The Proceedings of the
13th International Conference on Game
Based Learning
ECGBL 2019

Hosted By
University of Southern Denmark
Odense, Denmark

3-4 October 2019

Edited by
Lars Elbæk, Gunver Majgaard,
Andrea Valente and Md. Saifuddin Khalid
Review Process
Papers submitted to this conference have been double-blind peer reviewed before final acceptance to the conference. Initially, abstracts were reviewed for relevance and accessibility and successful authors were invited to submit full papers. Many thanks to the reviewers who helped ensure the quality of all the submissions.

Ethics and Publication Malpractice Policy
ACPI adheres to a strict ethics and publication malpractice policy for all publications – details of which can be found here: http://www.academic-conferences.org/policies/ethics-policy-for-publishing-in-the-conference-proceedings-of-academic-conferences-and-publishing-international-limited/

Conference Proceedings
The Conference Proceedings is a book published with an ISBN and ISSN. The proceedings have been submitted to a number of accreditation, citation and indexing bodies including Thomson ISI Web of Science and Elsevier Scopus.

Author affiliation details in these proceedings have been reproduced as supplied by the authors themselves.

The Electronic version of the Conference Proceedings is available to download from DROPBOX https://tinyurl.com/FCCGBL19 Select Download and then Direct Download to access the Pdf file. Free download is available for conference participants for a period of 2 weeks after the conference.

The Conference Proceedings for this year and previous years can be purchased from http://academic-bookshop.com

E-Book ISSN: 2049-100X
Book Version ISSN: 2049-0992

Published by Academic Conferences and Publishing International Limited
Reading
UK
44-118-972-4148
www.academic-conferences.org
Game Making in Italian Primary Schools: The Neglected Actor

Laura Freina, Lucia Ferlino and Augusto Chiocciariello
Institute for Educational Technologies of the National Research Council, Genova, Italy
freina@itd.cnr.it
ferlino@itd.cnr.it
augusto@itd.cnr.it
DOI: 10.34190/GBL.19.068

Abstract: Students in primary schools engage in designing and developing games as a way to foster Computational Thinking (CT) skills through a constructionist approach. CT includes several 21st Century Skills such as critical thinking and problem-solving skills, it has positive effects on school results and is needed for a successful integration into our digital society. Focus is usually on STEM related subjects, suggesting that CT is fundamental for success in scientific areas; nevertheless, benefits from game making go well beyond this, including a much wider set of skills and school subjects. The process of game making and playing can be seen as a communication between three different actors: the developer, who is designing the game and coding the computer behaviour; the computer that interacts with the player; the player. The present paper, starting from the analysis of games created by grade 4 and 5 classes of an Italian primary school, as well as teacher training courses on coding in class and lessons organized by coding clubs, argues that while coding certainly is central to game making, little attention is payed to the human player. When the development of a game is carried out while keeping attention on the final player, several elements have to be considered: contents organization, player instructions, aesthetics, etc. The programming language becomes a means of communication, to be practiced within a social context including other developers and players, and widening the advantages of game making in schools. Language arts, visual communication skills and creativity are thus addressed, extending the range of basic skills for a successful integration in the 21st Century society. The paper reports a specific game making experience in primary schools, based in the online Scratch programming environment, which offers a rich context where projects can be shared, commented, and remixed: a perfect setting supporting communications. A preliminary analysis of the developed games is reported and suggestions to address the communicative aspects are given.

Keywords: computational thinking, problem solving, communication skills, game making, primary school education

1. Introduction

The term "21st century skills" refers to a wide set of skills, knowledge, work habits, and character traits that are believed to be fundamental for an active participation in today’s world. The Partnership for 21st Century Learning (P21) (Battelle for kids, n.d.), was established in the USA in 2002 with the aim of defining the basic skills needed in today’s society and promoting their uptake into formal education. The P21 organization identified deeper learning competencies and skills they called the “Four Cs of 21st Century learning”:

- Collaboration
- Communication
- Critical thinking and problem solving
- Creativity and innovation

Van Laar et al. (2017) analysed, through a systematic literature review, the 21st Century Skills and digital skills, defining a framework which includes communication, collaboration, creativity, critical thinking and problem solving.

Some of these abilities are addressed by the so-called Computational Thinking (CT). In 2006, Jeanette Wing first drew attention to the need in everyday life to be able to think computationally. She later defined CT as “the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent” (Wing, 2011).

At present, CT has not been uniquely defined. Román-González et al. (2017) analysed several different definitions trying to determine the cognitive abilities underlying CT. They reported six main concepts: logic, algorithms, decomposition, patterns, abstraction and evaluation. From their analysis, it stands out that even though CT is a wider concept than coding, some CT skills can be developed through a coding activity. Furthermore, one of the indications which derives from an analysis of the literature in the field (Kafai, 2017), is that a simple and amusing way to develop CT skills is through coding in a gaming environment.
Laura Freina, Lucia Ferlino and Augusto Chioccariello

With respect to the educational value of CT during the early school years, starting from Papert’s research (1980), it was acknowledged that coding could help to reify some of such concepts and can be a useful tool to learn how to think. This is especially true when situations in which to explore “ Powerful” ideas are created (such as, for example, differential geometry with the turtle micro world or feedback with Lego robots).

CT is therefore very important for future generations, and the introduction of CT in school curricula is currently one of the main topics on the educational policy agenda of many countries. In Italy there has been a discussion on the introduction of CT in compulsory schools (age 6-16) led by the Ministry of Education since 2014. In 2015, the Italian National Plan for Digital Schools was launched as part of a general school reform (law n.107/2015) (PNSD, 2015).

Several projects were organized, and some are still ongoing, with the aim of finding the best way to introduce coding into the standard school curricula and integrate it with the other subjects. Usually, the focus is on CT, addressing, among others, the skills of creativity, critical thinking and problem solving. Nevertheless, this is not the whole story!

Another important 21st Century skill is related to digital literacy (Van Laar et al., 2017). Ng analysed the concept of Digital literacy, recognizing “the ability to create meanings and communicate effectively with others through digital tools” (Ng, 2009, p.1067) as one of its components. As technologies are more and more present in our world, people have to learn to deal with them in a responsible and active manner. This means to be able not only to receive information and be able to evaluate and use it correctly, but also to create new information.

Kafai defines a similar concept as technological fluency: “not only knowing how to use new technological tools but also knowing how to make things of significance with those tools and most important, develop new ways of thinking based on use of those tools” (Kafai, 2006, p.39). Just as children in primary schools learn to read and write, they also learn to use digital technologies as receivers and to be active in producing content. The digital technologies become a way to communicate with other people.

One of the possible ways to acquire this is through digital games. Children often play with digital games, they are part of their world. The aim of our project was to introduce children to game making, not only as a way to foster their CT skills, but also to promote their communication abilities and digital literacy, and give them the needed cognitive tools to become a socially active member of our digital society.

In the present paper, we want to argue that digital game making activities in primary schools may have a positive effect on students not only with respect to CT, but also to communication skills and digital literacy, provided that the communicative aspects of game making are taken care of.

In section 2, the Jakobson communication model is briefly described and a possible adaptation to digital games is outlined. The project we are currently working on is described in section 3, along with several examples and some initial results. Section 4 focuses on the analysis of the different communication components and their influence on primary education, and, in section 5, brief suggestions on how to maximise the positive impact of game making on the overall school performance are mentioned.

1.1 A revised communication model

The Jakobson communication model is a good starting point to be adapted to digital games. Jakobson (1961) recognised six main components of any communication. Central to the communication process is the message, which is the content of the communication, which refers to a referent: the subject it is talking about. As an example, in the case of written communication, the message is the written text and the referent the event it is about. The sender is the person who sends the message, in our example the writer. The receiver is the person who gets the message, the reader. The message needs a channel through which it can go from the sender to the receiver (the piece of paper) and a code (the used written language). Finally, every communication takes place in a context, which includes all the linguistic, physical, cultural knowledge and information needed to understand the message (see Figure 1).
Figure 1: The Jacobson communication model

But what happens when this is referred to a digital game?

A digital game is usually built around a story, it needs to tell the player how to play, what the aim of the game is, when he wins or loses, how the score changes, etc. But often there may also be a deeper message, for example a serious game promoting the knowledge of the water cycle actually refers to it, and tries to communicate to its player some information about it. In this case, the referent would be the water cycle. A digital game can therefore be considered as a complex message that goes from its author to the final player.

In the Jakobson communication model, we can recognize all the previously mentioned components:

- The sender of the message is the author who creates the game. This is usually not one single person, but rather a group of professionals. In our case, the sender can be the single or group of students who design and develop their digital game.
- The receiver is the player who plays with the game.
- The context includes the technical knowledge needed to use digital tools to connect to Internet and find the game, and knowledge needed to understand how the game works and what its message is, including the language to understand instructions.
- The channel through which the game can reach the player includes all what the needed technology to be able to use the game: computer, tablet or smartphone, connection, etc.
- And finally the code, which, in this case, are the game mechanics, the aesthetics, the elements a game is composed of, etc.

In a written text, the code is the written form of the used language. The knowledge that the sender needs to have of the written language can be different from that of the reader: while the sender needs to encode his message, (usually with a pen, using ink, etc.), the reader has no need to be proficient in the use of these technologies. In a similar manner, a digital game has to be programmed, and the sender needs to know the programming language to encode his game, while the player does not need to have any knowledge of programing to play the game.

When the receiver reads a text, a complex activity, based on the reader’s knowledge, the context, the words, takes place through which the reader makes sense from the written text. Something very similar happens when the player is engaged playing the game: he interacts with the computer and, through this interaction, the meaning of the game is constructed.

Due to the complexity of the communication and the great difference required to code a game compared to playing it, we like to widen the original communication model inserting a third actor: the computer (Figure 2).

On one side, the programmers communicate with the computer by coding the digital game with the chosen programming language. This is interpreted by the computer who has to translate the higher level formal language used by the programmers in lower level machine friendly instructions. The communication can be considered bi-directional: the programmers code the software and then execute it. They get a feedback from the computer according to which they can evaluate and correct the program.
Figure 2: An adaptation of the communication model

On the other side, the player interacts with the computer. The game and the player continuously interact as the player makes his moves and the game changes accordingly. Again this is a bi-directional communication. The player is constructing a meaning through his interactions with the game.

Finally, there often is some other communication from the programmer to the player in terms of instructions, notes, credits that may be external to the game itself.

2. Making games in primary school

2.1 Coding and communication in primary classes

In today's digital world, most of our communication is mediated through digital tools. The new generations are usually consumers of interactive digital media through digital games, music, videos, etc. Learning to generate their own contents through coding allows them to express themselves fostering their creativity.

Coding can therefore be considered as a means of expression, a new language to be used, along with the traditional ones. Integrating coding activities in primary education is not done with the aim of forming future programmers, but to give students the possibility to express their thoughts and feelings, to communicate and share with others and to deepen the study of curricular subjects.

Furthermore, some students who may have specific problems with traditional texts (e.g. immigrants who do not master the teaching language) find, through coding, the possibility to actively participate in class activities defining their role in the class community.

2.2 Description of the project

The Italian parliament has recently voted in favour of the introduction of CT in the Italian primary education by 2022 (Mozione 1-00117, 12/03/2019). At present, this objective has not yet been formalized into a set of laws and rules. By focusing attention at the primary level, it is clear that coding is not directly aimed at forming future developers, but rather it should be closely integrated into the already existing disciplines. Furthermore, CT skills develop slowly and supporting actions need to last several years (Edwards, 2013). The definition of a curriculum spanning over the whole primary school is needed, and such a curriculum should closely integrate coding with the other activities in class.

A project aiming at the introduction of CT into the Italian Primary Schools mainly through coding activities was started two years ago. The project addresses all grades, from kindergarten to grade 5, offering different activities according to the age of the students, while seeking a close integration with the other school subjects. The younger children work on some basic concepts through a wide set of age appropriate activities of various kinds, some of which are unplugged, others use tablets. In the last three years of primary school (grades 3 to 5, ages
ranging from 8 to 11) students started using the online Scratch programming environment (Resnick et al., 2009) to build their own digital games and stories.

2.3 The programming environment

In the Italian primary schools, there are mainly two programming environment that are used and are appropriate for the target age range: Scratch (Moreno-León and Robles, 2016) and Code.org (code.org, n.d.).

Scratch is a playful visual programming environment, based on the use of blocks that can be “snapped” together like traditional building blocks, created specifically for children from the age of 8 to 14. The use of blocks avoids syntax errors making programming much easier than it would be with traditional text-based languages. The online version of Scratch is available anytime and students can continue their projects outside school hours. Furthermore, Scratch offers a controlled online social environment, where projects can be shared, other people’s project can be seen, liked, commented or followed. Every Scratch project has its own instructions, notes and credits, which are not coded within the program, but go along side it. In this way, Scratch offers a communication channel different from the program itself, fostering a “direct” communication between the authors and the player. Giving its users the possibility to reuse and remix existing projects, it promotes the creation of a social environment, empowering people to create projects that are more complex than they would have managed alone. As a consequence, cooperation and collaboration among Scratch users is fostered.

Code.org offers a set of lessons to teach incrementally the main programming concepts to children through a playful and user-friendly environment based on block programming. Moreover, it offers specialized programming environment for making games, animating a character, building an app, etc. A code.org user can share his creations, see other peoples’ projects and remix them, but no communication is possible between different users. Finally, it allows to see JavaScript version of a code written with visual programming blocks (JavaScript is one of the most used programming languages, see Figure 3), shifting the focus to more complex programming. According to our opinion, code.org is the perfect instrument to build the basis for future programmers, but, in this age range, we wanted to keep the focus on the use of game making as an expressive tool.

Creativity and learning are deeply social practices, and, in particular, learning to code can be compared to learning a new language: it requires a long practice and can best be achieved when the language is used within a community for purposeful interactions (Brennan and Resnick, 2012). The possibility to be an active member of a social environment is of fundamental importance according to our approach, therefore Scratch was chosen.

![Figure 3: Code.org: the same code in blocks and in JavaScript](image)

2.4 Phases and projects

At the beginning of the project, students had no previous programming experience, therefore a first phase of about three to four months was devoted to the introduction to coding concepts and Scratch. During this phase, one hour a week was devoted to coding and students were sometimes prompted with a new idea and then left free to explore. Some other time, specific suggestions were given according to their needs and requests.

As soon as students knew enough to create their own projects, even though they did not completely master Scratch, activities started to focus more on the expressive and communicative aspects. In the first year of the project, focus was kept on digital games as such (Freina et al., 2018), while in the second year a closer integration with the other school subjects was sought.

Students worked on their projects organized in small groups. This allowed including in the activities all the students of the class, also those who were less interested in coding, but were very valuable for the group bringing
their specific knowledge (sometimes a foreign language, other times some graphic abilities, etc.). Students worked collaboratively in pairs, often through pair programming (Salleh et al., 2011), and cooperatively with the rest of the group.

Links with the other subjects was managed by the class teacher, who decided the topics on which students had to create a project, generally with the aim of explaining it to someone else. For example, a grade 5 class decided to work on the geological risk in the city of Genova (Italy). Students worked in small groups, ranging from two to four members, and focused on a specific issue: weather alerts and their meaning, the correct behaviour in case of flood, garbage recycling to keep rivers clear, etc.

Each group was free to decide how to communicate their ideas. Most of them included in their project some sort of textual explanation, either written or recorded, followed by a quiz or a game to foster the previously presented concepts.

All projects were fully implemented and worked correctly, but some elements of the communication process were often forgotten. For example, a project based on garbage recycling did not take correctly care of the context. The project was based on a game, in which the player had to choose an object and drag it to the correct garbage bin (Figure 4). The basic idea being that garbage should be recycled correctly, but the images of the bins used were different from the real ones. For example, the real life glass recycling is green, but the shape is different from the one that was used. Students used the image of a generic garbage bin, which has the correct colour, but not the shape.

In this example, the authors did not consider that their ideal player may get a wrong message from the game, associating glass to the wrong bin. The authors did not consider the context in which the game could be played, and the wrong message that may arise from the use of the picture of another bin.

![Metti gli oggetti nel cassonetto giusto](image)

**Figure 4:** On the left the player is asked: “Put the garbage in the correct bin”, but the bin for glass is not present. On the right the correct bin for glass garbage

### 3. The neglected actor

Even though all projects were started with the idea of expressing some content to another person, many projects did not have instructions. Students tended to have difficulty in taking someone else's perspective, and, since they know well the game they developed, they tend to forget that the future player may not have the needed information to play it.

An initial analysis of some of the developed games was performed, starting from those that were shared by their author in the class folder. 225 Scratch projects from grades four and five classes were analysed. 126 of these projects (56%) did not use the “Instructions” area available in Scratch (in the top right part of Figure 5). Sometimes, this was not really a problem: some projects were short digital stories or animations, probably only small experiments, and did not require much interaction, but in other cases the player did not have any information on how to interact with the game. 43 projects were digital games, 15 of them did not have instructions (nearly 35%). In some other cases, instructions were present, but they were not complete, and described only some of the possible actions to the player. Overall, in 33 cases, the player had to access the code in order to understand the required interaction.
Figure 5: The project page of a game. There are no instructions, the player cannot know how to play.

In Figure 6, a game in which the player has no clue on what he should do to play is shown. It is only by accessing the code that he can understand that the octopus can be moved with the arrow keys and the aim of the game is to catch the star.

Figure 6: The movement of the octopus is managed by the arrow keys.

A lot of interventions that aim at the development of CT in children, tend to focus on the coding activities, and the awareness that the games could be used by others, and therefore may need to be accompanied with instructions, tends to be neglected.

Going back to Figure 2, most of the attention is on the relation between the authors and the computer, and the effort that the authors have to pay in order to make the computer behave as they wish. The receiver, who, in this case, is the player, is neglected.

CT is actually strictly linked to the coding activities, when the problem is analyses, decomposed into smaller steps, expressed in an algorithmic manner and finally executed by the computer. But in doing so, the authors should keep in mind that their project is being developed with the aim of being used by someone else. This means that they have to design the whole communication process that will take place between the future player and the program itself, including instructions on how to play, hints to help understanding the contents, credits for any part that was created by someone else, etc.

In doing so, the authors will use their natural language skills, they have to provide a clear visual communication, and use their creativity to design their project in the best possible manner.

While these elements may come natural to a language teacher with her class, they are often neglected as soon as the class moves to the computer lab. Here coding becomes the focus of the activities. Most teachers are not so proficient with programming, and teacher training courses focus mostly on coding issues.
Furthermore, many programming environments available for children, even though they do offer a visual programming language that helps students in their first approaches to coding, sometimes give the possibility to share project and see those made by others, they do not offer a complete social environment, comments and likes are not possible. This limits interaction between the authors and the players.

In order to spur students to consider, while programming their games, their future players, direct interaction with peers proved to be, in our case, the best solution. While simply reminding students that they need to write instructions seemed not to have a great effect, having peers play with their games, leave comments and likes in Scratch was very convincing.

During the project, several opportunities for students to experiment directly the effectiveness of their games for communication were organised. The developed games were presented to peers, and, in some cases, played by them, allowing their authors to see directly the interaction player-game.

During the first year of the project, a presentation at the end of the year was organized with the head of the school and the families. Before the formal meeting, all presentations were tested within the class and commented by peers. Students were particularly satisfied by their work, nevertheless, they were not forced to evaluate their games from an external player’s point of view, since all games were presented by their authors.

In the second year of the project, a game contest is planned. Games will be played directly by judges from another class. In order to prepare for the final event, intermediate meetings were organized, during which the games were presented to the whole class, and then discussed together. Suggestions from peers were collected by group members and taken care of. This approach is proving to be more fruitful: students show that they are aware that instructions are needed to play with someone else’s project. Furthermore, issues related to the contents of the projects were discussed and suggestions were given to communicate the contents better. The final contest has not yet taken place at the time this paper is written, but initial results can be seen from the intermediate meetings: students are more aware that they need to help their future players understand their game.

4. Conclusions

This paper presents a project in which CT was introduced into an Italian Primary school through coding activities, aiming at its close integration with the other school subjects. Particular attention was paid to communication.

Coding was focused on giving students the possibility to use digital devices to express themselves, fostering their creativity. With this view, students discovered a means of communication, based on different skills than the traditional text. Students, at this age, live in a world full of images and music, and they like the possibility to assemble these, adding some interactivity to tell their stories. Games too are part of their everyday life, they know and play many different games, and are aware of which characteristics make a game. Multimedia stories and digital games are a natural way for them to express their creativity.

In the project, the online Scratch programming environment was chosen because it offers a block-based visual programming language, along with a social environment where programmers from all over the world can exchange opinions, help each other, find inspirations for their projects (Resnick and Robinson, 2017).

Nevertheless, students, while developing digital games, tended not to take into account the player, they often forgot to take the player’s perspective and took for granted what they know about their game. In our experience, most of the coding activities in schools are focused on CT and coding, and the player is frequently forgotten.

We argue that, if the communication aspects were adequately supported by helping students not to forget the final player, the positive impact of coding activities would be much wider. Language arts, visual communication skills and creativity would be addressed, along with CT skills, supporting the development and consolidation of a wider part of the 21st Century skills needed for a successful integration in a digital society. Furthermore, group work and interaction with a broader audience through Scratch, would give students the possibility to practice collaboration and cooperation skills towards a common goal.
References


