Encouraging early mastery of computational concepts through play

Dee, Hannah; Freixenet, Jordi; Cufi, Xavier; Muntaner Perich, Eduard; Poggioni, Valentina; Marian, Marius; Milani, Alfredo

Publication date:
2020

Citation for published version (APA):
Dee, H., Freixenet, J., Cufi, X., Muntaner Perich, E., Poggioni, V., Marian, M., & Milani, A. (2020). *Encouraging early mastery of computational concepts through play*. arXiv. https://arxiv.org/abs/2005.03930

Encouraging early mastery of computational concepts through play

H. M. Dee, J. Freixenet, X. Cufi, E. Muntaner Perich, V. Poggioni, M. Marian, A. Milani

***Abstract*—Learning to code, and more broadly, learning about computer science is a growing field of activity and research. Under the label of *computational thinking*, computational concepts are increasingly used as cognitive tools in many subject areas, beyond computer science. Using playful approaches and gamification to motivate educational activities, and to encourage exploratory learning is not a new idea since play has been involved in the learning of computational concepts by children from the very start. There is a tension however, between learning activities and opportunities that are completely open and playful, and learning activities that are structured enough to be easily replicable among contexts, countries and classrooms. This paper describes the conception, refinement, design and evaluation of a set of playful computational activities for classrooms or code clubs, that balance the benefits of playfulness with sufficient rigor and structure to enable robust replication.**

***Index Terms*—Play, Coding, Computer programming, Computational thinking, K-12.**

I. Introduction

This paper describes the development of a set of shared, ready-to-use activities that can be run in schools by teachers, in code-clubs by volunteers, or by university staff in schools as *outreach* activities. These activities are designed to promote and engage children with Computational Thinking [1]–[5], in a playful and open-ended way. All the activities have been tested and implemented by several teachers and/or academics, in several countries, and the workshop material is organized in order to emphasize opportunities to re-use and encourage context specific adaptation.

This paper opens with an analysis of the playful approach [6] to coding concepts, motivated by research into play and taxonomies of play. The content and thematic arrangement of the proposed activities is then described. Finally, the process used to generate and manage the workshop activities is presented by detailing the phases of design, refinement, test and editorial control. An earlier version of this work appeared in [7]: the work presented here has been extended to integrate taxonomies of play and playfulness within the proposed workshop activities, and to further explicitly clarify the *computational thinking* aspects of the workshops. The contributions of of this paper are:

- **To explicitly involve play in the learning process** through a clear analysis of taxonomies of play, and their relationship towards various computational thinking concepts and learning activities;
- **To share the process** through which playful workshops that engage school pupils in computational thinking can be iteratively refined and edited in order to maximize their reuse potential;
- **To show computational thinking concepts embedded in a set of cross-curricular activities**;
- **To encourage reuse and contextual adaptation of materials** through offering tested-in-the-field workshops to a broader computing education community.

Often a lot of excellent, playful and fun *outreach* work is trapped inside institutions. The proposed framework for activity representation, sharing and improvement presented here aims to shed light on these quality experiences. Through this we encourage colleagues to grow creatively and to engage with resources that will involve K-12 students in computing at all levels. A more long term goal is to encourage creative and passionate teachers to use this methodology for proposing, sharing and passing on their own original and creative contributions.

The proposed methodology for creating re-usable materials and workshops is shown in Figure 1. The main process can be summarized as:

1) Gather candidate playful workshop proposals;
2) Select a subset of workshops for further development according to our criteria;
3) Write selected candidates in draft form using a common structure providing *raw materials*;
4) Perform a preliminary paper-based review;
5) Revise to obtain *draft workshops*;
6) Test the workshops in schools, independently;
7) Revise them based upon school experience feedback creating thus *test workshops*;
8) Test them in closely observed conditions;
9) Finally revise, and release the workshops material to the web.

The proposed process emphasizes quality since it produces a workshop that has been thoroughly tested and refined in several situations – in the case of the project experience described here, several schools across several countries. In the following section, the local context for each of the participating countries is described briefly.

H. Dee is with the Dept. of Computer Science, Aberystwyth University, UK

J. Freixenet, X. Cufi and E. Muntaner Perich are with Udigital.edu, Universitat de Girona, Spain

V. Poggioni is with the Dept. of Mathematics and Computer Science University of Perugia, Italy

M. Marian is with the Dept. of Computers and Information Technology, University of Craiova, Romania

A. Milani is with the Dept. of Mathematics and Computer Science University of Perugia, Italy

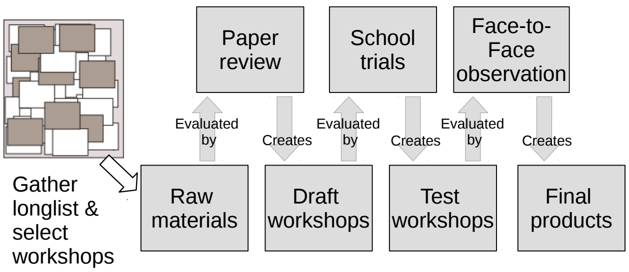

Fig. 1. An overview of the workshop creation process

II. The local context

The methodology and the experience here described is part of the outcome of a large-scale European Union project[1] involving participants from 5 countries, with four universities, two schools and a start-up company as core members. Workshops and activities have been tested across several additional contexts and events, including university visits, code clubs, science fairs and over 80 different schools. The diversity of the project participants (all European, but from different educational backgrounds, educational funding landscape, computational context, and regulatory systems) helped ensure a wide applicability of the resulting activities. In particular, the relationship between computer science and the school curriculum was not uniform across project teams, requiring a flexibility about computational context, classroom resources and realized experience. If the same activity works well in urban Spain, small-town Romania, and rural Wales we believe it has very broad appeal.

Across Europe, computing education has been changing, and the introduction of more computational thinking, practical computer science, coding and "informatics-type" activities is becoming a widespread curricular move [8]–[15]. The transition, frequently observed, ranges from educational systems in which computing enters the classroom thanks to a motivated teacher to organizations in which computing is a mandatory part of the curriculum [14]. This transition is rarely smooth. New subjects in the curriculum require new lesson plans, new teacher training, new approaches to learning and often new hardware or other types of equipment. In this context, university outreach efforts can become key sources of additional support for schools and colleges trying to keep up with the pace of curricular change.

In summary, IT was well-embedded across the curriculum for many of the project's participant countries. It is not unusual to see spreadsheets in History class, or word processing in English. Computer science and programming, however, were rarely seen outside computing classes. In this project, innovative ways have been proposed to foster creative and critical learning across the curriculum through the playful use of programming and robotics.

[1]Early Mastery project - http://playfulcoding.udg.edu/about/

III. Why playful?

Our commitment to providing creative and playful activities is based upon several ideas. First, there is a common belief that playing and games [6], particularly in the early stages of learning computing, can be a driver for progress. Exploration and self-directed learning are key to constructivist ideas [16], and the concept of playing as a "leading activity" comes earlier, from Vygotsky [17]. In this formulation, play represents a social construction that allows a child to move beyond their current ideas, developing new mental processes, *leading* to cognitive development.

This exploratory, playful approach has been part of learning to code for children since Papert's work on Logo [16]. Moving from exploration and open-ended discovery [18], to playful exploration and to active use of play in learning to code is one of the aims of the current work. It is important to avoid the suggestion that digital play is always educational: some educational software is playful but not open-ended, educational claims are often linked to marketing, and interactivity sometimes provides initial motivation but no further depth. As Stephen and Plowman state in [19], "*Digital interactivity alone does not guarantee either an educational or a playful encounter*".

These concerns lead us to our second key idea: it is useful to think more deeply about what it is meant by "playful" by further considering taxonomies of play, and the various ways in which play has been categorized by researchers in education and in child development [20], [21]. Considering the variety of different play types can greatly assist planning and structuring when creating playful coding experiences. Play categorizations and taxonomies include hierarchical systems, as are applied in [22] (where *epistemic* play is distinguished from *ludic* play, and then further subdivided), or observational. Whilst these have generally been developed with outdoor, physical play in mind, the applicability of these to the digital world is clear (as emphasized in [23]). These taxonomies of play types derive from extended observations of children playing in the real world or in virtual worlds, and break down the activities children engage in whilst playing into specific categories.

All the categories from [20] are applicable to the digital world in some sense [23], but in this work, a more concise taxonomy and a selection of subset of play types is proposed. The main reason for the restriction is that original general play categories are less applicable to computer-based activities (*rough and tumble play* and *locomotor play* for example rely on physical activity). The subset of play type for digital activities is presented in Table I and is confined to those types of play which can be found in the classroom within a computational learning context although others may occur in "CS-Unplugged" type activities [24], [25]. The list shows these play types in the approximate order of their popularity in a computing education context. Play is ubiquitous in computing, indeed "*Have a play with it*", meaning "try it out and see if you can work out what it does" is the first step for most expert users of technology presented with a new tool. This activity maps exactly on to what Hughes describes as *exploratory play*.

Through a consideration of these play taxonomies, is pos-

Play type	Brief description
Exploratory	Using the senses to explore and discover possibilities or find out information
Mastery	Play in which the players try to gain control of a skill or environment, maybe through practice
Symbolic	Using one object or item to stand in for another
Creative	Play that enables the player to develop ideas and make things
Communication	Using songs, rhymes, words and poetry in play
Dramatic	Dramatising events that the player has not been directly involved in
Imaginative	Play in which players pretend that things are otherwise
Object	The manipulation of objects and things through play
Role play	Play involving the adoption of different roles
Fantasy	Taking on roles which could not occur in real life e.g. superhero
Transgressive	Play which involves the player pushing at boundaries, for example, bending the rules of the game

TABLE I
A SUBSET OF PLAY TAXONOMIES WHICH CAN BE USED TO CONCEPTUALIZE DIGITAL ACTIVITY, ADAPTED FROM [20], [23]

sible to understand the progress in workshops in terms of types of play. Thus an important consideration [26] becomes whether children are able to move from exploratory play (asking "*what does this do?*") to imaginative or creative play (asking "*what can I do with this?*", or "*what can I make this do?*"). The category of play least likely to be found in the classroom context is that of transgressive play, or play which is crossing and stretching boundaries. The closest approximation of transgressive play that can be found in the proposed workshops, comes from robotics activities; for example, with some hardware configurations it is possible to make a robot "do a wheelie" by reversing direction rapidly. Once children discover this, they sometimes modify their code to deliberately cause and repeat the effect.

Thirdly, and perhaps most importantly, these workshops are designed to teach computational concepts, and also to encourage students (along with their teachers and parents) to see that computing can be *fun*. In a world in which many countries still do not have a formal computing curriculum, and in which others have a very formal and mathematical-oriented curriculum [27], [28], workshops emphasizing the creative and playful nature of computational skills have a key role to play, particularly in attracting a more diverse range of future computer scientists. We are not only computer science educators, we are also computer science evangelists.

Storytelling is something of a special case, and major potential as a workshop motivator. It brings together a particular subset of play types (creative, imaginative, communication, dramatic) and also involves the construction of an artifact (a story) [29]. But on top of that, there are a number of more general reasons for encouraging storytelling in the classroom:

- It increases the enthusiasm to read and re-read, because doing so allows to discover stories;
- It increases the motivation to write, as a means to express stories [30];
- It improves *soft skills*: the ability to listen, and to express ourselves publicly;
- Creating and telling stories allows participants to project and express their own emotions, feelings, and thoughts;
- Sharing stories helps participants put themselves in the role of others, developing empathy (and engaging in *role play*;

Digital storytelling is a relatively new term that refers to stories that include multimedia elements such as photographs, videos, sounds, texts, and also narrative voices, and this has found its way into the classroom in a number of contexts [31]. Robin in [32] provides a framework for thinking about digital storytelling in the classroom; authors have shown that digital storytelling can help visual memory [33], creativity [34], [35] and it can also improve academic achievement [36]. Creative computing, programming, or robotics fit this framework well. It is possible to program digital stories and animations with virtual characters and scenarios, but physical characters and scenarios can also be programmed in the form of robots or constructions outside the computer. Creative computing adds a new component to stories: *interactivity*.

IV. WORKSHOP SELECTION

The core of this project was the creation of a set of resources which could be used, re-used and modified by university staff engaged in outreach and collaborating with school teachers in search of innovative lesson plans. The starting point for this was a long-list of ideas and half-tested activities, written by the project partners. This long-list was cut down to a set of around 20, based upon inclusion criteria directly related to the main objectives of the project. All included workshops were addressing in some way, the following points:

- **Explore new ways to promote teaching and learning of computer programming in European schools**;
- **Help school children move from being digital consumers to digital creators**;
- **Make it easy for people to share the results of the project**;
- **Inspire schools to use computer programming in an interdisciplinary manner**;

In addition, the workshops were chosen for their *playful* nature: in fact, the project was nicknamed "Playful Coding", and the use of play within the activities was key to their inclusion. Thus, less emphasis was given to workshops with explicit, clear step-by-step instructions, and more interest and effort has been spent in workshops that allowed the participants time to experiment (exploratory play), repeat (mastery play), create and imagine (creative and imaginative play).

The second objective is explicitly linked to "maker" culture, constructivism [16] and computational creativity. The move from consumer to creator is one that many organizations are now championing, e.g. the World Economic Forum [37]. Workshops that lead to products or stories, such as animations, movies, games or applications have this enabling aspect. This emphasis on creativity and product implies that workshops were more likely selected if they would involve active learning methodologies [38], inviting students to create, design, modify and share. In these activities, technology is not an end in itself but merely a means to express creativity.

With regard to the third objective, there are some aspects of the global project that are vital in terms of sharing. The outputs are made accessible to the world, via a web platform [39]. Although the language of the platform is English, the Teachers' Guide [40] and complementary material have been translated into the languages of the project (English, Spanish, Romanian, French, Italian and Welsh). With regard to workshop selection, priority goes to workshops with minimal setup cost, free software, and open/platform neutral environments. The project includes participants using Apple, Microsoft and Linux environments; it is important to be able to support all platforms for maximum school applicability. Through encouraging reuse and contextual adaptation of the provided materials, teachers are encouraged to see how computing can inspire activities across the curriculum, leading themselves to develop further resources.

The fourth objective led to select examples of introduction of computer programming and robotics into school curricula in the broadest sense. Workshops in the final selection include a wide range of subjects including languages, poetry, geography, art, and many others across the curriculum.

V. Refinement of workshops

An extensive testing and refinement process (illustrated in Fig. 1) has been applied to the selected activities. This process consists of three sub-phases, resulting in an iterative improvement of the activities in terms of usability, structure, and presentation impact. Each partner recorded a summary of the workshops that they had proposed, providing the *raw materials* (initial workshop proposals). The *raw materials* were preliminarily subjected to a paper-based review by a project partner based in a different country (cultural context) to ensure cross-cultural applicability. Workshops were then revised based upon feedback to generate a *draft workshop* ready to become an in-class experiment. In the second phase, these *draft workshops* were evaluated in school trials and iteratively improved again, generating the *test workshops*. *Test workshop* proposals, already evaluated by several partners, were implemented in different countries with face-to-face observations, and reassessed in order to generate the *final products* in the last phase [41].

A. Paper review

Workshops were initially developed by each partner in close collaboration with their local community, including teachers, schools and after-school clubs. It has been observed that the first version of a workshop often carried the imprint of the local setting and the influence of national educational context. It has been also found that when designing playful and creative activities the input of several people can help build a richer experience. As an example, the first iteration of a workshop might involve some coding in Scratch [42] to make a quiz, and suggestions from the paper review might add asset creation (drawings, photos), swapping code to play with and test one another's work, and other game-like elements such as timers and high-score tables. Feedback was then integrated to generated the *draft workshop* revisions for actual in-the-field testing at a school.

B. Distributed testing: school trials

The next stage of activity refinement involved testing by the partners within the project consortium. Each partner was involved in organizing workshop trials with classes of local schools. It has been ensured, as much as possible, that each workshop was tested with classroom groups in at least two additional countries, beside the originating country. This allowed the project to provide workshop authors with feedback from outside the original social, national, economic and computational context. During trial sessions, the classroom teachers were asked to take notes, and upon completion of the trials, they were asked to fill in a simple web feedback questionnaire about the workshop. Only two workshops were difficult to test outside their home environment. These were both robotics workshops and the reason for difficulty was equipment availability in testing sites. The content of these workshops was approximated using alternative wheeled robots and modified accordingly.

The questions of the web teacher feedback questionnaire are shown in Figure II. As many other aspects of the project, the teacher feedback questionnaire underwent through several refinement iterations before stabilizing in the presented form. Some teachers are very keen to assist with the project, and will fill in a lot of detail. But other teachers, under strong time pressure, are less keen to provide detailed textual answers [28]. Therefore, the form evolved into a structure with predominantly yes/no questions, augmented with text boxes to enable the more willing teachers to fully contribute. Teachers were asked to complete the feedback form immediately, or soon after the workshop had been completed, and data was entered into the repository directly by means of Google Forms.

In earlier iterations, feedback was also collected from the children participating in the workshops; this proved very positive and confirmed the workshops were playful, interesting and fun for the children. However, the children's feedback was not as directly useful as teachers' feedback for further workshop improvements due to a lack of critical/negative comments.

Investigating the aggregate feedback from these school trials indicates that many teachers had some preliminary concerns when they first encounter the activity. Teachers' ratings (on a 1-5 scale) capture how easy they believe the activity is going to be to implement, how easy it was to implement, and an overall

Question	Answer
How much time did it take you to prepare for the activity?	[*Numeric answer*]
Please give a global mark for how easy it was to understand the activity before implementing it	[*1-5*]
How much time did it take you to implement the activity with kids?	[*Numeric answer*]
Please give a global mark for how easy it was to implement the activity	[*1-5*]
Do you think the proposed age range is adequate?	[*Yes/No*] [*Comments*]
Would you recommend this coding activity to other teachers/schools?	[*Yes/No*] [*Comments*]
Is the goal of the activity clear enough?	[*Yes/No*] [*Comments*]
Did you achieve the goal of the activity?	[*Yes/No*] [*Comments*]
Did the activity reach your expectations?	[*Yes/No*] [*Comments*]
Did kids enjoy the activity?	[*Yes/No*] [*Comments*]
Do you think kids have developed new skills while working on this activity?	[*Yes/No*] [*Comments*]
If yes, please, indicate which skills have they developed:	[checklist including *Heuristic method (Trial-Error); Communication Skills; Computational Thinking; Creative thinking; Problem solving ...*]
What do you think kids like the most/least?	[*Free text*]
What would you change in terms of description of the activity?	[*Free text*]
Please give a final, overall mark for the activity	[*1-5*]

TABLE II
A SUMMARY OF THE TEACHER FEEDBACK FORM FROM THE DISTRIBUTED TESTING PHASE. THESE QUESTIONS WERE PRECEDED BY A SERIES OF QUESTIONS ABOUT THE SCHOOL AND THE CHILDREN IN THE CLASSROOM (AGE RANGE, CLASS SIZE, PUBLIC/PRIVATE SCHOOL, AND SO ON)

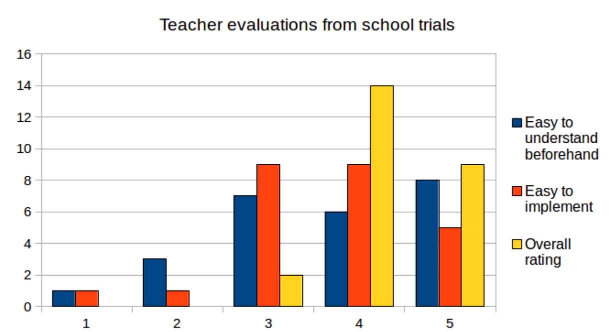

Fig. 2. Teacher judgements of ease of comprehension before implementation, ease of implementation, and overall rating after the workshops.

rating for the activity. Overall ratings were always higher, than those for representing teachers' perceptions of the workshop beforehand. This is probably not surprising, considering that the teachers involved in the project were often non-specialists. The chart in Figure 2 shows clearly this change of rating. All except one of the responses to the question "Would you recommend this coding activity to other teachers/schools?" were positive. The one negative response was associated with a workshop teaching Arduino and C programming for robotic control, which was thought to be too advanced for the class participants.

Feedback on specific workshops from these distributed school trials was passed back to the workshop originator, and the workshop description was then updated as required.

C. Face-to-face workshop observation

The final stage of workshop revision involved dedicated training events in which educators from different schools were asked to perform the activities with classes of pupils. This took place during several intense weeks of workshop evaluation in which the project as a whole visited schools in the UK and Spain, and hosted school visits in France. These sessions were observed by project partners from different institutions, taking notes on various aspects of the activity. Verbal feedback was also solicited, from the observers and the observed instructor, in a round-table feedback session immediately following each workshop. Figure 3 shows the observation criteria for the written aspect of this face-to-face evaluation.

As before, the observers were oriented to look explicitly for creative, collaborative work in a playful context, and for constructive workshop improvements. This process resulted in several pages of handwritten feedback for each workshop leader, with comments on all aspects of the learning experience from content type to classroom management. The feedback received from these sessions had much greater depth than that received from the earlier distributed in-school trials. The feedback was then incorporated into the workshop description, to provide our final iteration of improvements.

VI. CONTENT OF THE WORKSHOPS

The output of this lengthy refinement process has been a set of workshops that have computational thinking at their heart, and which combine different types of play in order to motivate and explore a variety of computational thinking concepts. In this section, some the proposed workshops are described in detail, and their possible concatenation and/or integration is presented.

Most activities are based on the use of Scratch programming language [43] in combination with other topics from the core curriculum in order to amplify the interest of both pupils and school teachers [42]. This interdisciplinary connection could be seen as taking advantage of the entertaining nature of Scratch for supporting the learning of specific topics from other subjects, but it could also be seen as taking advantage of students' interest in other subjects to encourage engagement with computing. The observation is that both directions can benefit from this situation: to quote Resnick, it is an opportunity to "learn to code and code to learn" [44].

TASK: To observe and detect those aspects you consider important to take into account for the evaluation of the activity. Please take as a guideline the following items for the analysis:
Methodological Aspects: Introduction, presentation of the activity, rhythm of the work, timing, types of intervention during the activity, types of groups (individual, in pairs, in groups . . .), time spent on the activity, reorientation of errors.
Interaction types: teacher-student, student-student, student-material.
Learning process: playful, trial and error, planned strategy, collaborative work.
Evaluation: follow-up work, collection of evidence and results, observation guidelines.
Classroom environment: number of students, materials, space distribution, noise, orderliness, stimulus.

Fig. 3. The classroom observation criteria for face-to-face testing sessions

The drive towards cross-curricular workshops led to an unconventional organization of our materials. Rather than classifying workshops strictly in one single category, "menus" of activities have been provided along particular themes. As with a restaurant menu, it is possible for one dish to appear in several different meals. In this way, it has been possible to highlight the interdisciplinary nature of created activities whilst still making the disciplinary connections explicit to potential users. Some examples of the provided workshop thematic menus are shown in the next sections.

Fig. 4. Participants "talking about the weather" with Scratch

A. *Geography menu*

Three workshops explore various geographical concepts in a playful way. The first workshop uses basic Scratch ideas to allow participants to talk about their country, town, or area. In this workshop, participants start with a map, which they can draw themselves if time allows. This map is then decorated with sprites (a typical Scratch entity) that represent parts of the area with pictures for each city, or street, or building. Choosing images to use as sprites that represent locations is a fun activity, and this can be made into a group task depending upon time. Sprites are then animated so that clicking on the sprites brings up information - for example, clicking on the sprite for "Cardiff" could bring up information about population, or sporting venues, or parks. . .

The second workshop is a coding activity with Scratch that aims to teach vocabulary items, especially about weather and free time activities. Images representing the four seasons are annotated with images for weather events, and textual descriptions of these. This workshop originated as a language teaching workshop for older students learning foreign language weather vocabulary, but can usefully be used with younger students in the participants' main language. Through this activity, participants learn about representations of weather, and illustrate changes in the seasons. This workshop is illustrated in Figure 4.

The final workshop further challenges the children, by moving beyond Scratch and introducing HTML. In this workshop, participants make a web page that can load a dynamic weather forecast, provided on-line. This can lead to a discussion about linking the world's geographical information through computing in order to provide news about their local area/town/country. For example, how does their region compare to others in the world? All of these three activities combine aspects of playful interaction with creativity and design; in all of them, participants are engaged with building something with software (a Scratch program or a web page) that helps convey geographical facts that are relevant to the students local area or town.

B. *Mathematics menu*

The *mathematics* thematic menu is specifically designed to motivate students to explore mathematical concepts in a playful way [27], [45]. Mathematical concepts are often implicitly embedded in the work rather than explicitly leading the activity. Mathematics can be demotivating for a subset of students, so activities aim to explore concepts like angles, sizes, proportions and other geometric concepts through animation, illustration and graphics. Participants are asked to draw, program, count, play, cut paper, measure, build things and animate. In a typical constructivist approach, through these activities, the participants learn about angles, shapes and geometry, while avoiding a potentially demotivating formal approach.

One example activity is the creation of a timer/chronometer with Scratch which has a dial with hands for seconds and minutes. The first step in this is to choose or create different programming sprites: a clock face, showing the seconds and minutes and two characters for the hands of the clock. Once chosen, these programming objects are linked to different sub-programs that will allow us to achieve other goals. The mathematical challenge underlying this activity is to discover the angle that is necessary to turn the clock hand each second or minute. Advanced participants can go on to build a digital chronometer, or to add hour hands, or "pause" and "reset" buttons.

Another activity exploring mathematical concepts starts with a playful photo-walk. Participants walk around the building and the surroundings (or just their classroom) with a camera, and take pictures of the regular geometrical shapes that they see in their day-to-day life (rectangular windows, circular traffic signals, etc.). Pupils then collect their photos on a computer with Scratch installed. In teams, they build a program with Scratch in which one of the images previously taken, forms a background. In the program, a character (sprite) has to follow the contour of the geometrical shape appearing in the picture. The same activity could be implemented with robots, for instance Lego Mindstorms, in which case the shapes the robot needs to trace are drawn and marked with colored tape on cardboard or the floor.

C. Storytelling menu

As mentioned in Section III, storytelling is a very important form of communication, and one of the clearest ways to bring creativity and playfulness into a computing context [46]. Several storytelling-based workshops have been created in the project; two examples of them are described here.

Collaborative digital storytelling with Scratch involves the participants working in teams to design and program a collaborative story. Each team programs one part of a global story in their own computer, then they synchronize the parts, and at the end of the activity all the computers are put together in a row, and the students can watch the full story where the characters move, speak and "jump" from one computer to the next.

At the beginning, the class decides on the topic and the main storyline. This begins with brainstorming and storyboards with paper and pencils. Then, each team works on creating the characters and backgrounds of their part of the story (with traditional or digital techniques), and pupils can even record their own voices for the dialogues. They upload these creations to a new Scratch project, and they start the programming the characters to move, speak and perform the required actions. This is the part of the activity that involves coding. For each section of the story, the characters should come in from one side of the stage, and leave through the other, consistently (so the story reads from left-to-right, for example).

After coding the entire story, the transitions between story sections need to be synchronized. The easy way to achieve this is to work out how long each animation section is, and then add an appropriate delay to the start of all programs except the first one. When participants press "go" concurrently, the program corresponding to a single story section will wait until its own turn. Finally, the computers are positioned to form a line; in this way, all children watch the final animation with the chained stories, and the characters will appear to jump from one screen to the next.

A very engaging further extension of this activity involves synchronizing the stories by using physical sensors and motors instead of using timers. The idea is to combine collaborative storytelling and chain reactions. One way to implement this challenge is by means of LEGO WeDo sensors and motors, which are compatible with Scratch. To make physical-world *contraptions* some physical-world materials are helpful: cardboard tubes, balls, tape, dominoes, sticks, etc.

Another example of story-related activity consists of creating an animated story from a poem that has been previously studied in class, or a poem that children have created. The idea is to combine images, e.g. drawn by children or downloaded from the Internet, with the verses of the poem. They have to appear sequentially, so the final result is a visual version of the poem that could also include recordings of their own voices reciting the verses.

Children then use Scratch to program images and texts to appear and disappear at the right time, following the rhythm of the language in the poem. This involves timing, animation, and an understanding of the links between poetry and imagery. This workshop again be adapted to the context of a foreign language class, but can also be used in the context of literature classes of the students main language.

D. Artificial intelligence

Whilst coding is clearly a core competence for computational thinking, computing is much more than just programming. The AI workshop considers more theoretical questions based around the Turing Test, and whether computers can think. This workshop proceeds through a mixture of game-playing and mini scientific experiments, discussing questions about what makes a thing intelligent, and how one knows that a thing is intelligent. The games include:

- **Text-message Turing Test** - in which one of the pupils leaves the room with a helper, and the class sends questions by SMS to guess which person is answering;
- **Intelligence ordering game** - in which a set of around 30 photographs of objects or organisms (a rock, a chess computer, a sheep, a kitten, a drone, a robot ...) are ordered by perceived intelligence by the group. This is followed by a discussion of what qualities make a thing intelligent; is it language? Sight? Being able to make friends?;
- **Can you trip up an AI?** - in which the pupils converse with a chatbot, competing to get it to make the stupidest answer.

The mini science experiment involves coming up with three questions for chatbots, and then trying these questions on different chatbots, recording the answers, and trying to decide which chatbot is more convincing. The participants are encouraged to think about what the question is testing, and how they expect the chatbot to respond, forming hypotheses about the chatbot's behaviour, and then testing these hypotheses.

E. Robotics and hardware related workshops

Robots are by their very nature motivating to some students [47]–[49]. Students love seeing their program having an effect in the real world: they can literally move things around with their ideas. However, robots can be temperamental, and if they break, there can be catastrophic effects for a workshop experience. That said, the motivational aspects far outweigh the risks. There is nothing more rewarding than seeing the sense of achievement on a student's face when they have finally managed to get the robot do what they wanted it to.

Several workshops involving robots have been tested and refined through our project, based upon three different hardware families (POB, Arduino-based robots, and Lego Mindstorms).

In the case of robots, schools will need to use what they have, as the cost of setup with new technology can be prohibitive. If schools have some other platform (that is, some robot other than Arduino, POB or Lego robots), the key ideas and concepts can be transferred to another wheeled robot. From a computational thinking perspective, the robot workshops deal with ideas of **control structures** (decisions, reacting to the environment), **iteration** (looping), **debugging** and other development related concepts, and perhaps most uniquely in child-focused workshops, **precision** and imprecision. Dealing with feedback loops, noisy sensors and real-world robots is a great learning experience from a computational perspective [50].

1) The general structure of a robot workshop: All of the projects' robot workshops have a similar overall structure, regardless of platform. This makes it easy to adapt the workshop structure for use with any suitable wheeled robot. The basic steps are listed below:

- **Move the robot** forward and backwards. Getting the robots moving early in the session is important for motivation, and will also expose any hardware faults at the outset;
- **Make the robot follow a path corresponding to a shape** for example, a square, showing that the controls entered into the robot's program can make it move more-or-less precisely. This also involves combining two types of command (move and turn), or involves moving one wheel and not another (depending upon the dynamic properties of the robot);
- **Modify your program so that the robot makes a different shape** for example a triangle, or a pentagon: this involves modification of the control code, and some calculation of angles. Advanced students at this point can be challenged to consider the problem of making a circular path;
- **Make the robot repeat this** e.g. creating *four* triangles;
- **Make the robot read a sensor** and then react to the value of that sensor, through printing to a terminal, or making the robot do something (like beeping or flashing);
- **Make the robot change behavior in response to a sensor** by avoiding an obstacle, or responding to a loud noise by changing the shape it draws.

The details of these steps can depend on the platform [39], but taken together they provide the opportunity to learn more about loops, control, precision, sensors, and debugging.

F. Packaging workshops for reuse

The workshops represent the core of the project, and the process just described has resulted in well-tested, standalone workshop packages and teaching material for the use of educators outside of the project. The single workshops are available on the project website [39], as individual pages or as part of thematic menus.

Workshops are also represented in the form of an extended book [40], translated into the languages of the project consortium members and available for free download from the web platform. Through the process of creating, testing, and refining the workshop descriptions, it has been possible to acquire significant experience and develop new ideas about computing, teaching, curriculum, classroom experiences and applied pedagogy. It is possible for someone to just pick up one of the packaged workshops and implement it immediately, independently from the others without looking at any of the surrounding materials. However, the book enhances this with links between the workshops, and relationships between themes which have emerged in the workshop. The book also contains sections on the learning environment, and also feedback (formative and summative) and assessment [51] tools.

VII. Outcome and results

During the two-year project, it has been possible to accomplish both tangible and intangible results, briefly:

- Building a catalog of inspiring playful coding activities that are integrated and available in the project web platform [39];
- A teacher's guide edited and published in book format (physical and digital) [40] that has been translated into 6 languages of the project consortium members. This includes the playful coding activities, explanation and discussion of the underlying pedagogical methodology, technical advice, ideas, suggestions and challenges;
- Establishing a process methodology for generating, testing, refining and managing the workshops and activities that potentially enable this kind of educational resources to live and grow over time. The methodology was developed by the consortium, and has been used during the creation and refinement of project activities.

More intangible outputs are represented by exchange of ideas and experiencing good practices, while conducting a deep inter-cultural dialogue. Firstly through the close collaboration and involvement of the core project partners. Secondly through the exchange diffusion of the refined and tested workshop packages and the sharing of good practices with schools and entities external to the project. The overall project impact currently includes more than 45 talks, seminars, training sessions for external educators, activities in more than 80 schools, with a total reach of more than 600 teachers and 4000 school-aged participants across five different countries.

The analysis of the feedback survey collected from teachers participating in the project shows that the playful approach was positively valued by our collaborators. The most common positive themes mentioned in the feedback, both from written questionnaires and from focus group meetings, was that the open-ended nature of creative workshops allow the whole class to contribute to the playful activity. Feedback analysis highlighted general positive impacts upon students, teachers and schools. Negative comments were rare. The only negative theme to remark upon was the teacher self-confidence bias: before seeing the activities in action, several teachers were wary of trying to implement them in class. Almost univesally, they later revise their opinion.

One example of feedback summarizing the project is the following quote from a Romanian high-school teacher:

"Children perceived the whole activity more as a game and less as a typical school activity." ... *"When we talked about Scratch, I told them it's meant to make Math or other school subjects easy to learn. I got their attention when I told them they can even draw or ride a bike, or fly a kite, whatever they want. The sky is the limit. In the end the students were amazed".*

The feedback from universities was also strong. Academics from all five participating countries have visited local schools to share these activities. These schools have used the workshops, at first with assistance from university and the project team, but in many cases they have gone on to adopt the activities independently regardless of country of origin. For instance, many months after the official end of the project, British schools still regularly run activities that were originated in Romania, Italy or Spain, and vice versa. Indeed the "Poetry animation with Scratch" workshop, which originated in Romania, has been run as an annual Welsh schools national contest, with tens of schools across Wales taking part 2017-2019.

In summary, the robustness of the design, testing and refining process has enabled activities from across the project to be implemented successfully in Romania, UK, France, Italy and Spain. When asking teachers and pupils about what they have learned, the kind of answers collected have a clear common theme – the use of play and creativity has unlocked the potential of computational thinking and coding across the curriculum.

VIII. Conclusions

In this paper a framework for transnational collaboration on teaching resources has been presented, that has enabled the production of a set of workshops encouraging computational thinking, and a guidebook for engaging schools to reproduce, adapt, share and contribute to extend the proposed activities. Ideas from computational thinking and learning through play have been integrated in a computational context. The three-phase workshop testing process approach (paper-based review, practical activity review in the classroom, and then face-to-face observed activity) has allowed the development and the delivery of high quality workshop activities with real robustness to situational variation. The process developed has enabled a very productive collaboration and resource sharing amongst a large number of transnational partners and external participants, despite different computational, cultural, economic and financial contexts.

Looking forward and beyond the project end: the teachers' guide book has been downloaded more than 4000 times across all six languages. Feedback on the book is also continuously collected. Continuing activity on the playful coding platform represents the final, tangible output of the two-year project, which will be improved and extended. The project partners are still contributing with designing and iteratively improving new activities, and by inviting new schools and teachers to join in testing and submitting activities for schools to use. This represents the foundation of an effective living library of methods and tools for engaging young minds with creative computing.

IX. Acknowledgments

We would like to thank all of the academics and teachers who directly supported this project by writing, testing, and feeding into our activities and their evaluation: Abir Zanzouri, Alexandrine Beya, Amanda Clare, Anna Ferrarons, Anna Rhys Davies, Anna Roura Rabionet, Antoanela Dan, Caterina Lombardi, Cioaca Norica, Dan Selisteanu, David Strubel, Delia Pirvu, Dorel Lupu, Elvira Popescu, Erin Good, Esther Villarroya, Eugen Ganea, Ferran Jambert, Guillaume Lemaitre, Gwenan Philips, Joan Massich, Laura Coravu, Mariona Niell, Mark Neal, Marta Peracaula, Martina Sabatini, Martin Nelmes, Meritxell Estebanell, Mireia Frigola, Mireia Pomar, Mohamed Belkacemi, Nigel Hardy, Ouadi Beya, Samantha Roberts, Tegid Owen, Tomi Rowlands and Wayne Aubrey. We would also like to thank all of the teachers and school children who tested and evaluated our workshops. This work was supported by **EU Erasmus+ 2014-1-ES01-KA201-004462**

References

[1] J. M. Wing, "Computational thinking and thinking about computing." *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 366(1881), pp. 3717–3725, 2008.

[2] ——, "Computational thinking benefits society," in *Social Issues in Computing New York: Academic Press 40th Anniversary Blog*. University of Toronto, 2014. [Online]. Available: http://socialissues.cs.toronto.edu/index.html%3Fp=279.html

[3] C. Brady, K. Orton, D. Weintrop, G. Anton, S. Rodriguez, and U. Wilensky, "All roads lead to computing: Making, participatory simulations, and social computing as pathways to computer science," *IEEE Transactions on Education*, vol. 60(1), pp. 59–66, 2017.

[4] M. S. Peteranetz, A. E. Flanigan, D. F. Shell, and L.-K. Soh, "Computational creativity exercises: An avenue for promoting learning in computer science," *IEEE Transactions on Education*, vol. 60(4), pp. 305–313, 2017.

[5] J. Lockwood and A. Mooney, "Computational thinking in secondary education: Where does it fit? a systematic literary review," *International Journal of Computer Science Education in Schools*, vol. 2, no. 1, 2018. [Online]. Available: https://doi.org/10.21585/ijcses.v2i1

[6] K. Seaborn and D. I. Fels, "Gamification in theory and action: A survey," *International Journal of Human-Computer Studies*, vol. 74, pp. 14 – 31, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1071581914001256

[7] H. Dee, X. Cufi, A. Milani, M. Marian, V. Poggioni, O. Aubreton, A. R. Rabionet, and T. Rowlands, "Playfully coding: Embedding computer science outreach in schools," in *Innovation and Technology in Computer Science Education (ITiCSE)*, 2017.

[8] "Le plan numérique pour l'Éducation," Ministère de l'Éducation Nationale, Tech. Rep., 2015, http://ecolenumerique.education.gouv.fr/plan-numerique-pour-l-education/.

[9] "Digital Competence Framework," in *Learning Wales*, 2016, http://learning.gov.wales/splash?orig=/resources/browse-all/digital-competence-framework/.

[10] S. Furber, "Shut down or restart? The way forward for computing in UK schools," The Royal Society, Tech. Rep., 2012, https://royalsociety.org/topics-policy/projects/computing-in-schools/report/.

[11] "Competències bàsiques de l'àmbit digital," Generalitat de Catalunya Departament d'Ensenyament, Tech. Rep., 2013, http://ensenyament.gencat.cat/web/.content/home/departament/publicacions/colleccions/competencies-basiques/primaria/prim-ambit-digital.pdf.

[12] "Piano Nazionale Scuola Digitale," 2015, http://www.istruzione.it/scuola_digitale/allegati/Materiali/pnsd-layout-30.10-WEB.pdf.

[13] "Programma il futuro (version updated to Dec. 2016)," MIUR, Ministero dell'Istruzione, dell'Universita e della Ricerca, Tech. Rep., 2016, http://programmailfuturo.it/media/docs/Descrizione-progetto-Programma-il-Futuro.pdf.

[14] S. Bocconi, A. Chioccariello, G. Dettori, A. Ferrari, K. Engelhardt, P. Kampylis, and Y. Punie, "Developing computational thinking in compulsory education," European Union, Tech. Rep., 2016, http://dx.doi.org/10.2791/792158.

[15] Monitorul oficial AL ROMÂNIEI, "Ordin privind aprobarea planurilor-cadru de învăţământ pentru învăţământul gimnazial," Romanian Ministry of National Education and Scientific Research, Tech. Rep. Part I, No. 446/15.VI.2016, 2016, http://www.edums.ro/OMENCS_3590_2016___planuri_cadru_gimnaziu.pdf.

[16] S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*. New York, NY, USA: Basic Books, Inc., 1980.

[17] L. S. Vygotsky, "Play and its role in the mental development of the child," *Soviet Psychology*, vol. 5(3), pp. 6–18, 1967.

[18] W. Burlson, D. B. Harlow, K. J. Nilsen, K. Perlin, N. Freed, C. Jensen, B. Lahey, P. Lu, and K. Muldner, "Active learning environments with robotic tangibles: Children's physical and virtual spatial programming experiences," *IEEE Transactions on Learning Technologies*, vol. PP(1), pp. 53–60, 2017.

[19] C. Stephen and L. Plowman, "Digital play," in *The SAGE handbook of Play and Learning in Early Childhood*. London: Sage, 2014, pp. 330–341.

[20] B. Hughes, *A Playworker's Taxonomy of Play Types, 2nd edition*. London: PlayLink, 2002.

[21] P. Else, *Making Sense Of Play: Supporting Children In Their Play*. London: McGraw-Hill Education, 2014.

[22] J. Bird and S. Edwards, "Children learning to use technologies through play: A digital play framework," *British Journal of Educational Technology*, vol. 46, no. 6, pp. 1149–1160, 2015. [Online]. Available: http://dx.doi.org/10.1111/bjet.12191

[23] J. Marsh, L. Plowman, D. Yamada-Rice, J. Bishop, and F. Scott, "Digital play: A new classification," *Early Years, An International Research Journal*, vol. 36(3), pp. 242–253, 2016.

[24] T. Nishida, S. Kanemune, Y. Idosaka, M. Namiki, T. Bell, and Y. Kuno, "A cs unplugged design pattern," *SIGCSE Bulletin*, vol. 41, no. 1, pp. 231–235, Mar. 2009. [Online]. Available: http://doi.acm.org/10.1145/1539024.1508951

[25] B. Tangney, E. Oldham, C. Conneely, S. Barrett, and J. Lawlor, "Pedagogy and processes for a computer programming outreach workshop - the bridge to college model," *IEEE Transactions on Education*, vol. 53(1), pp. 53–60, 2010.

[26] B. McLaren, S. Lim, and K. Koedinger, "When and how often should worked examples be given to students? new results and a summary of the current state of research," in *Proceedings of the 30th Annual Conference of Cognitive Science Society*, 2008, pp. 2176–2181.

[27] V. Aleven, B. M. McLaren, and J. Sewall, "Scaling up programming by demonstration for intelligent tutoring systems development: An open-access web site for middle school mathematics learning," *IEEE Transactions on Learning Technologies*, vol. 2(2), pp. 64–78, 2009.

[28] L. P. Prieto, J. I. Asensio-Prez, J. A. Muoz-Cristbal, Y. A. Dimitriadis, I. M. Jorrn-Abelln, and E. Gmez-Snchez, "Enabling teachers to deploy cscl designs across distributed learning environments," *IEEE Transactions on Learning Technologies*, vol. 6(4), pp. 324–336, 2013.

[29] Y. N. Harari, *Sapiens: A Brief History of Humankind*. Harvill Secker, 2014, ISBN: 978-0-099-59008-8.

[30] K. Kori, M. Pedaste, H. Altin, E. Tonisson, and T. Palts, "Factors that influence students' motivation to start and to continue studying information technology in estonia," *IEEE Transactions on Education*, vol. 59(4), pp. 255–262, 2016.

[31] N. Smeda, E. Dakich, and N. Sharda, "The effectiveness of digital storytelling in the classrooms: a comprehensive study," *Smart Learning Environments*, vol. 1, no. 1, p. 6, Dec 2014. [Online]. Available: https://doi.org/10.1186/s40561-014-0006-3

[32] B. R. Robin, "Digital storytelling: A powerful technology tool for the 21st century classroom," *Theory Into Practice*, vol. 47, pp. 220–228, 2008.

[33] H. Ç. Sarıca and Y. K. Usluel, "The effect of digital storytelling on visual memory and writing skills," *Journal of Computers and Education*, vol. 94, pp. 298–309, 2016.

[34] U. Spierling, D. Grasbon, N. Braun, and I. Iurge, "Setting the scene: playing digital director in interactive storytelling and creation," *Journal of Computers and Graphics*, vol. 26, pp. 31–44, 2002.

[35] P. Gresham, "Fostering creativity through digital storytelling: "It's a paradise inside a cage"," *Metaphor*, vol. 1, pp. 47–55, 2014.

[36] Y.-T. C. Yang and W.-C. I. Wu, "Digital storytelling for enhancing student academic achievement, critical thinking, and learning motivation: A year-long experimental study," *Journal of Computers and Education*, vol. 59, pp. 339–352, 2012.

[37] "From consumers to creators: Empowering the digital generation," World Economic Forum, Tech. Rep., 2012, http://www3.weforum.org/docs/WEF_ITTC_ConsumersCreatorsEmpoweringDigitalGeneration_Report_2012.pdf.

[38] R. K. Sawyer, Ed., *The Cambridge Handbook of Learning Sciences*. Cambridge: Cambridge University Press, 2006, ISBN: 978-0-521-84554-0.

[39] EarlyMastery, "Playful coding web site," 2016, http://playfulcoding.udg.edu/.

[40] ——, "Playful coding teachers' guide," 2016, http://playfulcoding.udg.edu/teacher-guide.

[41] J.-W. Strijbos, "Assessment of (computer-supported) collaborative learning," *IEEE Transactions on Learning Technologies*, vol. 4(1), pp. 59–73, 2011.

[42] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai, "Scratch: Programming for all," *Communications of the ACM*, vol. 52, no. 11, pp. 60–67, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1592761.1592779

[43] "Scratch programming platform," 2002, – http://scratch.mit.edu/, MIT Media Lab.

[44] M. Resnick, "Let's teach kids to code," in *TEDx*, 2012, https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code.

[45] T. Augustin, C. Hockemeyer, M. D. Kickmeier-Rust, and D. Albert, "Individualized skill assessment in digital learning games: Basic definitions and mathematical formalism," *IEEE Transactions on Learning Technologies*, vol. 4(2), pp. 138–148, 2011.

[46] B. M. Slator, C. Hill, and D. D. Val, "Teaching computer science with virtual worlds," *IEEE Transactions on Education*, vol. 47(2), pp. 269–275, 2004.

[47] J. R. del Solar, "Robotics-centered outreach activities: An integrated approach," *IEEE Transactions on Education*, vol. 52(1), pp. 38–45, 2009.

[48] T. L. A. Crenshaw, "Using robots and contract learning to teach cyber-physical systems to undergraduates," *IEEE Transactions on Education*, vol. 56(1), pp. 116–120, 2013.

[49] J. Shim, D. Kwon, and W. Lee, "The effects of a robot game environment on computer programming education for elementary school students," *IEEE Transactions on Education*, vol. 60(2), pp. 164–172, 2017.

[50] S. H. Kim and J. W. Jeon, "Introduction for freshmen to embedded systems using lego mindstorms," *IEEE Transactions on Education*, vol. 52(1), pp. 99–108, 2009.

[51] P. Black and D. William, "Assessment and classroom learning," *Assessment in Education*, vol. 5(1), pp. 7–74, 1998.