geographic and economic backgrounds. An accessibility statement should be clearly available, along with channels for participants to request additional accommodation as needed. By prioritizing inclusivity, the platform will empower every learner to participate fully, regardless of their abilities or circumstances.

Accessibility and improved inclusion are core pillars of effective coding camps.
OSCAR directly supports these goals by:

- **Personalized Goal Setting**: Through participant profiles and project tracking, individuals can define personal learning objectives.

- **Skill Visualization**: OSCAR's Dashboards allow students to see their skills grow, helping facilitators adapt to teaching strategies.
- **Adaptive Materials**: Material Documents can be differentiated for varying levels of difficulty or learning preferences, supporting flexible content delivery.
- **Formative Feedback**: Exercise Documents allow real-time and asynchronous feedback on tasks, supporting individualized learning paths.
- **Accessible Technologies**: The platform supports remote participation equally with onsite, ensuring equitable access for all learners.
- **Diverse Content**: Organizers can publish socially meaningful projects, integrate interdisciplinary challenges, and tailor sessions to broader interests using the Project Result spaces.
- **Cultural Adaptation**: Material templates and session examples can be adapted to different cultural backgrounds, ensuring broader relevance and engagement.

By offering these functionalities, OSCAR enables a dynamic, inclusive, and adaptive learning environment that fosters differentiated instruction and empowers every participant to thrive.

In the OSCAR Platform (first release)

Accessibility features in the platform are supported by ARIA attributes/elements inserted into page content. This enables accessibility tools, like screen readers, to read the page correctly. The platform also supports keyboard navigation. Visual styles can be customized via global or user-specific stylesheets or themes, supporting high contrast settings for visually impaired users.

Multilingual content is supported via document translations for the documents themselves as well as for embedded video content by supporting video subtitles for different languages. The platform's user interface has localization support: currently supported languages are English, Finnish and Swedish. Supported languages will be extended with Italian and Spanish during the project, or as needed.

Differentiated teaching and inclusion are mostly dependent on the coding camp design. The platform facilitates these by providing a large collection of different task templates, as well as tools such as progress trackers; user profiles; feedback, visualization and assessment instruments; and so on.

2.3.1.1 How to create accessible materials

Creating accessible learning materials starts with treating accessibility as a design goal from the very beginning, not as a final "compliance pass." In practice, this means planning how learners will perceive, navigate, and interact with each activity through multiple modalities, and checking early whether the intended learning outcomes remain achievable when visual, auditory, motor, or cognitive demands change. Research on teaching "inclusive thinking" in computing suggests that instructors can meaningfully raise accessibility awareness by making accessibility a recurring lens in course content and assignments, rather than an isolated topic (Ludi et al., 2018). Similarly, experiential formats that include structured activities and reflection can help learners understand why accessibility matters and how design choices affect real users (Shi et al., 2020).

A consistent best practice is to design materials using principles aligned with Universal Design for Learning, so that the same concept can be accessed and expressed in more than one way. Work on curriculum modifications highlights the educational value of offering alternatives in how content is represented, how it is delivered, and how learners demonstrate mastery; these adaptations can reduce competing behaviors and support engagement when they are actually present during instruction (Lee et al., 2010). Instructors can operationalize this by ensuring that every key resource has a workable equivalent: visuals are accompanied by meaningful text descriptions; videos have accurate captions (and, when feasible, additional language subtitles); and instructions are written clearly enough to be followed without relying on layout or color cues. When materials include code, diagrams, or UI elements, the text should explicitly state what learners need to notice and why it matters, so that the learning task does not depend on "seeing" the answer.

For learners with visual impairments, accessible coding materials benefit from deliberate attention to auditory and tactile access. Design-oriented work on teaching coding to children with visual impairments emphasizes the importance of voice-based outputs and meaningful, feedback-rich practice, informed by interviews with learners and teachers and by iterative prototyping of learning materials (Kert et al., 2021). Evidence from tactile teaching strategies in programming education also points to the value of multi-modal activities that help learners build mental models through hands-on experience, supporting progress while acknowledging common challenges such as navigating code and grasping abstract concepts (Alotaibi et al., 2020). Practically, instructors can translate these insights into materials that encourage structured verbal reasoning (for instance, reading code aloud with purposeful pauses), provide stepwise navigation cues for code exploration, and include tactile or manipulable representations where appropriate (for example, physical tokens for control flow or structured worksheets that guide "tracing" without requiring visual diagrams).

Accessibility also improves when instructors collaborate with learners and stakeholders instead of designing isolation. Practitioner-oriented guidance on supporting diverse learners through educational technology underscores that digital media can be a powerful equalizer when it is chosen and used intentionally to widen participation and reduce barriers, rather than to add novelty (Lock & Kingsley, 2007). In higher-education computing contexts, inclusive making research further argues that accessibility-oriented learning experiences are strengthened by critical discussion, community partnerships, and design work that centers disabled perspectives, which can broaden what "counts" as making and who can participate (Worsley & Bar-El, 2020). For instructors, this translates into routinely collecting feedback from learners who use assistive technologies, validating materials with authentic use cases (not only with automated checkers), and iterating on materials after each delivery.

Accessible materials are easier to sustain when they are built as reusable templates rather than one-off artifacts. Instructors can maintain a small set of "accessible defaults" for documents and slides, such as consistent heading structure, descriptive link text, readable typography, and predictable navigation patterns. Over time, these defaults reduce the workload of making each new resource accessible and allow instructors to focus on refining the learning experience itself. The overall goal is not merely to remove obstacles, but to create learning materials that support engagement, agency, and successful participation for all learners while preserving the integrity of the learning objectives (Lee et al., 2010; Ludi et al., 2018).

2.3.2 Differentiated Instruction

Differentiated Instruction is an inclusive teaching approach that aims to develop students' skills by adapting the learning process to individual needs. It uses assessment data to tailor instruction to each student's learning profile. Case studies show that this method enhances learning outcomes across all education levels (J. Hattie, 2023) by encouraging growth in the skills of both the students in the classroom (social skills, teamwork) and students working individually online (digital skills, hybrid-work). The process becomes even more effective when positive emotions are actively involved (J.

D. Mayer and P. Salovey 1993). The core of differentiation is the ability of teachers to adapt to five curricular elements (G. Iaccarino et al. 2025):

- *Content:* what students need to learn. To differentiate the content, it is necessary to provide a collection of similar or previous teaching activities. This ensures the application of diverse ideas and learning strategies while enabling the effective utilization of the student's working style.
- *Process:* how students make sense of ideas and information. Teachers can differentiate the process by finding out where their students are in their learning processes and adaptively provide support (N. Sellier, P. An. 2020). For this purpose, an overview of the students' progress is needed. This requires digital support to track skills over time, not only during the coding camp but also in previous experiences, both conventional and non-conventional. In addition, a tool should check, in real time, the group (or individual) development process against the instructor's proactive planning. This allows *"to make small adjustments"* (C. A. Tomlinson 2017).
- *Product:* how students demonstrate their learning (C. A. Tomlinson 2017). The teacher should not force the target of the final product for the students but allow each student or team to create something different and unique. To do this, both the teachers and the students need the products developed in the previous camps and the continuous monitoring of the skills.
- *Learning environment:* the "climate" of the classroom (N. Sellier and P. An. 2020). A differentiated classroom respects the diversity of learners, provides a safe space for them to express their needs, and encourages both accountability and autonomy. It should also support individual work as well as collaboration. In face-to-face, online or hybrid coding camps, the learning environment also means virtual environments in which to communicate and share digital materials effectively and quickly.
- *Assessment and feedback:* an analysis of the skills acquired by the student followed by formative feedback during the learning process (S. S. Fahim and R. MR Khalil 2015). This approach fosters problem solving, skills acquisition (R. Vollmeyer and F. Rheinberg 2005) and enables differentiated teaching by increasing motivation and a sense of self-efficacy.

These five curricular elements can be greatly improved with a platform that offers personalized services. To meet different learning needs, teachers should provide varied or previous materials. This allows students to try different strategies and learn in ways that fit their needs. Digital tools are needed to track students' skill development over time, not just during the coding camp, but also through past formal and informal experiences. The platform should also monitor individual or group progress in real time and align it with the teacher's plans. This allows for timely adjustments. Learning outcomes often result from several connected activities, so both teachers and students need clear documentation and tools to follow the learning journey, reflect past steps, and plan. The environment should support both solo and group work. In-person, online, or hybrid camps also need virtual spaces for fast communication and sharing of materials. Finally, evaluation tools should track each student's growth, offer useful feedback, and allow for personalized assessment.

Implementing Differentiated Instruction in an intensive and hybrid camp requires deliberate design choices. One effective strategy is to work with personalized or team-specific goals, negotiated on the basis of frequent project releases. As teams progress at different paces, facilitators can propose additional or alternative challenges with varying levels of complexity, allowing each group to work at an appropriate level of difficulty without fragmenting the overall trajectory of the camp. Exercises can be offered in multiple variants, from more guided to more open-ended, so that teams can select tasks that match their confidence and skills. Formative feedback should be embedded in the camp rhythm: intermediate prototype presentations, structured reflection moments, and short written or oral comments provide regular opportunities to discuss both the product and the learning process. At the level of the learning environment, technologies and tools should be chosen for low entry thresholds and equitable access, while content and challenges are progressively tuned to maintain engagement in the presence of heterogeneous backgrounds and prior experience.

2.3.2.1 Digital tool to support Differentiated Instruction

The use of a dedicated platform serves as an essential tool for implementing Differentiated Instruction within short-term learning activities (G. Iaccarino et al. 2025). When used effectively, technology can enhance student motivation, facilitate teamwork, and improve the overall learning environment. Even a decade ago, early studies indicated that tools like ambient displays could provide teachers with real-time insights, allowing them to better tailor their lessons (E. Van Alphen, S. Bakker 2016). During a coding camp, participants can connect and collaborate with others from different places, even remotely. Digital tools make it easy to chat, share work, and interact with each other. In (S. Asim et al. 2020), authors gave tips on how to plan, and use Differentiated Instruction in online settings, the forerunners of today's hybrid classrooms. So far, teachers have mostly used generic tools, which are often clunky and not well connected. A dedicated platform makes things easier and faster, offering better support for short-term teaching like coding camps.

Some of the tools already support Differentiated Instruction, while others are custom-built for this purpose.

ePortfolio. It is a digital version of a student portfolio that helps keep students engaged and supports assessment. It lets them collect their work, reflect on it, and build self-awareness and a personal brand (S. Asim et al., 2020). Adding an ePortfolio to the platform, building on the *"tangible, lasting results"* feature, would let students save their projects, documentation, and learning materials. It also supports differentiated content and learning paths.

Product review tool. It could help with assessment by giving automatic, data-based feedback. It supports more personalized evaluation and helps differentiate both assessment and feedback.

Skills dashboard. A skills equalizer to better plan camp goals based on students' needs, especially to strengthen weaker skills. This allows for real-time tracking and quick adjustments to activities. Skill equalizers help tailor both the learning process and outcomes.

Skills growth chart. It helps track progress over time, making it easier to assess both short- and long-term learning. It's a feedback tool that supports more personalized assessment and feedback.

AI-powered tool. It could help students brainstorm and design their projects, supporting more personalized and creative outcomes. It contributes to differentiating the product.

Collaboration tool. To support different learning styles, it's important to encourage both collaboration and independent work, whether in a virtual classroom or a hybrid setup (S. Asim et al., 2020). That's where digital tools come in. Easy-to-use software for real-time chat, file sharing, and teamwork is key. These tools should also create space for reflection, exploration, and critical thinking. Including them in the platform helps differentiate the learning environment.

2.3.3 Teaching materials to enhance inclusion

Differentiated Instruction aims to meet the different needs, preferences and learning abilities of students. This method requires the use of a wide variety of tools, resources, and strategies to create flexible learning environments where all students can succeed. Among the many techniques employed, one particularly promising and increasingly recognized method is the integration of supplementary material such as mind maps, videos, gamified activities, interactive resources, and video games into the educational process.

For example, video games offer dynamic, interactive, and engaging platforms that can be tailored to individual learning profiles. Educational video games can provide immediate feedback, clear goals, and a structured yet stimulating environment that helps these students remain engaged and better manage their learning experience.

Moreover, video games can foster the development of important cognitive and executive functioning skills, such as problem-solving, working memory, and sustained attention. By integrating such tools thoughtfully and purposefully into differentiated instruction, educators can create more inclusive and supportive classrooms that promote success for all students, particularly those with special educational needs. In the following, the key aspects identified in the literature review and the proposed solutions for integrating these criteria into the created games:

Motivational Aspects. Positive reinforcement is essential for fostering motivation and progress awareness in learners (J. García-Guerrero and JM. Calleros 2021, N. Echeverry Chaves 2015). To minimize frustration, it should be audiovisual and focused on achievements rather than errors (J. García-Guerrero and JM. Calleros 2021). Accordingly, the game is designed to display only successful outcomes, accompanied by encouraging audiovisual feedback (e.g., “Wow”, “Congratulations”) upon task completion.

Playful activities. The attention, coordination, and planning tasks to be performed in video games are approached as games to avoid monotony. In the proposed games, the interval between stimuli the child receives will not be excessively long, so that performance is not affected (C. Tye et al. 2016).

Minimizing Excessive Audio-Visual Stimuli. To support attention, only success or failure sounds will be used, with higher tones for success. Background music will be subtle and non-intrusive. Primary colors will be employed in clean, distraction-free environments. One game will feature a simple, non-animated avatar, while the others will use a first-person perspective without avatars (N. Echeverry Chaves 2015, J. García-Guerrero and JM. Calleros 2021).

Level of Difficulty. Game instructions must be clear and concrete (N. Echeverry Chaves 2016), with a demo and practice option provided. Difficulty levels should be manageable (J. García-Guerrero and JM. Calleros 2021), with each game offering three levels: easy, medium, and hard. Difficulty is determined by time constraints or increased distractions (S. Bioulac et al., 2014). Texts will use the Lexia Readable-Regular font, designed for individuals with dyslexia, a common comorbidity in ADHD (N. Echeverry Chaves 2016).

Time Spent Playing. Research on video game interventions for children with ADHD highlights the importance of managing playtime. Excessive exposure can lead to irritability, anxiety, and inattention (CA. García-Ríos and VE. García-Ríos 2020). Studies recommend limiting play to no more than 30 minutes, with some suggesting even shorter sessions of around 10 minutes (P. Garcia-Redondo et al. 2019, V. Benzing and M. Schmidt 2019, S. Cortese et al. 2015). Accordingly, the proposed video games will last approximately 8 to 9 minutes, including the preparation phase.

Immediate Evaluation. Immediate feedback allows players to monitor their progress and identify mistakes. In the proposed video games, results will be displayed at the end of each difficulty level (P. Garcia-Redondo et al., 2019).

Reward System. Positive musical stimuli will play during gameplay as children achieve goals (CS. González et al. 2018). Upon game completion, visual feedback such as “Congratulations!” or “Wow!” will be displayed, accompanied by a musical cue. Points are awarded for successes, with no deductions for failures, in line with positive reinforcement principles. After each game, the earned points will be shown. Additionally, at the end of each game, children can participate in a creative reward game to further enhance engagement.

2.4 Strategies to attract more girls

Coding camps, as short-term, immersive educational experiences, have the potential to counteract these barriers by fostering an inclusive learning culture that attracts and retains female participants. One of the most effective strategies for achieving this goal is the reframing of key competencies and learning goals to align with diverse interests and learning preferences, thereby broadening the appeal of coding education.

Traditionally, coding curricula have been framed within a narrow scope, emphasizing algorithmic efficiency, computational rigor, and competitive problem-solving. While these aspects are critical, research suggests that women and other underrepresented groups are more likely to be engaged when coding is presented in a broader, application-driven context (Cheryan et al., 2015). By integrating interdisciplinary projects—such as coding for social good, digital storytelling, or user experience design—coding camps can create an environment where problem-solving is linked to real-world impact, thereby appealing to a wider range of learners (Master et al., 2016).

Furthermore, the adoption of collaborative, mentorship-based pedagogies has been shown to enhance the participation and persistence of women in STEM (Dasgupta & Stout, 2014). Pair programming, peer-led instruction, and exposure to female role models in tech can significantly reduce the intimidation factor often associated with coding, creating a psychologically safe space for learning. Additionally, designing curricula with flexible learning pathways allows participants to build confidence through personalized engagement rather than rigid, one-size-fits-all instruction.

Shifting towards human-centered, diverse, and interdisciplinary learning objectives does not dilute the rigor of coding education; rather, it enriches it by cultivating a more diverse talent pool, fostering creativity, and ensuring broader societal engagement with computational skills (Barker & Aspray, 2006). This transformation is essential for closing the gender gap in computing and establishing a more equitable and innovative tech industry.

2.4.1 Cultural barriers and stereotypes

Several cultural barriers play a significant role in deterring girls from engaging in computing and game development camps, as they reinforce gendered perceptions of competence and belonging in STEM fields (Cheryan et al., 2015).

One of the most common cultural barriers is the stereotype that computer science is inherently "male" field. Successful programmers are often described as socially isolated, highly intelligent, and overwhelmingly male, which can create a sense of alienation among female participants (Master et al., 2016). These portrayals contribute to the widely studied phenomenon of stereotype threat, where individuals who are aware of negative stereotypes about their group's abilities may experience anxiety and reduced performance as a result (Spencer, Steele, & Quinn, 1999). In the context of coding camps, stereotype threats can manifest as lower self-confidence, reluctance to participate in competitive coding challenges, or withdrawal from activities that reinforce the perception of exclusivity (Shapiro & Williams, 2012).

The perception that success in computing requires an innate ability rather than effort is another cultural hurdle that disproportionately affects girls. Research suggests that fields emphasizing "brilliance" over hard work and persistence tend to have lower female representation (Leslie et al., 2015). Shifting the focus towards a growth mindset—where skills are developed through effort and learning—has been shown to improve retention and confidence among underrepresented groups in STEM (Dweck, 2006).

Socialization patterns also reinforce gendered career expectations from an early age, further exacerbating cultural barriers. Studies show that boys are more likely to be encouraged to engage with computers, video games, and problem-solving activities, while girls receive less exposure and encouragement in these areas (Wang & Degol, 2017). The cumulative effect of this early gendering of interests results in fewer girls developing a sense of belonging in computing environments, making them less likely to enroll in coding camps or pursue computing careers (Eccles, 2011).

To counteract these barriers, interventions must address the deeply ingrained cultural narratives that shape perceptions of computing and gaming. Creating learning environments that emphasize collaboration over competition can mitigate stereotype threats and foster a sense of belonging (Dasgupta, 2011). Providing female role models and mentors in computing education can also help challenge existing stereotypes and demonstrate diverse pathways to success in the field (Cheryan, Siy, Vichayapai, Drury, & Kim, 2011). Furthermore, educational initiatives that highlight the societal impact of computing (such as its applications in healthcare, sustainability, and education) can make the field more appealing to a diverse range of students (Diekman et al., 2010).

Incorporating visible female role models—such as tech entrepreneurs or game designers—into camps and promotional materials helps challenge stereotypes and provide inspiration. Mentorship programs and peer-led instruction further reinforce belonging and motivation.

Addressing cultural barriers in computing education requires a multi-pronged approach that involves educators, policymakers, parents, and industry leaders. By reshaping narratives about who belongs in computing and creating inclusive educational environments, it is possible to dismantle the cultural obstacles that prevent girls from fully engaging in coding camps initiatives.

Non-conventional learning experiences (such as hackathons and coding camps) are used to promote inclusion and attract more girls to STEM (Iaccarino et al., 2024). However, a current lack of operational tools hinders the practitioners' efforts to have a systematic approach to make these learning experiences more inclusive and appealing to girls and prevents them from having the means to appraise the effectiveness of their initiatives in this regard.

2.4.2 Inclusion and gender balance

It is widely known that achieving gender balance in education is crucial for creating inclusive educational systems and societies: when inclusion and gender balance are achieved, girls and boys have equal rights and opportunities for education, as well as the power and agency to shape their lives and futures (UNESCO, 2023).

In Figure 6, the transition from blue to green highlights a shift from focusing on inclusivity to gender balance. The middle group focuses on both.

Figure 6. Assessment framework for inclusive and gender balanced coding camps (Iaccarino et al., 2024)

Event strategy. Beginner-friendly activities and low competition are good strategies to attract more women, who often feel insecure about their skills and knowledge (C. Ferraz and K. Gama, 2019). Moreover, they are interested in socially significant projects and look for recruitment opportunities.

Organizational recommendations. Ensuring diversity within the organizing team creates a female-friendly environment (G.T. Richard et al., 2015). Moreover, team-building activities allow teams to overcome differences and fears, and girls feel more motivated and willing to participate. A code of conduct guarantees tolerance in words, actions, and communication. Finally, it is crucial to have a healthy working pace and diet (e.g., avoiding sleepless nights).

Event promotion and information sharing. Femvertising (i.e., advertising campaigns aimed at women), accurate wording, and neutral graphic design in advertising can boost women's participation. Providing a detailed task description can help potential participants understand their role within a team, while publishing statistical data from previous editions can attract more women.

2.4.3 Event strategy for girls' attraction

Girls' interest in computing drops early during primary and secondary education, with minimal recovery in later education stages (i.e., University or professional courses) (Happe et al., 2020). The gender diversity at the coding camp currently aligns with the trend in STEM roles in the European Union (from 22% to 46% in 2021) (European Commission, 2022).

Coding camps have been recognized as environments that broaden participation in computing and engage end-users. Participants have reported these camps to be more open and inclusive, creating opportunities for diverse groups to collaborate and develop skills. However, despite these positive aspects, research in computing education has identified several barriers that participants might face, such as stereotypes of nerdiness and intelligence. These stereotypes, coupled with the intensive nature of activities, often require significant perseverance and confidence, which can deter underrepresented groups, including girls, from participating. (Lewis et al., 2016).

Diversity within organizing teams has been linked to more female-friendly and inclusive environments (Richard et al., 2015). However, there are multiple obstacles to gender balance in coding camps, including:

- Social Stereotypes: Coding is often seen as a male-dominated field.
- Confidence Gap: Girls are more likely to underestimate their programming abilities (Faenza et al., 2021).
- Lack of Role Models: Few women in tech leadership positions.
- Competitive and Isolated Environments: Many coding events emphasize competition, which can discourage female participation (Kovaleva et al., 2022).

To support girls' participation, coding camps should clearly communicate that no prior expertise is required. The first session provides foundational knowledge and ensures a beginner-friendly environment (Ferraz & Gama, 2019). Reducing competition and promoting teamwork are also proven strategies for encouraging broader participation, particularly among girls who may undervalue their skills (Kovaleva et al., 2022).

Mentorship programs featuring female industry professionals offer an additional layer of support, contributing to increased confidence and motivation (Microsoft & KRC Research, 2018). Moreover, the representation of diverse participants in marketing materials, particularly those that depict girls actively engaging in coding, challenges stereotypes and fosters inclusivity (Master, Cheryan, & Meltzoff, 2016).

Tailoring content to appeal to diverse interests is another significant factor in increasing female participation. Business game development should be framed as an interdisciplinary and problem-solving endeavor rather than solely a technical discipline. Emphasizing storytelling, design thinking, and real-world applications can resonate with a broader range of interests, particularly when framed around real-world applications, social entrepreneurship, or sustainability goals (Kaplan & Flum, 2012). Furthermore, incorporating interdisciplinary projects that merge coding with business strategy, marketing, and user experience design can attract individuals with varied backgrounds and interests (Barker & Cohoon, 2009). Providing flexibility in project themes, including those centered on sustainability, social entrepreneurship, or community-driven business models, can further enhance engagement.

To increase engagement, coding should be framed as an interdisciplinary, application-driven activity—such as through business games, storytelling, or UX design—rather than solely technical problem-solving. Fostering a collaborative learning approach has been shown to be effective in enhancing female participation in coding-related fields. Research suggests that girls often thrive in cooperative learning environments rather than highly competitive settings. Structuring activities to emphasize teamwork, collaboration, and problem-solving, rather than individual performance-based challenges, can be a more effective pedagogical approach (Master, Cheryan, & Meltzoff, 2016). The promotion of mixed-gender team challenges, ensuring equitable group dynamics, can also facilitate a more inclusive experience.

2.4.4 Organizational recommendation to attract more girls

To address these challenges and attract more girls, it is essential to design coding camps that are more engaging and inclusive. This approach not only makes the camps more relevant to a wider audience but also ensures that participants' individual learning goals are addressed.

Strategic outreach and partnerships play a crucial role in expanding recruitment efforts. Collaborations with schools, parent-teacher associations, and organizations dedicated to supporting girls in STEM can facilitate increased enrollment in coding camps (Margolis, Fisher, & Miller, 2000). Engaging parents through informational sessions is equally important, as it helps to dispel misconceptions about coding and business being male-dominated fields and fosters familial support for female participation (Kaplan & Flum, 2012).

Ensuring diversity within the organizing team creates a female-friendly environment (Richard et al., 2015). Team-building activities allow teams to overcome differences and fears, and girls feel more motivated and willing to participate (Kovaleva et al., 2022).

OSCAR findings
Several design choices can help increase the participation and retention of girls in coding camps. First, the event should be clearly framed as beginner-friendly, explicitly stating that no prior coding experience is required and ensuring that all essential notions are introduced from scratch; choosing a block-based or otherwise low-threshold programming environment further supports this message. Second, the overall climate should be "low competition": competitive elements, if any, can be confined to short games or playful challenges rather than to the core project work, which should emphasize collaboration and mutual support. Third, project themes should be socially meaningful and connected to real-world issues (e.g. environment, community well-being, inclusion), highlighting the impact that digital solutions can have. Visual and textual communication should be neutral and free from stereotypes, showing diverse participants in active, competent roles. When forming teams, it is advisable to avoid isolating girls by ensuring, where possible, that mixed-gender groups include at least two female participants, thereby reducing tokenism and supporting a stronger sense of belonging.

2.4.5 Event promotion and information sharing

The vision of connected coding camps emphasizes fostering an inclusive environment by broadening the scope of topics and activities to engage participants with varied interests. By overcoming barriers such as stereotypes and promoting confidence-building experiences, coding camps can encourage more girls to take part, ultimately contributing to a more diverse and balanced participation in computing education.

Promotional strategies should use neutral wording and visuals, highlight participant diversity, and avoid reinforcing gender stereotypes (Kovaleva et al., 2022).

The coding camp promotion prioritizes neutrality (also in the graphic and website template) as well as accurate wording, over femvertising. Publishing participant statistics could enhance the coding camp in this area (Iaccarino et al., 2024).

Addressing stereotypes and building confidence is critical to encouraging greater participation among girls. Early exposure to coding and business concepts through school partnerships and extracurricular clubs can help normalize engagement and reduce intimidation (Dasgupta & Stout, 2014). Providing pre-camp workshops or beginner-friendly preparatory materials is an effective way to bridge knowledge gaps and build foundational confidence before formal participation begins (Microsoft & KRC Research, 2018). Highlighting successful female-led business game projects through showcases and competitions reinforces the notion that girls belong in this field and can achieve success.

In the OSCAR platform (first release)
To foster gender inclusion, OSCAR offers several built-in supports:

- **Visibility and Representation**: Participant profiles allow the upload of profile pictures and personal descriptions, showcasing diversity and breaking gender stereotypes.
- **Beginner-Friendly Environments**: OSCAR's material templates and project workspaces are customizable to include socially relevant, non-competitive, and application-driven challenges, aligning with research findings on what attracts more girls.
- **Mentorship and Role Models**: Peer tutor functionalities allow the promotion of female peer leaders within the platform, strengthening the presence of role models.
- **Data Collection for Improvement**: Through Participant Forms and profile analytics, organizers can track gender participation statistics to measure and improve the inclusivity of their events.
- **Confidence Building**: The integrated Stepper and Certificate systems allow girls to track and celebrate incremental achievements, building self-efficacy.
- **Promotional Content**: Project Results pages can feature success stories and socially impactful projects led by diverse participants, inspiring future attendees.

These elements help coding camps systematically implement recommendations to make events welcoming, inclusive, and attractive for girls, thus contributing to broader participation in STEM fields.

2.5 Operational Checklist and Coding Camp Evaluation

To align pedagogical goals, organizational choices, and impact evaluation, it is useful to define a set of working procedures and criteria for evaluating the coding camp already in the design phase. Operational procedures may include a multichannel invitation strategy (coordination with schools and partners, institutional email, in-person announcements, social media), a pre-event communication sent a few days before the start with clear instructions on access, required materials, and hybrid participation rules, and an explicit distribution of staff roles (facilitators, online support, technical support, logistics), with minimum staffing levels for each. Coding camp evaluation can include quantitative aspects such as number of participants, completion rate, total hours delivered, and gender balance, as well as qualitative indicators such as participants' intention to recommend the camp, perceived learning gains, or self-reported confidence in the targeted skills. Regularly monitoring these indicators across editions allows organizers to iteratively refine the design and document the impact of coding camps over time. Operational checklist and coding camp evaluation are topics covered in more detail in chapter 5 of the handbook.

In the OSCAR platform (first release)

To ensure effective collaboration, task assignment, and project management within the coding camps, the OSCAR platform provides the following procedural supports:

- Role Assignment: Participants, Peer Tutors, Facilitators, and Coordinators are assigned to role-specific user groups. The platform's permission system uses these user groups to determine access to the platform features and the coding camp content.
- Task Distribution: Facilitators can distribute tasks via dedicated task documents or each team's own team page. Goals and milestones can be represented visually in many ways; as an event in the Calendar, or a series of discrete steps using stepper and goal tools, etc.
- Team Management: Participants are grouped into teams using the platform's user group functionality. Team pages include functions to manage the team members, team name, etc.: access to and visibility of these functions can be controlled at the level of user groups or even single user accounts.
- Meeting Coordination: Team meetings, feedback sessions, and checkpoints are scheduled using the shared Calendar.
- Document Sharing: Materials and exercises are stored centrally, ensuring that every participant has immediate access to up-to-date documents.
- Progress Monitoring: Coordinators track project development and team performance through integrated Dashboards.
- Feedback Loops: Feedback is collected through Exercises, Team pages, and direct messaging, supporting continuous improvement.

These structured working procedures enhance coordination, reduce ambiguity, and ensure that coding camps remain agile, participant-centered, and outcome-driven.

2.6 Strategies to support the green and digital transformation

2.6.1 Activities to support green and digital transformation

As detailed in Section 2.1, engaging students in solving real-world problems by creating solutions with a social value serves as a great motivator, promotes inclusion, and encourages innovation. Specifically, asking students to develop solutions toward solving an environmental problem will help incorporate environmental and climate considerations into the coding camp, thus fostering civic and environmental skills, particularly in terms of safety. For example, participants may develop digital tools for monitoring carbon footprints, optimizing energy consumption, or gamified applications that promote eco-friendly behaviors.

Developing solutions to address environmental problems also fosters green skills, which are the skills workers need to reduce emissions in their working practices. Specifically, it can promote the transversal skill "engaging others in environment friendly behaviors" (as classified by the ESCO framework of the European Commission[4]), which involves the ability to promote sustainability and environmentally friendly actions. To this end, participants may be tasked with creating solutions that educate users about environmental issues and sustainability practices. Examples may include apps that encourage recycling or serious games designed to raise awareness of the impact of human activities on the environment.

2.6.2 Assessment strategy

Following the guidelines in this handbook, instructors can promote civic and environmental skills during coding camps, with a particular emphasis on:

- Safety (protecting the environment), i.e., being aware of the environmental impact of digital technologies and their use[5];
- Green skills, focusing on the ability to "engage others in environmentally friendly behaviours".

Table 5 outlines the proficiency levels related to "safety: protecting the environment," as defined in DigComp 2.2. This table can be used to evaluate participants' learning outcomes. For instance, if a team has created an app that utilizes AI, they can be rated

[4] https://esco.ec.europa.eu/en

[5] Vuorikari, R., Kluzer, S. and Punie, Y., DigComp 2.2: The Digital Competence Framework for Citizens - With new examples of knowledge, skills and attitudes, EUR 31006 EN, Publications Office of the European Union, Luxembourg, 2022, ISBN 978-92-76- 48882-8, doi:10.2760/115376, JRC128415

as "intermediate" (4) if they incorporate strategies in their app to limit the environmental impact associated with heavy data processing and computing power.

Table 5. Safety: protecting the environment. Proficiency levels defined in the DigComp 2.2.

Foundation	**1**	**At basic level and with autonomy and appropriate guidance where needed, I can**	Recognize simple environmental impacts of digital technologies and their use.
	2	**At basic level and with autonomy and appropriate guidance where needed, I can**	Recognize simple environmental impacts of digital technologies and their use.
Intermediate	**3**	**On my own and solving straightforward problems, I can**	Indicate well-defined and routine environmental impacts of digital technologies and their use.
	4	**Independently, according to my own needs, and solving well-defined and non-routine problems, I can**	Discuss ways to protect the environment from the impact of digital technologies and their use
Advanced	**5**	**As well as guiding others, I can**	Show different ways to protect the environment from the impact of digital technologies and their use.
	6	**At advanced level, according to my own needs and those of others, and in complex contexts, I can**	Choose the most appropriate solutions to protect the environment from the impact of digital technologies and their use.
Highly specialized	**7**	**At highly specialized level, I can:**	- create solutions to complex problems with limited definitions that are related to protecting the environment from the impact of digital technologies and their use. - integrate my knowledge to contribute to professional practice and knowledge and guide others in protecting the environment.
	8	**At the most advanced and specialized level, I can:**	- create solutions to solve complex problems with many interacting factors that are related to protecting the environment from the impact of digital technologies and their use. - propose new ideas and processes to the field.

OSCAR Findings

For assessing the ability to "engage others in environment friendly behaviours", the solutions developed by the participants should be assessed in terms of their ability to promote sustainability and environmentally friendly actions, based on the levels defined in Table 6.

Table 6. Green skill "engage others in environment friendly behaviours". Proposed proficiency levels.

Foundation	**1**	- The solution vaguely aims at engaging others in environmentally friendly behaviors
	2	- The solution sets the goal of engaging others in environmentally friendly behaviors. However, the goal is too broad.
Intermediate	**3**	- The solution sets the goal of engaging others in environmentally friendly behaviors. The goal is feasible.
	4	- The solution sets the goal of engaging others in environmentally friendly behaviors. The goal is feasible - The solution is not designed effectively to achieve the intended goal.
Advanced	**5**	- The solution sets the goal of engaging others in environmentally friendly behaviors. The goal is feasible. - The solution is partially effective in achieving the intended goal
	6	- The solution effectively promotes sustainability and environmentally friendly actions
Highly specialized	**7**	- The solution effectively promotes sustainability and environmentally friendly actions - The solution adopts at least one strategy to limit its environmental impact
	8	- The solution effectively promotes sustainability and environmentally friendly actions - The solution adopts more than one strategy to limit its environmental impact

In the OSCAR platform (first release):

The OSCAR platform serves as an example for participants on how digital transformation can be supported. In fact, the platform allows participants to link their results to a personal portfolio, thereby encouraging the use of digital tools to document and showcase their achievements. The platform supports generating digital coding camp certificates in an automated way, relying on the assessment data saved on the platform. The platform includes several tools to make student assessment faster and easier for teachers: for example, gathering and visualizing assessment data can be automated with scripts. The platform supports arbitrary assessment forms and rubrics. Additionally, participants could get suggestions for leveraging new skills, continuing previous projects, and accessing earlier projects at any time. Furthermore, the platform allows facilitators to gather all the necessary data for assessing civic and environmental skills.

2.7 References

Alotaibi, H., Al-Khalifa, H. S., & AlSaeed, D. (2020). Teaching Programming to Students with Vision Impairment: Impact of Tactile Teaching Strategies on Student's Achievements and Perceptions. Sustainability, 12(13), 5320. https://doi.org/10.3390/su12135320

Altintas, T., Gunes, A., & Sayan, H. (2016). A peer-assisted learning experience in computer programming language learning and developing computer programming skills. Innovations in Education and Teaching International, 53(3), 329–337.

Asim, S., Ponners, P. J., Bartlett, C., Parker, A., & Star, R. (2020). Differentiating instruction for middle school students in virtual learning environments. Delta Kappa Gamma Bulletin, 86(3).

Barker, J., Aspray, W., (2006) The State of Research on Girls and IT, in Joanne Cohoon, and William Aspray (eds), Women and Information Technology: Research on Underrepresentation. https://doi.org/10.7551/mitpress/9780262033459.003.0001

Benzing, V., & Schmidt, M. (2019). The effect of exergaming on executive functions in children with ADHD: A randomized clinical trial. Scandinavian Journal of Medicine & Science in Sports, 29(8), 1243–1253.

Bian, L., Leslie, S. J., & Cimpian, A. (2017). Gender stereotypes about intellectual ability emerge early and influence children's interests. Science, 355(6323), 389–391. https://doi.org/10.1126/science.aah6524

Bioulac, S., Lallemand, S., Fabrigoule, C., Thoumy, A. L., Philip, P., & Bouvard, M. P. (2014). Video game performances are preserved in ADHD children compared with controls. Journal of Attention Disorders, 18(6), 542–550.

Boos, R. (1971). Creativity in education. Journal of Thought, 6(4), 274–280.

Bouthillette, K. (2016). Tutor, guide, lead: Examining the experiences of peer tutors. Higher Education Student Work, 11. https://scholarworks.merrimack.edu/soe_studentpub/11

Bugaj, T. J., Blohm, M., Schmid, C., et al. (2019). Peer-assisted learning (PAL): Skills lab tutors' experiences and motivation. BMC Medical Education, 19, 353. https://doi.org/10.1186/s12909-019-1760-2

Burnett, M., & Myers, B. A. (2014). Future of end-user software engineering: Beyond the silos. In Proceedings of the Future of Software Engineering (pp. 201–211). https://doi.org/10.1145/2593882.2593896

Castelli, F. R., & Sarvary, M. A. (2021). Why students do not turn on their video cameras during online classes and an equitable and inclusive plan to encourage them to do so. Ecology and Evolution, 11(8), 3565–3576. https://doi.org/10.1002/ece3.7123

Chen, H., Park, H. W., & Breazeal, C. (2020). Teaching and learning with children: Impact of reciprocal peer learning with a social robot on children's learning and emotive engagement. Computers & Education, 150, 103836. https://doi.org/10.1016/j.compedu.2020.103836

Cheryan, S., Master, A., & Meltzoff, A. N. (2015). Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering by diversifying stereotypes. Frontiers in Psychology, 6, 49. https://doi.org/10.3389/fpsyg.2015.00049

Cheryan, S., Siy, J. O., Vichayapai, M., Drury, B. J., & Kim, S. (2011). Do female and male role models who embody STEM stereotypes hinder women's anticipated success in STEM? Social Psychological and Personality Science, 2(6), 656–664. https://doi.org/10.1177/1948550611405218

Cortese, S., Ferrin, M., Brandeis, D., Buitelaar, J., Daley, D., Dittmann, R. W., et al. (2015). Cognitive training for attention-deficit/hyperactivity disorder: Meta-analysis of clinical and neuropsychological outcomes from randomized controlled trials. Journal of the American Academy of Child & Adolescent Psychiatry, 54(3), 164–174.

Dasgupta, N. (2011). Ingroup experts and peers as social vaccines who inoculate the self-concept: The stereotype inoculation model. Psychological Inquiry, 22(4), 231–246. https://doi.org/10.1080/1047840X.2011.607313

Dasgupta, N., & Stout, J. G. (2014). Girls and women in science, technology, engineering, and mathematics: STEMing the tide and broadening participation in STEM careers. Psychological Science in the Public Interest, 15(1), 1–78. https://doi.org/10.1177/2372732214549471

Diekman, A. B., Brown, E. R., Johnston, A. M., & Clark, E. K. (2010). Seeking congruity between goals and roles: A new look at why women opt out of science, technology, engineering, and mathematics careers. Psychological Science, 21(8), 1051–1057. https://doi.org/10.1177/0956797610377342

Doğru, Ç. (2021). Fostering innovation in organizations with the help of novel management strategies. In Strategic Outlook in Business and Finance Innovation: Multidimensional Policies for Emerging Economies (pp. 173–183). Emerald Publishing Limited.

Dweck, C. S. (2006). Mindset: The new psychology of success. Random House.

Eccles, J. S. (2011). Gendered educational and occupational choices: Applying the Eccles et al. model of achievement-related choices. International Journal of Behavioral Development, 35(3), 195–201. https://doi.org/10.1177/0165025411398185

Echeverry Chaves, N. (2015). Diseño de un videojuego didáctico de educación cívica para niños autistas, TDAH y discapacidad cognitiva. Teoría y Praxis Investigativa, 9.

European Commission. (2022). STEM roles and gender diversity trends in the EU. Retrieved from https://ec.europa.eu/eurostat/web/products-eurostat-news/-/edn-20220211-2

Faenza, F., Canali, C., & Carbonaro, A. (2021). Evaluating different approaches to closing the gender gap at ICT summer camps in Italy. In 4th International Conference on Gender Research (ICGR 2021) (pp. 104–113).

Fahim, S. S., & Khalil, R. M. R. (2015). Addressing differentiation: Effective classroom teaching strategies. On Research Methodology, 3(2), 200.

Ferraz, C., & Gama, K. (2019). A case study about gender issues in a game jam (ICGJ ’19). In Proceedings of the International Conference on Game Jams, Hackathons and Game Creation Events (8 pages)

Fisher, J.A. (Ed.). (2021). Augmented and Mixed Reality for Communities (1st ed.). CRC Press. https://doi.org/10.1201/9781003052838

Fronza, I., & Corral, L. (2024). A facilitator’s guide to create and consolidate a teenage coding camp. ACM Inroads, 15(2), 31–38. https://doi.org/10.1145/3643726

Fronza, I., Corral, L., & Pahl, C. (2020). End-user software development: Effectiveness of a software engineering-centric instructional strategy. Journal of Information Technology Education, 19, 367–393. https://doi.org/10.28945/4580

Fronza, I., Corral, L., Iaccarino, G., & Pahl, C. (2021). Enabling peer-led coding camps by creating a seed effect in young students. In Proceedings of the 22nd Annual Conference on Information Technology Education (SIGITE '21) (pp. 117–122). ACM. https://doi.org/10.1145/3450329.3476860

Fronza, I., Corral, L., Wang, X., & Pahl, C. (2022). Keeping fun alive: An experience report on running online coding camps. In ICSE SEET 2022. https://doi.org/10.1145/3510456.3514153

Fronza, I., Iaccarino, G., & Corral, L. (2024). Nurturing hybrid work literacy in upper secondary schools: Selecting the best hybrid work configuration for coding camps. In 2024 IEEE Frontiers in Education Conference (FIE) (pp. 1–9). IEEE. https://doi.org/10.1109/FIE61694.2024.10893177

Fronza, I., Ihantola, P., Riikola, O. P., Iaccarino, G., Mikkonen, T., García Rytman, L., … & Rossano, V. (2025). Towards s'more connected coding camps. In Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1 (pp. 353–359). ACM.

Gammie, E., & Matson, M. (2007). Group assessment at final degree level: An evaluation. Accounting Education: An International Journal, 16(2), 185–206. https://doi.org/10.1080/09639280701234609

García-Guerrero, J., & Calleros, J. M. (2021). Videojuegos en educación especial: niños con TDAH – Video games in special education: Children with ADHD. Revista de la Asociación Interacción Persona Ordenador (AIPO), 2(1), 48–59.

García-Redondo, P., García, T., Areces, D., Núñez, J. C., & Rodríguez, C. (2019). Serious games and their effect improving attention in students with learning disabilities. International Journal of Environmental Research and Public Health, 16(14), 2480.

García-Ríos, C. A., & García-Ríos, V. E. (2020). Videojuegos para niños con trastorno por déficit de atención e hiperactividad. Dominio de las Ciencias, 6(2), 706–717.

González, C. S., del Río, N., & Adelantado, V. (2018). Exploring the benefits of using gamification and videogames for physical exercise: A review of state of art. International Journal of Interactive Multimedia and Artificial Intelligence, 5(2), 46–52.

Happe, L., Buhnova, B., Koziolek, A., & Wagner, I. (2020). Effective measures to foster girls' interest in secondary computer science education. Education and Information Technologies, 26, 2811–2829.

Happonen, A., Tikka, M., & Usmani, U. A. (2021). A systematic review for organizing hackathons and code camps in COVID-19-like times: Literature in demand to understand online hackathons and event result continuation. In 2021 International Conference on Data and Software Engineering (ICoDSE) (pp. 1–6). IEEE. https://doi.org/10.1109/ICoDSE53690.2021.9648459

Hardt, D., Nagler, M., Rincke, J. (2022) Tutoring in (Online) Higher Education: Experimental Evidence. CESifo Working Paper No. 9555, Available at SSRN: https://ssrn.com/abstract=4031193 or http://dx.doi.org/10.2139/ssrn.4031193

Hattie, J. (2023). Visible learning: The sequel: A synthesis of over 2,100 meta-analyses relating to achievement. Routledge

Herbsleb, J. D., & Moitra, D. (2001). Global software development. IEEE Software, 18(2), 16–20. https://doi.org/10.1109/52.914732

Herinek, D., Woodward-Kron, R., & Ewers, M. (2024). "Between formulas and freestyle": A qualitative analysis of peer tutor preparation and its impact on peer relations. BMC Medical Education, 24(1), 1173. https://doi.org/10.1186/s12909-024-06191-7

Iaccarino, G., Fronza, I., Mikkonen, T., Ihantola, P., & Tosi S. Towards Differentiated Instruction in Coding Camps. In 26th Annual ACM Conference on Cybersecurity and Information Technology Education (ACM SIGCITE '25), November 06–08, 2025. Sacramento, CA, USA. https://doi.org/10.1145/3769694.3771168

Iaccarino, G., Fronza, I., Tosi, S., & Corral, L. (2024). Designing inclusive and gender-balanced coding camps: A comprehensive assessment framework. In The 25th Annual Conference on Information Technology Education (SIGITE '24), October 10–12, 2024, El Paso, TX, USA. ACM. https://doi.org/10.1145/3686852.3689701

Ip, C. Y. (2024). Fostering social entrepreneurship in university students: The moderating role of entrepreneurial creativity. Studies in Higher Education, 1–19.

Kaplan, A., & Flum, H. (2012). Identity formation in educational settings: A contextualized view of theory and research in practice. Contemporary Educational Psychology, 37(3), 171–175. https://doi.org/10.1016/j.cedpsych.2012.01.003

Kert, S. B., Read, J., Uğraş, T., Erkoç, M. F., Özçakır, F. C., Berksoy, İ., Tuncer, T., Cassidy, B., & Zubair, M. S. (2021). Design Ideas for a Learning Material to Teach Coding to Children with Visual Impairment. In ICERI2021 Proceedings (pp. 5864–5871). IATED.

Köhlke, J., Hanna, S., & Schütz, J. (2021). Cross-domain stakeholder alignment in collaborative SoS – LEGO® Serious Play® as a boundary object. In 2021 16th International Conference of System of Systems Engineering (SoSE) (pp. 108–113). IEEE. https://doi.org/10.1109/SOSE52739.2021.9497469

Kovaleva, Y., Happonen, A., & Hasheela-Mufeti, V. (2022). Pros and cons of running educational hackathons in a gender-neutral fashion. In 3rd International Workshop on Gender Equality, Diversity and Inclusion in Software Engineering (pp. 27–34).

Krug, D. L., Bowman, E., Barnett, T., Pollock, L., & Shepherd, D. (2021). Code Beats: A virtual camp for middle schoolers coding hip hop. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21) (pp. 397–403). ACM. https://doi.org/10.1145/3408877.3432424

Kurkovsky, S., Ludi, S., & Clark, L. (2019). Active learning with LEGO for software requirements. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19) (pp. 218–224). ACM. https://doi.org/10.1145/3287324.3287444

Lee, S.-H., Wehmeyer, M. L., Soukup, J. H., & Palmer, S. B. (2010). Impact of Curriculum Modifications on Access to the General Education Curriculum for Students with Disabilities. Exceptional Children, 76(2), 213–233. https://doi.org/10.1177/001440291007600205

Leslie, S. J., Cimpian, A., Meyer, M., & Freeland, E. (2015). Expectations of brilliance underlie gender distributions across academic disciplines. Science, 347(6219), 262–265. https://doi.org/10.1126/science.1261375

Lewis, C. M., Anderson, R. E., & Yasuhara, K. (2016). "I don't code all day": Fitting in computer science when the stereotypes don't fit. In Proceedings of the 2016 ACM Conference on International Computing Education Research (pp. 23–32).

Lock, R. H., & Kingsley, K. V. (2007). Empower Diverse Learners With Educational Technology and Digital Media. Intervention in School and Clinic, 43(1), 52–56. https://doi.org/10.1177/10534512070430010701

Lőrincz, B., Iudean, B., & Vescan, A. (2021). Experience report on teaching testing through gamification. In Proceedings of the 3rd International Workshop on Education through Advanced Software Engineering and Artificial Intelligence (pp. 15–22). https://doi.org/10.1145/3472673.3473960

Ludi, S., et al. (2018). Teaching Inclusive Thinking to Undergraduate Students in Computing Programs. In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18) (pp. 717–722). ACM. https://doi.org/10.1145/3159450.3159512

Margolis, J., Fisher, A., & Miller, F. (2000). Unlocking the clubhouse: Women in computing. MIT Press. DOI: 10.1145/792548.611896

Master, A., Cheryan, S., & Meltzoff, A. N. (2016). Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science. Journal of Educational Psychology, 108(3), 424–437. https://doi.org/10.1037/edu0000061

Mayer, J. D., & Salovey, P. (1993). The intelligence of emotional intelligence. Intelligence, 17(4), 433–442.

Microsoft & KRC Research. (2018). Closing the STEM gap: Why STEM classes and careers still lack girls and what we can do about it. https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RE1UMWz

Mikkonen, T., Adil, M., Fronza, I., Iaccarino, G., & Ihantola, P. (2025). To co-locate or not to co-locate? On the impact of hybrid work on software design process. In The ACM International Conference on the Foundations of Software Engineering (FSE 2025). In press.

Mooney, C., & Becker, B. A. (2021). Investigating the impact of the COVID-19 pandemic on computing students' sense of belonging. ACM Inroads, 12(2), 38–45. https://doi.org/10.1145/3408877.3432407

Newton, K. J., Leonard, J., Buss, A., Wright, C. G., & Barnes-Johnson, J. (2020). Informal STEM: Learning with robotics and game design in an urban context. Journal of Research on Technology in Education, 52(2), 129–147. https://doi.org/10.1080/15391523.2020.1713263

Oakley, B., Felder, R. M., Brent, R., & Elhajj, I. (2004). Turning student groups into effective teams. Journal of Student-Centered Learning, 2(1), 9–34

Porras, J., & Khakurel, J. (2021). Experiences and lessons learned from onsite and remote teamwork-based courses in software engineering. In 2021 International Conference on Data and Software Engineering (ICoDSE) (pp. 1–9). IEEE. https://doi.org/10.1109/ICoDSE53690.2021.9648490

Porras, J., Knutas, A, Ikonen, J., Happonen, A., Khakurel, J., & Herala, A. (2019). Code camps and hackathons in education – Literature review and lessons learned. In Proceedings of HICSS 2019. https://hdl.handle.net/10125/60213

Powell, J., Hayden, L. B., Cannon, A., Wilson, B., & Nolte, A. (2021). Organizing online hackathons for newcomers to a scientific community: Lessons learned from two events. In Sixth Annual International Conference on Game Jams, Hackathons, and Game Creation Events (pp. 78–82). https://doi.org/10.1145/3472688.3472700

Resnick, M. (2014). Give P's a chance: Projects, peers, passion, play. In Constructionism and Creativity: Proceedings of the Third International Constructionism Conference (pp. 13–20). Austrian Computer Society.

Richard, G. T., Kafai, Y. B., Adleberg, B., & Telhan, O. (2015). StitchFest: Diversifying a college hackathon to broaden participation and perceptions in computing. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (pp. 114–119).

Roscoe, R., & Chi, M. (2007). Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions. Review of Educational Research, 77, 534–574.

Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. American Psychologist, 55(1), 68–78. https://doi.org/10.1037/0003-066X.55.1.68

Sablis, A., Gonzalez-Huerta, J., Zabardast, E., & Šmite, D. (2019). Building LEGO towers: An exercise for teaching the challenges of global work. ACM Transactions on Computing Education (TOCE), 19(2), 1–32. https://doi.org/10.1145/3218249

Schleyer, G. K., Langdon, G. S., & James, S. (2005). Peer tutoring in conceptual design. European Journal of Engineering Education, 30(2), 245–254.

Sellier, N., & An, P. (2020). How peripheral interactive systems can support teachers with differentiated instruction: Using fireflies as a probe. In Proceedings of the 2020 ACM Designing Interactive Systems Conference (pp. 1117–1129)

Shapiro, J. R., & Williams, A. M. (2012). The role of stereotype threats in undermining girls' and women's performance and interest in STEM fields. Sex Roles, 66(3–4), 175–183. https://doi.org/10.1007/s11199-011-0051-0

Shi, W., Malachowsky, S., El-Glaly, Y., Yu, Q., & Krutz, D. E. (2020). Presenting and Evaluating the Impact of Experiential Learning in Computing Accessibility Education. In Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '20). ACM. https://doi.org/10.1145/3377814.3381710

Sorathia, K., & Servidio, R. (2012). Learning and experience: Teaching tangible interaction and edutainment. Procedia – Social and Behavioral Sciences, 64, 265–274. https://doi.org/10.1016/j.sbspro.2012.11.031

Spencer, S. J., Steele, C. M., & Quinn, D. M. (1999). Stereotype threat and women's math performance. Journal of Experimental Social Psychology, 35(1), 4–28. https://doi.org/10.1006/jesp.1998.1373

Takeda, S., & Homberg, F. (2014). The effects of gender on groupwork process and achievement: An analysis through self- and peer-assessment. British Educational Research Journal, 40(2), 373–396. https://doi.org/10.1002/berj.3088

Tomlinson, C. A. (2017). How to differentiate instruction in academically diverse classrooms (3rd ed.). ASCD.

Topping, K. J. (1996). The effectiveness of peer tutoring in further and higher education: A typology and review of the literature. Higher Education, 32(3), 321–345.

Tye, C., Johnson, S. P., Kelly, K. A., Asherson, P., & Kuntsi, J. (2016). Response time variability under slow and fast-incentive conditions in children with ASD, ADHD and ASD+ADHD. Journal of Child Psychology and Psychiatry, 57(12), 1414–1423

UNESCO. (2023). Inclusion and gender equality: Brief on inclusion in education. In UNESCO briefs on inclusion in education (pp. 1–11).

Van Alphen, E., & Bakker, S. (2016). Lernanto: Using an ambient display during differentiated instruction. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (pp. 2334–2340).

Vollmeyer, R., & Rheinberg, F. (2005). A surprising effect of feedback on learning. Learning and Instruction, 15(6), 589–602.

Wahid, A. H., Rahman, R. A., Mustaffa, W. S. W., Ahmad, N. L., Ramdan, M. R., & Muslimat, A. M. (2024). Best social entrepreneurship teaching and learning strategies for promoting students' social entrepreneurial minds: A scoping review. International Journal of Learning, Teaching and Educational Research, 23(3), 23–47.

Wang, M. T., & Degol, J. L. (2017). Gender gap in science, technology, engineering, and mathematics (STEM): Current knowledge, implications for practice, policy, and future directions. Educational Psychology Review, 29(1), 119–140. https://doi.org/10.1007/s10648-015-9355-x

Wang, T. (2024). Online peer tutoring programs fostering community and learning skills among college students. Education and Information Technologies, 29, 21751–21788. https://doi.org/10.1007/s10639-024-12656-5

Worsley, M., & Bar-El, D. (2020). Inclusive Making: Designing tools and experiences to promote accessibility and redefine making. Computer Science Education. https://doi.org/10.1080/08993408.2020.1863705

Xhomara, V., & Uka, A. (2023). Creativity in education: Fostering creativity in the classroom using creative teaching methods. Zenodo, 26(2), 2–29.

Yousof, S. M., EidAlsawat, R., Almajed, J. A., Alkhamesi, A. A., Alsuhaimi, R. M., Alsed, S. A., & Salem, I. M. W. (2021). The possible negative effects of prolonged technology-based online learning during the COVID-19 pandemic on body functions and wellbeing: A review article. Journal of Medical Science, 90(3), e522–e522. https://doi.org/10.20883/medical.e522

Zyda, M. (2005). From visual simulation to virtual reality to games. Computer, 38(9), 25–32. https://doi.org/10.1109/MC.2005.297

Chapter 3 Coding Camp Development

This chapter provides a comprehensive description of the materials that were planned during the design phase. It delves into the specifics of what will be covered in the course and the resources that will be used to facilitate learning.

3.1 Strategies for online/hybrid non-conventional learning experiences

3.1.1 Schedule of the coding camp

The OSCAR coding camps comprise 20 hours of activities delivered over five sessions in consecutive days. The program is structured into three main parts: 4 hours dedicated to problem understanding and problem-solving; 4 hours on fundamental programming concepts and development tools; and 12 hours devoted to the iterative, team-based development, culminating in final presentations.

3.1.2 Structure of each session

Table 5. Template for planning each session of a coding camp.

Session	Hours	Objective	Content	Skills	Assessment
Session number	Number of hours	Main objective of the session	Material that students will learn during the session	List of skills fostered in this session	Assessment strategy for each skill

As an example, in the following we show how the template was completed for each OSCAR coding camp domain (Apps, Serious Games, and XR).

Table 6. Sessions of the app coding camp (use case)

Session	Hours	Teaching style	Objective	Content	Skills
1	4	In-person	Introduction; problem understanding and problem-solving; team bonding	Presentation of main concepts; game (team bonding); Presentations	• Problem solving • Communication and collaboration

2	4	Online / Hybrid	Fundamental programming concepts and development tools; Usage of AI in app development; AI and sustainability	Live step-by-step exercises and assignments; Presentation;	• Problem solving • Communication and collaboration • Digital content creation
3	4	Online / Hybrid	How apps can solve environmental issues; iterative development of apps in teams	Presentations; Live step-by-step exercises and feedback; game (teamwork, communication and collaboration)	• Problem solving • Communication and collaboration • Digital content creation • Civic and environmental skills
4	4	Online / Hybrid	Iterative development of apps in teams	Live step-by-step exercises and feedback; game (teamwork, hybrid work)	• Problem solving • Communication and collaboration • Digital content creation
5	4	Online / Hybrid	Iterative development of apps in teams; pitch	Live step-by-step exercises and feedback	• Problem solving • Communication and collaboration • Digital content creation • Civic and environmental skills • Entrepreneurial skills

The assessment strategy included observing participants by facilitators and peer tutors, assessing final presentations (i.e., pitches), and final projects. To collect observations, specific surveys have been created on the OSCAR platform for facilitators and peer tutors to complete.

Table 7. Skills and corresponding assessment strategies.

Skills	Assessment
Problem solving	Observations (facilitators and peer tutors) Evaluation of the final project.
Communication and collaboration	Observations (facilitators and peer tutors)

	Evaluation of the final project.
Digital content creation	Observations (facilitators and peer tutors) Evaluation of the final project.
Civic and environmental skills	Evaluation of the final project.
Entrepreneurial skills	Evaluation of the final pitch.

The assessment of the final projects is based on the extraction of several metrics that contribute to assessing various skills. The assessment, therefore, checks whether the app solves a problem related to society/the environment, whether it incorporates strategies to limit the environmental impact associated with heavy data processing and computing power, and whether it engages others in environmentally friendly behaviors. The overall quality of the apps is then evaluated by extracting component metrics, computational concepts, blocks, code smells, complexity metrics, and size.

Table 8. Sessions of the Serious Games coding camp (pilot).

Session	Hours	Objective	Content	Skills	Assessment
1	4	Kick-off & game concept	Team formation, theme selection, intro to serious games, overview of GDevelop UI, first storyboard skeleton	• Problem solving • Communication and collaboration	Observations (facilitators and peer tutors) Participation Draft storyboard
2	4	Core loop prototype	Sprites/scenes, events logic, collisions, variables; build a minimal playable loop	• Problem solving • Communication and collaboration • Entrepreneurial skills • Digital content creation	Observations (facilitators and peer tutors) Playable milestone ("core loop")
3	4	Mechanics & progression	Level design, difficulty balance, feedback/rewards, UX basics; iterate via playtesting	• Problem solving • Communication and collaboration • Entrepreneurial skills • Digital content creation	Observations (facilitators and peer tutors) Peer playtest feedback Iteration notes

4	4	Polish & coherence	Audio/visual polish, bug fixing, readability, onboarding/tutorial, narrative coherence	• Problem solving • Communication and collaboration • Entrepreneurial skills • Digital content creation	Observations (facilitators and peer tutors) Near-final build
5	4	Showcase & reflection	Final build, presentation/pitch, discussion of design choices and learning goals	• Communication and collaboration • Entrepreneurial skills • Digital content creation	Observations (facilitators and peer tutors) Final artefact assessment Assessment of final pitch

Table 9. Sessions of the XR coding camp (pilot).

Session	Hours	Objective	Content	Skills	Assessment
1	4	Introduction to XR and camp overview	Overview of the camp, team formation, theme selection, platform introduction, and introductory tasks	• Problem solving • Communication and collaboration • Creativity	Observations (facilitators and peer tutors) Participation
2	4	Understanding 3D environments and defining the project idea	3D exploration, project ideation, use of generative AI for concept design	• Creativity • Communication and collaboration • Digital content creation	Review of project ideas and facilitator feedback
3	4	Creation of 3D characters aligned with the project theme	Voxel character modelling, individual design aligned with team concept	• Problem solving • Communication and collaboration • Creativity • Digital content creation	Review of character designs and peer feedback

4	4	Introduction to interaction and animation	Basic interaction concepts, simple programming exercise, character animation	• Problem solving • Communication and collaboration • Entrepreneurial skills • Digital content creation	Demonstration of interactive scene and animated characters
5	4	Integration in AR and final presentations	AR integration, testing, project refinement and final presentations	• Presentation skills • Communication and collaboration • Entrepreneurial skills • Digital content creation	Final project presentation, peer and mentor feedback

3.1.3 Teaching style of each session

The camp offered hybrid participation, allowing participants to alternate between online and onsite attendance on different days. The first session was conducted in person, while the remaining four sessions were held in hybrid format. For those unable to reach the venue, there was also an option for fully online participation. The coding camp followed a learning-by-doing approach, supported by short, visual, and progressively structured presentations, and live demonstrations. Teaching materials included slide decks and descriptions of the games executed during the camp.

During the Serious Game camp, a blended set of learning resources was used: slide decks and short presentations introducing key concepts (game mechanics, goals, feedback loops, and rapid prototyping) and a set of short video tutorials focused on the core functionalities of the low-code/no-code framework GDevelop. Video tutorials were intentionally used to (i) support students' initial familiarization with the tool and (ii) enable asynchronous recap of key procedures (e.g., events logic, sprites, scenes, collisions), which is particularly helpful in intensive workshop formats. To structure the design process, a "skeleton" templates for storyboarding and serious game design was provided. Each group compiled these templates to define learning/engagement goal, target audience, core loop, rules, level progression, feedback/rewards, assets list, and testing plan.

The XR coding camp adopted a hands-on, practice-oriented learning structure designed for secondary school students with little or no prior experience in programming or extended reality technologies. Teaching materials were intentionally simple, visual, and progressive, focusing on enabling participants to achieve concrete outcomes at each stage of the camp.

The learning process was supported by a combination of short introductory presentations, live demonstrations, and step-by-step written guides. These materials covered key aspects such as basic 3D concepts, asset creation, importing and configuring elements in Unity, and implementing simple interactions. Preconfigured Unity templates were provided to reduce technical barriers and allow students to focus on understanding core concepts rather than on complex setup procedures.

Rather than following a lecture-based approach, the camp was structured around micro-tasks and practical challenges. Each session concluded with a tangible result, such as a character moving in a 3D environment, an interactive object, or a basic augmented reality scene. This incremental structure helped maintain motivation, supported learning-by-doing, and allowed students to immediately see the impact of their work.

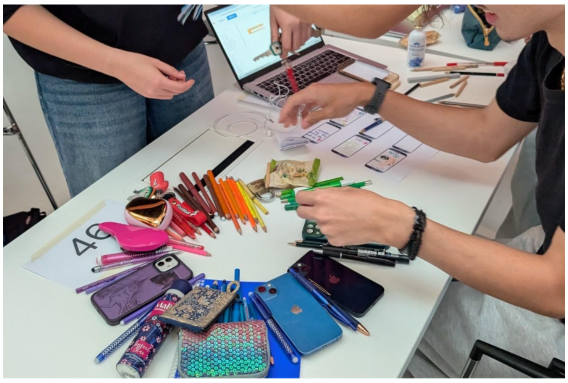
3.2 Facilitation strategies

3.2.1 Games and fun activities

Paper airplane: Each participant creates a paper airplane and writes their name, the reason they are attending the coding camp, and what they hope to learn on it. Once everyone has launched their airplanes together, each participant collects a different airplane and attempts to discover who made it. This process involves interacting with the original maker to learn more about them.

Paper tower: Teams have 18 minutes to build the tallest freestanding tower using 20 A4 sheets of paper provided at each location. Participants can choose their strategy for constructing the tower, such as collaborating in the classroom, involving online participants, or even building multiple towers. Learning goals include prototyping and iterating, communication and collaboration, simple design, and teamwork.

Color wheel: Teams have 15 minutes to create a color wheel using as many colors and objects from their surroundings as possible. Teams can decide to construct the wheel in the classroom, engage online participants in the process, or build multiple wheels. Learning goals include prototyping and iterating, teamwork, and hybrid work.



3.3 Technology

The OSCAR platform supports all the coding camps activities by providing assessment and reporting tools and facilitating teamwork and collaboration via gamification and group tasks. Real-time communication platforms and different software solutions were used to support the specific objective of coding camps. A preference was given to browser-based solutions requiring little or no installation.

3.3.1 Technical requirements (Hardware and Software)

Hardware:

- A laptop PC with a video-audio system (webcam, speakers and microphone)

- A headset for hybrid sessions (to participate in online components) and useful for in-person sessions to prevent audio disturbances between groups

Software: The OSCAR coding camps integrate a videoconferencing tool with plenary + breakout rooms, a low-configuration (ideally browser-based) development environment, and a central hub for materials, communication, submissions, and assessment/badges. Optional tools (polls, quizzes, quick feedback) can support facilitation.

- Thunkable (https://thunkable.com)
- Microsoft Teams (to ensure hybrid collaboration): Team members communicate via Microsoft Teams. When not in a plenary session, both in-person and remote participants accessed their team's breakout room to collaborate.

- GDevelop (https://gdevelop.io)
- Microsoft Teams (to ensure hybrid collaboration): During the serious camp the hybrid participation (remote/on-site) was ensured through Microsoft Teams, using dedicated channels for announcements, Q&A, resource sharing, and group support. For the serious games camp, hybrid operational setup (Teams) was created to support the activities. Structured Teams with: (1) a general channel for announcements and resources, (2) a Q&A/helpdesk channel, and (3) group-specific channels for coordination. Learning resources (slides, design skeletons, tutorial videos) were released with a clear cadence (e.g., “watch before session”, “use during build”, “recap after”). Virtual/remote jam studies recommend explicit facilitation routines (milestones, check-ins, structured sharing moments) to avoid fragmentation and keep teams aligned.

- MagicaVoxel (https://ephtracy.github.io/) - Mixamo (https://www.mixamo.com/#/) - Geenee (https://geenee.ar/) - Unity (https://unity.com/): for the XR coding camp, participants developed an interactive extended reality experience using a simple XR development pipeline. MagicaVoxel was used to create voxel-based characters, which were then animated using Mixamo. The interactive application was developed in Unity, where the assets and animations were integrated, and the interaction logic was implemented. Finally, the experience was deployed as an Augmented Reality (AR) application using Geenee, allowing participants to visualize and interact with the content through AR technology.

- Google Classroom/Meet (to ensure hybrid collaboration): Google Classroom was used for resource sharing and Google Meet for synchronous communication.

Set-up principles:

- Minimize account friction.
- Prefer low-configuration development tools when possible.
- Prepare recovery assets.
- Designate online support staff.

3.3.2 Spaces

Physical space was organized to support collaborative work in small groups, with tables arranged to allow device sharing and close interaction with mentors.

3.3.3 Monitoring and data centralization

Surveys can be used to collect feedback from participants about their overall experience, satisfaction with the program, perceived learning outcomes, and suggestions for improvement.

On the OSCAR platform, a survey was created to collect feedback from participants regarding the platform and their overall satisfaction.

A post-activity questionnaire was administered via Google Forms, (https://forms.gle/9W4w6ZdxqH81hp3v8) divided into thematic sections: perception and degree of satisfaction regarding the effectiveness of teamwork, confidence in the learned concepts, satisfaction with the game engine, and an overall assessment of the coding camp organization

Learning materials and assignment management were organized through Google Classroom, which served as the central hub for the XR coding camp. The platform was used to distribute slides, tutorials, and other learning resources, as well as to collect the final deliverables from each team. As part of the final activity, participants submitted a short video demonstrating their XR prototype, allowing facilitators to review and document the outcomes of the coding camp.

A post-activity questionnaire was administered via Google Forms (https://forms.gle/ejCKzrMjuvaWMegT7), covering modeling, animation, programming, AR visualization, and overall experience. It assessed clarity of instructions, task difficulty, creativity, tool adequacy, teamwork, hybrid collaboration, and overall satisfaction with the coding camp.

3.4 Strategies for creating teams

As described in Section 2.1.6, our experience suggest that the teams should be heterogeneous, with students from different schools, avoiding female unbalance by assigning two females in those groups with female participants. In our findings, the best group composition is with 3 or 4 participants.

Here is a template table to facilitate the creation of teams.

Table 10. Table template to create teams

ID group	Group's name	Surname	Name	email	account
1	Group1	Surname1	Name1		
		Surname2	Name2		
		Surname3	Name3		
		Surname4	Name4		

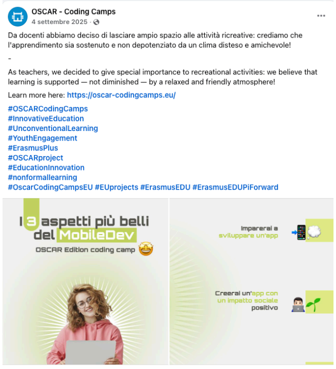
3.5 Strategies to attract more girls

1. **Emphasize collaboration over performance**, valuing multiple forms of contribution, including design, storytelling, testing, and basic programming tasks.

2. Inclusive communication. Use neutral, welcoming visuals; avoid stereotypes; highlight social impact themes (sustainability); and explicitly state that multiple skill sets are valued.

Below some examples of social posts presenting the coding camp as an inclusive and welcoming learning opportunity for everyone.

Table 11. Example of coding dissemination posts to attract girls



3.6 Communication and dissemination strategies.

Each coding camp should include a short "Camp Snapshot" box to make logistics and access requirements immediately clear: delivery mode (e.g., hybrid, enabling remote participation), total hands-on hours, dates/times and venue, target audience and minimum school year, prerequisites (explicitly stating when no prior software development experience is required), required equipment (e.g., laptop/tablet and headphones), and cohort policy (minimum enrolment threshold to confirm the event and maximum number of participants).

The Camp Snapshot should include:

- Delivery mode with specification of remote participation options;
- Total hands-on hours and overall duration;
- Dates, daily schedule, and venue;
- Target audience, including minimum school year or age range;
- Prerequisites, clearly stating if no prior programming experience is required;
- Required equipment (e.g., laptop/tablet, headphones, internet connection, installed software if needed);
- (if necessary) Cohort policy, specifying minimum enrolment threshold to confirm activation of the camp and maximum number of participants to ensure quality interaction.

Providing this structured overview enhances clarity, reduces administrative queries, and supports informed participation.

All relevant information must be made available well in advance to enable students and schools to prepare adequately. Communication should be timely, structured, and consistent across channels.

The dedicated **coding camp web page** should include:

- Dates and timetable
- Venue and access information
- Detailed description of content and learning objectives
- Organisational details (partners involved, trainers, contact persons)
- Target users and expected level of prior knowledge
- Registration procedure and deadlines
- Downloadable materials (e.g., brochure, consent forms, technical instructions)

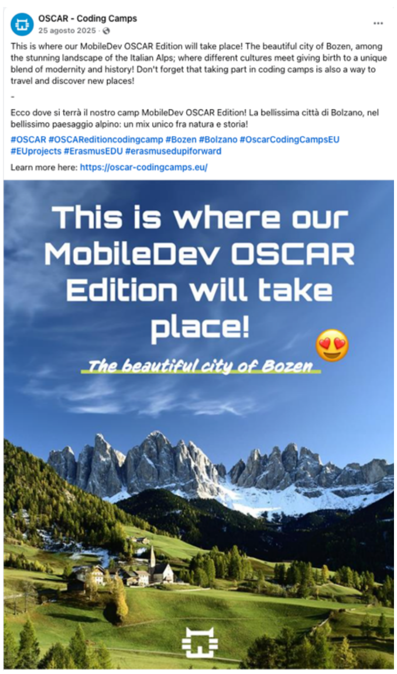
The webpage acts as the central reference point and should remain updated throughout the preparation phase.

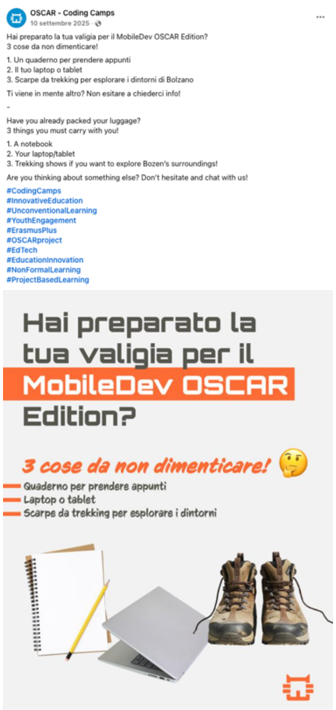

Home Coding camp Argomenti Docenti Dicono di noi Registrazione
LIBERA UNIVERSITÀ DI BOLZANO
MobileDev 2025: Il coding camp di sviluppo di app Android e iOS
15-19 SETTEMBRE, 2025
14MA EDIZIONE
unibz
MobileDev
Immagina, impara, crea!
Il coding camp
Una settimana di attività laboratoriali per simulare un ambiente professionale in cui si sviluppa software per dispositivi mobili. MobileDev 2025 si svolge in modalità ibrida (simulando il lavoro di un team distribuito), facilitando quindi la partecipazione di studenti DA TUTTA ITALIA.
L'iniziativa adotta gli strumenti più innovativi per la formazione in ingegneria del software, sviluppati nell'ambito della ricerca degli organizzatori.


SERIOUS GAME CAMP
UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO
Co-funded by the European Union
OSCAR CODING CAMPS
CAMP
19-20-21-27 Maggio
Università degli Studi di Bari
Dipartimento di Informatica
Impara a sviluppare serious game immersivi!
ISCRIVITI SUBITO
Rappresento una scuola
Sono uno studente

To maximize participation, information about the coding camp should be disseminated through multiple channels

Social media communication, using engaging and accessible language to motivate students to participate. Posts should highlight the practical nature of the activities, showcase past experiences where available, and clearly communicate registration deadlines and benefits.





Chapter 4 – Coding Camp Implementation

Building on the design decisions described in Chapter 2 and the materials and instruments presented in Chapter 3, this chapter provides a step-by-step implementation guide structured around a detailed camp timeline (Figure 7).

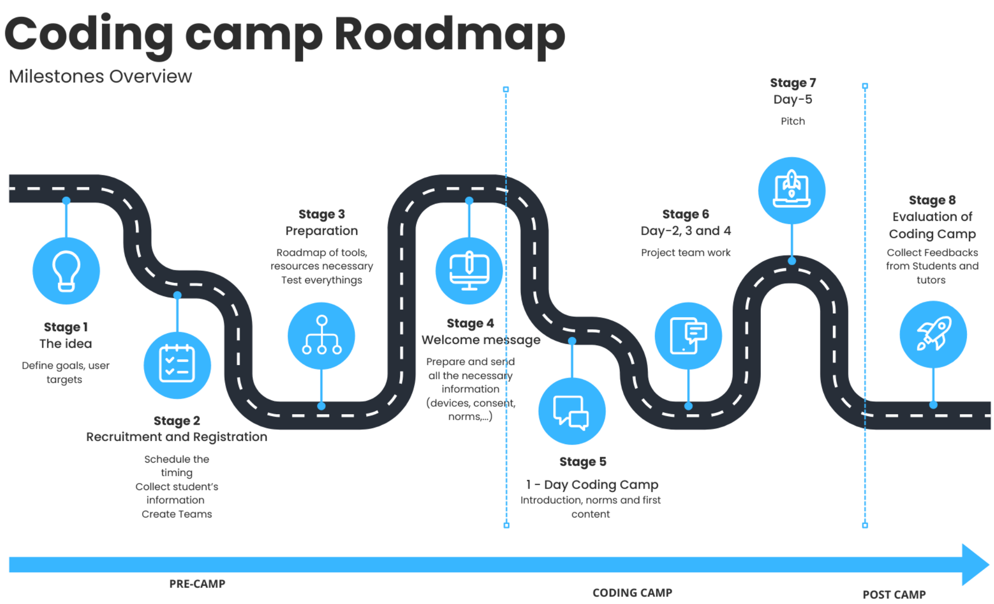


Figure 7. Coding camp roadmap

4.1 Before the coding camp

Stage 1: The idea

- Define the goals and the context of the coding camp. Clearly identify the main learning outcomes and the thematic focus of the camp (e.g., app development, serious games, extended reality).
- Defined the target users. Specify the intended audience in terms of school level, age group, and prior experience with programming or digital technologies.

Stage 2 : recruitment and registration

- Promote the camp through schools. Invitation letters or digital brochures can be sent directly to schools, school principals, and teachers. These materials should summarise the objectives of the camp, expected learning outcomes, organisational details, and the added value for students.

- **Finalize the schedule and delivery mode.** Confirm the dates, daily timetable, and whether the camp will be onsite, online, or hybrid. With hybrid participation, ensure that the format allows smooth collaboration between onsite and remote participants.
- **Complete the registration process.** Collect basic information about participants (e.g., background, gender, possible mobility limitations). This information helps organisers better understand the group composition and anticipate potential support needs.
- **Create teams** mixing students with different backgrounds, skills, and levels of experience. This supports peer learning and collaboration.
- **Define a code of conduct -** A Code of Conduct for participation should be defined, ensuring compliance with the applicable regulations and policies in force in the country, as well as with the rules and procedures of the school or hosting institution where the camps are implemented.

Stage 3: Preparation of tools and testing, set-up spaces

- **Verify access to all critical tools.** Ensure that the required platforms, programming environments, or collaboration tools are accessible to all participants, both onsite and online.
- **Test communication channels.** Confirm that participants can access the main communication channel (e.g., videoconferencing platform, chat tool, learning platform).
- **Prepare a backup communication channel.** Define an alternative channel in case of technical issues with the main platform.
- **Test tools on the intended devices and networks.** Verify compatibility with the devices used by participants and check that school networks or firewalls do not block required services.
- **Test file sharing and submission workflows.** Ensure that participants can easily upload and share files or project versions.
- **Set up the learning environment.** Prepare team spaces, shared resource folders, and repositories containing slides, exercises, templates, and learning materials.

Stage 4: Welcome message

- Send a welcome message to participants approximately one week before the camp. The message should:
  - Confirm dates, times, and venue (or online access details);
  - Recap technical requirements (software installation, accounts, equipment);
  - Outline the basic rules and code of conduct;
  - Provide a brief overview of the camp structure;
  - Include contact information for organisational support.

This step reduces last-minute uncertainties, reinforces commitment, and contributes to a smooth start of the activities.

4.2 During the coding camp

Day 1: before going-live checks

- Check that all sessions and meeting links are active and working.
- Verify that slides, exercises, and learning materials are available on the platform.
- Ensure that communication channels and team spaces are correctly configured.
- Prepare the materials and instructions for the first session of the camp.

Day 1 — Introduction, norms, team bonding, and challenge framing

- Welcome participants and introduce the goals of the week.
- Present the expected project outcomes and working norms.
- Run an **icebreaker or team-bonding activity**.
- Set up collaboration structures (teams, roles, communication channels).
- Introduce the challenge and expected project deliverables.

Monitoring:
Facilitators and peer tutors observe team discussions and progress. By the end of the day, each team should have defined the concept of its project.

- Presentation of the plan for Day 2

Day 2 — From concept to specification: roles, tasks, and tool onboarding

- Present the goals and expected outputs for the day.
- Introduce the development tools and environment.

- Present the key concepts required for the coding activities.
- Support teams in defining tasks, roles, and development plans.

Monitoring:

Teams should define a clear development plan for their project.

- Archive observations, activity results, submissions, and project releases.
- Presentation of the plan for Day 3.

Day 3 and Day 4 — Produce a working prototype

- Define the goals for Day3.
- Support **iterative project development**, encouraging testing and refinement.
- Facilitators and peer tutors monitor team progress and provide guidance when needed.

Monitoring:

Teams should progress toward the expected output (facilitator and peer tutor observation notes, project releases) -- Day 3: teams should demonstrate a first working prototype

- Archive: observations, results of activities, submissions, releases
- Presentation of the plan for the next day

Day 5 — Project closure and pitch

- Define the final goals for project completion.
- Support the final development and refinement of the project.
- Assist teams in preparing a **short presentation or pitch** of their work.
- Collect participants feedback

Expected outcome:

Each team delivers a final pitch and submits the completed project.

4.2.1 Troubleshooting guidelines:

1. **Authentication and access failures.**
   Participants cannot log into the videoconferencing tool or platform.
   Action
   Activate the backup communication channel and assign one staff member to support affected participants. If only a few participants are blocked, keep them in a support channel until access is restored.

2. **Loss of progress or corrupted project files**

   A team loses its project state, files are overwritten, or a critical asset breaks the build.

   Action

   Focus on recovering available assets rather than rebuilding the project from scratch. Suggest all teams to maintain periodic backups whenever possible.

3. **Uneven progress among teams**

   Some teams finish tasks early while others struggle.

   Action

   Prepare optional extensions, additional challenges, or creative enhancements to keep faster teams engaged while facilitators support teams that need more time.

4.3 After the coding camp

Learning assessment and follow-up

1. **Participant feedback.**

   Collect feedback using a short survey with a limited number of targeted questions focusing on the learning experience and organisation.
2. **Facilitator debrief**

   Organisers and facilitators should conduct a brief internal reflection to discuss strengths, challenges, and possible improvements for future editions.
3. **Archiving and reporting**

   Archive project outputs and documentation in a consistent structure. Prepare a short report including participation numbers, completion rates, and key implementation notes to support future coding camps.

4.4 Coding Camp Organizer Quick Checklist

Stage	Key Actions	Completed
The idea	Define the goals, learning outcomes, and thematic focus of the coding camp.	☐
	Identify the target participants (age group, school level, prerequisites).	☐
Recruitment	Prepare invitation materials (brochure, web page, communication for schools).	☐

	Share the invitation with schools, teachers, and potential participants.	☐
	Finalize the schedule and content delivering.	☐
Registration	Open and manage the registration process and collect baseline participant information.	☐
	Form heterogeneous teams.	☐
Preparation	Prepare slides, exercises, templates, and learning materials.	☐
	Ensure that required software tools and platforms are accessible to participants.	☐
	Test communication channels, collaboration tools, and file submission workflows.	☐
	Set up team spaces and shared resource areas.	☐
Pre-camp communication	Send a welcome message including schedule, technical requirements, and links.	☐
Day before the camp	Verify that meeting links, sessions, and materials are accessible.	☐
During the camp	Introduce goals, working norms, and project expectations.	☐
	Facilitate team bonding and collaboration activities.	☐
	Monitor team progress and provide guidance when needed.	☐
	Archive submissions, prototypes, and activity results.	☐
Closing the camp	Organise final project presentations and pitches.	☐
After the camp	Collect participant feedback through a short survey.	☐
	Conduct facilitator debrief and identify improvements.	☐
	Archive project outputs and prepare a short summary report.	☐

4.5 List of document templates

This section contains a set of template documents prepared for the Oscar use cases and pilots implemented during the first year of the project. As the project is ongoing, in the next iteration of the use case these templates will be made available in all the official languages of the project. All the templates are available for download in the following folder (https://scientificnet-my.sharepoint.com/:f:/g/personal/ilefronza_unibz_it/IgD2VAEwc484S7xscK7jSZ4VAVwrIFiEIlI7ViSKgPtORpI?e=Nw7xbD)

Pre-camp

1. Letter to Schools_Serious Game camp (ITA).pdf - Letter of invitation (Serious game camp version - Italian)
2. SeriousGame Regitration(ITA).pdf - Registration form (Serious game camp version – Italian)
3. XR Coding Camp – Poster.pdf - Poster for dissemination (XR camp version - Spanish)

4. Template content.xls - Template for planning each session of a coding camp
5. Template participants.xls - Template table to facilitate the registration of participants and the creation of teams.
6. Code of Codunct.pdf – Template of code of conduct.

Post-Camp

7. Questionario di valutazione – Serious Game camp.pdf – Questionnaire for feedback collection (Serious game camp version - Italian)
8. Test Piloto Spain-OSCAR.pdf – Questionnaire for feedback collection (XR game camp version - Spanish)

Co-funded by the European Union

Chapter 5 - Coding Camp Evaluation

5.1 Student assessment

Assessment should determine whether the intended learning objectives of the coding camp were achieved. To support assessment, each team should submit:

- a final project (or a link to it), clearly labeled with team name and version;
- a short project description explaining the problem addressed, the intended users, and the core idea/mechanic;
- the final pitch artifact, such as slides or a short pitch outline used during presentation.

Criterion	Group	Definition	Values / Notes
Educational objective	/	Presence of an educational objective in the serious game	**Yes:** learning objectives have been defined, even if they may be too generic or not ideal for the target audience. **No:** otherwise.
Target audience	/	Characteristics of the end users for whom the game was designed	**A:** age is specified, but skills/competences are not mentioned. **B:** both age and skills/competences are specified. **C:** neither age nor skills/competences are specified. **D:** age is not specified, but skills/competences are specified.
Storytelling	Game mechanics	Presence of a plot in the game	**Yes:** the game is set in a meaningful context that conveys a narrative structure, even if the plot is not explicitly written or detailed. **No:** otherwise.
Levels	Game mechanics	Presence of different levels in the game	**Yes:** at least two levels are defined. **No:** otherwise.
Scoring	Game mechanics	Presence of a clear progression mechanic in the game	**Yes:** a scoring system is adopted. It does not need to be complex, but it should provide players with feedback enabling them to understand whether an action is correct or incorrect. **No:** otherwise.
Characters	Game mechanics	Presence of one or more characters with different roles in the game	**Yes:** one or more characters have been defined; their roles may not be clearly defined within the game context. **No:** otherwise.

Context	Game mechanics	Presence of a clear game setting, including information about the type of game (e.g., card game, board game)	**Yes:** the game makes it easy to find information on how to use it, play it, and modify settings (e.g., mute audio). **No:** otherwise.
Skill/challenge balance	/	Appropriateness of challenges with respect to the expected skill levels of the target audience	**Yes:** the sequence of skills required to progress is appropriately structured across levels, ensuring that the player is never asked to apply a skill in a level before it has been properly introduced in a previous one. **No:** otherwise.
Suitability of missions for the educational objective	/	Appropriateness of tasks to ensure that players understand the game topics	**Yes:** game missions have been defined with the purpose of achieving the learning objectives. **No:** otherwise.
UI/Logic balance	/	Effort devoted to improving the user interface and the underlying logic	**Balanced:** both UI and logic were considered equally. **User interface:** UI improvement was prioritized while logic was neglected. **Logic:** logic includes advanced coding paradigms, structures, and blocks, but the UI is lacking.

5.2 Coding camp evaluation

Evaluation supports continuous improvement by helping organizers identify strengths, weaknesses, and areas for refinement in future editions of the camp. Moreover, it enables a meaningful connection with the school context by turning camp outcomes into actionable inputs for curriculum integration. Evaluation should investigate, for example:

- engagement and participation patterns across the week;
- how facilitation and organisational strategies supported progress most reliably, and under what conditions;
- equity and inclusion, to understand if all learners had meaningful opportunities to contribute.
- scalability and replicability across contexts, while maintaining quality.

5.2.1 Data sources

Evaluation should rely on a combination of evidence types collected during the camp, including:

- Observation (facilitators, peer-tutors)
- Participants' projects

- Attendance data

5.2.2 Data consolidation and reporting

Organizers should be able to confirm that all team projects are collected in the evaluation repository, that assessment results are exported when used, that observation sheets and peer tutor notes are consolidated, and that attendance or participation records are archived. At that point, the evaluation summary report and improvement action plan should be drafted, and at least one follow-up project proposal should be ready to share with schools. The report should be concise, evidence-driven, and oriented to action. It should focus on strengths, weaknesses, and improvement actions so that it can be translated into concrete revisions of templates, timelines, and facilitation strategies.

5.2.3 Follow-up projects with schools

The post-camp phase should also support curriculum integration by translating camp outcomes into follow-up activities that can be carried out with the schools attended by participants. These follow-ups help extend learning beyond the camp timeframe, reinforce both technical and transversal skills, and make non-conventional learning experiences more compatible with the constraints of formal education.