Design, Implementation and Evaluation of a Computer Science Teacher Training Programme for Learning and Teaching of Python Inside and Outside School

Establishing and Supporting Code Clubs to Learn Computer Programming by Self-Contained Examples

Nektarios Moumoutzis[1], George Boukeas[2], Vassilis Vassilakis[2],
Nikos Pappas[1], Chara Xanthaki[2], Ioannis Maragkoudakis[1],
Antonios Deligiannakis[1], Stavros Christodoulakis[1]

[1] Laboratory of Distributed Multimedia Information Systems and Applications,
School of Electrical and Computer Engineering, Technical University of Crete, Greece
[2] Computer Science Teacher in Secondary Education, Greece

nektar@ced.tuc.gr, boukeas@gmail.com, vassilakis1978@gmail.com,
nikos@ced.tuc.gr, chara.xanthaki@gmail.com, imarag@ced.tuc.gr,
adeli@ced.tuc.gr, stavros@ced.tuc.gr

Abstract. We present the design, implementation and evaluation of a training programme for Computer Science teachers on the educational use of the Python programming language inside and outside school. The programme used educational resources centered on meaningful, self-contained programming projects. The training programme followed a blended-learning approach thus offering an opportunity to many computer science teachers make their first steps towards the educational use of the Python language within a very promising learner-centered pedagogical framework. Using initially an online course made it possible to reach a much greater number of Computer Science teachers, especially those living in remote areas and through them, have a considerable impact on students through the subsequent establishment of local code clubs. The synchronous interaction with the course facilitators during the monthly sessions, the forming of local groups and the systematic communication through the learning platform used, made it possible to alleviate many of the disadvantages usually linked with online courses as it is evident from the evaluation results.

Keywords: Python programming, computer science in secondary education, code clubs.

1 Introduction

Teaching introductory programming courses has received much attention the last years. This is mainly due to the ubiquitous use of computers, the proliferation of the so called cultures of participation [1], end-user programming [2] and end-user software engineering [3]. These trends are addressing software tools that provide power-

adfa, p. 1, 2011.

ful scripting languages to enable flexible customization and rich interactive content development by end-users. In this respect, knowledge of computer programming concepts is nowadays necessary for most knowledge workers including scientists and engineers. Consequently, many higher education departments have included introductory programming courses in their curricula [4]. Furthermore, many countries extend their curricula in secondary or even primary education to address the development of basic programming skills [5]. The importance of computer programming has received even more attention through computer coding campaigns such as the *The Hour of Code* and *Europe Code Week*. Informal learning opportunities are also offered in many countries following the organizational approach of coding clubs [6].

Following the above trends, Python has recently been introduced in vocational training curricula (professional lyceums) as well as in upper secondary education in Greece. However, many Computer Science teachers are not familiar with Python and all available professional development opportunities (mostly short webinars) are in high demand and overbooked. To address the need of enabling secondary Computer Science teachers in Greece to get familiar with the Python programming language and adopt effective learner-centered pedagogies, a 7-month teacher training programme was designed and implemented with partial funding from the Google CS4HS initiative. This programme is described here along with its evaluation. It was offered in a blended-learning fashion starting with a 3-month distant learning phase to study the Python language through self-contained programming projects and a subsequent 3-month phase with face-to-face collaboration focusing on the establishment of local code clubs. The programme finished with an evaluation phase of one month duration.

The rest of this paper is organized as follows: Section 2 presents previous experience in organization of training programmes to enable teachers enrich their teaching practices and use engaging learning scenarios for promoting coding at schools. Section 3 presents very briefly the training portal that we used to support distant training services while section 4 focuses on the training programme philosophy and structure combining distant training with local collaboration thus creating a blended learning framework. Section 5 presents the methodology and the results of the evaluation Section 6 concludes and presents future plans.

2 Previous steps on promoting coding in schools within a creative learning context

An initial exploration on the potential of programming in education from the perspective of providing opportunities for students in primary and secondary education develop their own coding projects for learning was undertaken during the pSkills project (October 2009 – September 2011) [7]. The target was to build a community of primary and secondary teachers, mainly computer science teachers, to exploit modern educational programming languages in their courses. The approach adopted could be described as a teacher training framework rather than a learning framework. This proved to be rather limiting in exploiting the full potential of the approach. However, the fact that most of the participants were computer science teachers made it possible for them to proceed very quickly with the appropriation of the underlying technologies and enter the classrooms successfully. Other teacher specialties faced significant problems and were unable to do the same.

The pSkills teacher training activities were structured as a three-step process: (a) initial training and community building workshops with over 400 participants in total; (b) pilot workshops, one in each one of the four participating countries with over 40 participants in Greece, Austria, Italy and Estonia; and (c) the pSkills Summer School, a one-week intensive training event with 10 participants. In all these phases, a training portal was used to host all materials and provide communication between the participants before and after the events, thus providing the basis of a blended-learning approach. Though its training activities, the project raised awareness and provided insight and inspiration through indicative learning scenarios targeting courses in primary and secondary education. Furthermore, it offered materials to enable teachers guide their students through an engaging learning process during which they develop their own coding projects.

The focus was on developing digital games using an appropriated educational programming language [8]. This process starts from the inception of a game employing brainstorming. The game design is facilitated by storyboards while testing by evaluation rubrics. Game development was based on Scratch [9] and game distribution on the Scratch community site. The offered materials included: (a) Brainstorming guidelines and selected Scratch projects to provide inspiration; (b) Storyboarding templates in the form of slide presentations; (c) Game skeletons in Scratch along with introductory hands-on tutorials to enable game development; (d) Worksheets for step-by-step development of simple games and ideas for their extension; (e) Rubrics for peer assessment containing criteria regarding playability, usability and user experience qualities; and (f) Links to Scratch community site pages with important information and additional links to related resources on the web.

The ultimate goal of the pSkills project was to foster a systemic change beyond the mainstream focus on the so-called IT literacy towards IT fluency. IT literacy is linked to surface technical skills related to office automation applications (word processing, spreadsheets, presentations etc.) and communication tools (email, web browsers etc.). IT fluency focuses on sufficiently foundational material with “staying power” to promote understanding of computers and their applications and the ability to fully exploit the potential of modern systems and computer applications through programming.

Towards the end of the implementation period of the project, an explosion of interest of computer science teachers on its themes was observed. Over 250 teachers participated in the local workshops held in Crete, Greece. Most of them started to apply the ideas into the classrooms and, more importantly, within the context of creative projects with their students. Following a viral pattern, the use of Scratch was soon spread in other schools and the student projects developed were enough to justify a first attempt to organize a Student's Digital Creativity Fest in 2011. Starting from Crete, this annual event has now reached most of the Greek regions with more than 7000 participating students, around 1000 teachers and more than 400 schools. These developments had a notable contribution in firing a positive change in the national education system: In the new curricula for computer science at secondary education and the supporting textbooks, there is now an explicit focus on game design, game development and the use of appropriate platforms such as Scratch and App Inventor. This observed impact justifies our decision to adopt a bottom-up approach and provides evidence that small changes in everyday learning activities in the classrooms can trigger systemic changes as well.

Following the successful impact of the pSkills project, our focus expanded beyond computer science education to include other domains including non-formal and informal education. The opportunity for this was given within the context of the ALICE project [10] and its decision to include games as a creative language to be adopted by adult trainers along with Music, Digital Storytelling, and Children Narratives. ALICE targeted adult trainers with the aim to enable them design and implement intergenerational creative learning environments. The project designed and piloted a graduate programme for educators in Greece, Italy, Switzerland, UK and Romania. The programme was offered through a learning portal and included face-to-face sessions as well in each one of the five participating countries, thus adopting a blended-learning approach. The training consisted of six learning units and a final project. One of these learning units targeted digital games and a number of participants adopted games as a creative language to work on their individual projects during which they designed and implemented learning activities where adults and children they were invited to learn together in creative ways. The term gaming-literacy was adopted to signify a step towards the accommodation of a new way of thinking, working, collaborating, teaching and learning, as initially introduced by Zimmerman [11].

The gaming literacy learning framework offered the opportunity to the ALICE participants to address topics on three levels: (1) Understanding and evaluating games through critical analysis that promotes the acquisition of the language pertaining to technology, genres, values, stereotypes, production processes of games and their learning value. (2) Critical consumption through reflection on gamers' behavior in order to better exploit free time, foster learning and enrich human relationships. Time spent for video game playing, game preferences, social aspects of game play, type of entertainment and learning offered are issues related to this critical self-reflection. (3) Crafting digital artifacts using modern tools that enable non-technical people to invent their own video games and be engaged in their realization by creating rules, characters, narratives, graphics, audio, and animations.

The work reported in this paper is a third step towards supporting educators on the integration of new technological tools in their teaching practices emphasizing on digital skills and creative learning. This time, the focus is on general purpose programming language, not an educational or creative language as it was the case in the pSkills and the ALICE projects. In particular, the programming language adopted in this case is Python and its use as a first programming language for novices taking into account the parallel developments in the school curricula in Greece and the need to support Geek computer science teachers on their use of the Python language within a pedagogical framework that promotes active learning within engaging learning scenarios in coding clubs. Selecting coding clubs as the pedagogical framework of our approach incorporates the positive prior experience on promoting coding in formal, informal and non-formal learning settings in pSkills and ALICE projects.

3 The training portal

The training portal used for setting up the blended learning framework of the training programme is the evolution of MOLE, a multimedia online learning environment [12] that was initially developed to support educational activities in university departments. MOLE integrated tools and services for educational material reuse in an interoperable manner [13]. After its successful adoption in the academic environment, it has been adapted and enhanced under the name *Coursevo* to support professional development and training within a context that enables the establishment and sustainable operation of Communities of Practice (CoP) [14].

Coursevo enables communication between tutors/trainers and trainees, cooperation among trainees and access to coursework information and learning resources. It can combine traditional classroom-based lessons and practical sessions, with self-study and eLearning. Coursevo platform hides the complexity and frees the trainers from tedious system maintenance tasks, since a course or even a full functional learning site can be created in a few steps following the SaaS (Software as a Service) paradigm

Coursevo integrates BigBlueButton (http://bigbluebutton.org/) to enable video teleconferencing. This proved very important for the implementation of the programme: One synchronous teleconference was organized each month to give guidance to participants, present best practices and examples, answer to questions and solve practical problems, especially for the organization of local code clubs. Furthermore, the workgroup support offered by Coursevo was used and appropriately adapted to enable the coordination of work in each group (code club organizers) and facilitate the development and submission of assignments.

4 The training programme

The design of the training programme was based on the principles of social constructivism: Initially, participants explored the course material and develop Python skills, as they delved into the programming projects and tackled the assignments. The assignments, meant to induce structure on the learning process, were implemented in groups exploiting the special features of the underlying learning platform. There were no lectures on Python programming. The course facilitators served as peer advisors, guides and coordinators. Following this initial phase of getting familiar with the Python language and the proposed pedagogical methodology, participants were asked to apply the knowledge and skills they have acquired, in workshops or coding clubs for their students, exploiting scenario-based pedagogical approaches. In this context, they eventually composed their own training material and developed strategies for cooperating with other participants from their regional group, learn from and support each other. There was also a strong element of reflection, self- and peer-evaluation at the final phase of the project. As already stated, an important aspect of the training programme was its blended-learning approach to promote collaboration among computer science teachers in many locations parts of Greece including several remote areas.

The program consisted of three phases. In the first phase (3 months) participants studied, explored and evaluated the course material, familiarizing themselves with Python and the programming projects approach. In the second phase (3 months) participants implemented the course material in coding clubs. The final phase (1 month) involved extensive reflection and evaluation of the course.

The kick off was done via a teleconference that presented the overall structure and objectives of the training programme. Every month there was a live online session with the course facilitators, where participant groups had the opportunity to make presentations or engage in structured discussions. During the course, participants also communicated via discussion forums, online chat rooms and videoconferencing facilities offered by the Coursevo platform. Participants worked together in regional groups and posted their assignments online, each group creating a portfolio that was reviewed by their peers. All results were documented and shared in the form of adaptable learning scenarios (i.e. project-based scenarios and/or lesson plans) that referenced teaching objectives of the Greek CS curricula and were organized in a digital repository that is available to all CS teachers through a Creative Common license for further reuse after the end of the project.

5 Evaluation of the training programme

The evaluation of the training programme was based on three complementary elements: (1) a detailed questionnaire for the participating computer science teachers; (2) an initial and a final questionnaire targeting the students participating in code clubs; (3) a self-evaluation report for each one of the established code clubs that was prepared by the organizers, i.e. the computer science teachers participating in the training programme that collaborated in the design and implementation of each code club.

Teacher evaluation of the training

The teachers' questionnaire contained 4 sections: (1) Demographic data; (2) Likert scale questions for the evaluation of the training programme; (3) Likert scale questions for personal and group evaluation; (3) Open-ended questions regarding the objectives of each participant and the degree of satisfaction as well as on the strong and weak points of the training programme. The table below summarizes the socio-demographic and professional profile of the participating teachers. There was equilibrium in gender while the vast majority of the participants were within the most active age-band for teaching (ages 36-55 sum up to more than 86% of the participants). Their distribution in terms of basic degrees, resonates with the overall distribution in computer science teacher community as a whole. Finally, their level of studies shows a high percentage of participants with postgraduate degrees, which is true for the computer science teacher community as a whole in Greece as well.

Table 1. Socio-demographic and profesional profile of the participating teachers (N=80)

Variable	N	(%)

Gender		
Male	40	(50.0)
Female	40	(50.0)
Age band		
<=25	1	(01.3)
26-35	4	(05.0)
36-45	35	(43.8)
46-55	35	(43.8)
>=56	5	(06.3)
Specialization		
University graduate in Computer Science	61	(76.3)
Technological institution graduate in Computer Science	17	(21.3)
Other	2	(02.5)
Level of studies		
Bachelor	33	(41.3)
Masters	43	(53.8)
PhD	4	(05.0)

By the analysis of the answers to the teacher evaluation questionnaire (N=80) it is evident that the official goals of the training programme, as listed in section 4, were met (Table 2 below). The training programme provided an opportunity for the participating teachers to make their first steps on the educational use of the Python language within a promising pedagogical framework. Using an online course made it possible to reach a much greater number of teachers (especially those living in remote areas) and, through them, have a considerable impact on students through the establishment of code clubs. The synchronous interaction with the course facilitators during the monthly sessions, the forming of regional groups and the systematic communication through the forums helped alleviate many of the disadvantages usually linked with online courses as it is evident from the analysis of the results from the teachers' questionnaires and the self-evaluation reports of the code clubs. The three-phase structure of the course enabled participants to gain practical, hands-on experience both while learning to program in Python themselves and while using it to teach programming to their students. The implementation of the course material by the participants ensured that the knowledge and skills acquired can be transferred into the classroom and put to practical use.

The coding club approach adopted for engaging students in programming projects was an important aspect of the training programme. It constituted its pedagogical background that differentiated it from other initiatives for computer science teacher professional development where training is usually decoupled from a certain student learning model. Coding clubs are ideal for implementing social constructivism and related pedagogical ideas that are considered very important for cultivating student initiative, creativity and innovative thinking. Students learn by creating things and teachers have a direct experience of organizing such structures and sustaining them.

Table 2. Evaluation of the training programme as a whole (N=80)

Variable	1	2	3	4	5	mean
Training progr. success with respect to its initial objectives						
Familiarize with the basic characteristics of Python	0	1	4	44	31	**4.31**
Use self-contained examples in teaching programming	0	1	3	41	35	**4.38**
Enable sustainable local communities of practice	0	3	18	33	26	**4.03**
Evaluation of the learning material						
The learning material can be used without modification	2	5	16	45	12	**3,75**
Autonomous study of learning mat. is easy for novices	4	11	25	25	15	**3,45**
Introduces basic Python concepts in an engaging way	1	2	15	39	23	**4.01**
The content of the learning material is easy to understand	1	3	7	44	25	**4.11**
The learning material is attractive and appealing	0	3	11	40	26	**4.11**
Evaluation of the trainers						
They had adequate knowledge to support trainees	0	0	1	20	59	**4.73**
They effectively transmit their knowledge	0	0	6	28	46	**4.50**
They were supportive in the practical exercises	0	1	10	24	45	**4.41**
They responded to questions	0	0	4	16	60	**4.70**
They were adequaly prepared	0	0	2	18	60	**4.73**
Evaluation of teleconferences						
Presentations were adequate and complementary	0	1	13	40	26	**4.14**
Presentations and discussion addressed my needs	1	9	24	31	15	**3.63**
Teleconferences triggered reflection and discussion	0	2	11	40	27	**4.15**
Teleconferences gave me interesting ideas	0	4	15	41	20	**3.96**
Teleconferences stimulated further study	0	3	24	37	16	**3.83**
Evaluation of the organization and workload of exercises						
The material was adequate for doing the exercises	0	2	4	44	30	**4.23**
There was adequate support for finishing the exercises	0	1	8	43	28	**4.23**
Adequate time was given for doing the exercises	1	13	8	35	23	**3.83**
Evaluation of the training platform (Coursevo)						
The platform is easy to use	0	9	15	40	16	**3.79**
The platform has adequate functionality	0	9	11	46	14	**3.81**
Content navigation is easy and effective	2	13	26	30	9	**3.39**
Support for teleconferences is effective	0	2	6	42	30	**4.25**
Searching for content is adequately supported	5	13	24	29	9	**3.30**
There is adequate support for working in groups	0	6	10	46	18	**3.95**
There is adequate support on technical issues	0	0	10	38	32	**4.28**
Overall evaluation of the training programme						
It had clear objectives	0	6	17	40	17	**3.85**
It had activities that reflect and follow the objectives	0	2	12	43	23	**4.09**
The duration was adequate	0	7	10	48	15	**3.89**
The workload was adequate	3	12	23	31	11	**3.44**
There was adequate guidance by the trainers	0	1	7	33	39	**4.38**
My expectations were fulfilled	1	8	22	34	15	**3.68**
I can apply what I have learned	0	1	5	42	32	**4.31**
Now I will perform better in my teaching duties	1	3	14	40	22	**3.99**
I will not need support to apply what I have learned	0	10	27	22	21	**3.68**

Interesting findings were documented regarding issues related to the collaboration between participants in working groups and the establishment of code clubs as summarized in Table 3 below. The participants reported that they generally succeeded in reaching the goals of the programme especially with respect to familiarizing with the basic characteristics of Python and usage of self-contained examples in their teaching.

Their interest and participation was adequately focusing on the different phases of the training programme with a slightly more emphasis on the first phase. This is mainly due to some problems some participants had in combining their professional obligations with the code club creation phase. Positive evaluation of the collaboration of working groups was also reported. However, it was in some cases necessary for the trainers to intervene in working groups to help activities run smoothly. We believe this is important in such blended learning activities: The trainers should proactively and appropriately intervene to avoid problems in working groups. The training platform used offered adequate support for this monitoring taking into account the organization of the training through the use of particular services and the capability to send personal messages to individuals or group messages to members of working groups.

Table 3. Personal and group evaluation of the training programme (N=80)

Variable	1	2	3	4	5	mean
Training progr. success with respect to its initial objectives						
I familiarized with the basic characteristics of Python	0	2	11	47	20	**4.06**
I can use self-contained examples in teaching progr.	0	0	12	47	21	**4.11**
I was enabled to participate in a sustainable local CoP	4	6	17	36	17	**3.70**
Evaluation of personal participation						
I was much interested throughout the programme	0	10	20	30	20	**3,75**
I was more interested in Python programming (phase 1)	5	11	20	24	20	**3.54**
I was more interested in code club creation (phase 2)	5	17	27	19	12	**3.20**
I actively participated throughout the programme	0	10	11	38	21	**3.88**
I was more active in Python programming (phase 1)	5	9	22	25	19	**3.55**
I was more active in in the code club creation (phase 2)	5	18	23	19	15	**3.26**
I generally did the exercises within the deadlines	2	2	12	34	30	**4.10**
I did all/almost all individual and group exercises	1	8	8	23	40	**4.16**
I assimilated all/almost all the content of the programme	1	8	9	42	20	**3.90**
I participated adequately in the preparation and implementation of the code club of my working group	4	3	11	23	39	**4.13**
Evaluation of my working group						
Adequate collaboration when studying the learning mat.	1	8	20	27	24	**3.81**
Adequate collaboration during code club creation	1	5	12	28	34	**4.11**
Adequate collaboration during group exercises	1	9	9	34	27	**3.96**
Effective distribution of workload among group members	4	6	21	23	26	**3.76**
Fair distribution of workload among group members	7	13	16	24	20	**3.46**

An important aspect of the evaluation is that coding club establishment was seen as a complex and risky task initially while at the end of the programme, the organizers were in many cases surprised by their students asking to continue with the code club and even extend its theme and contents in many other topics exploiting the wide range of uses of the Python programming language. This finding was documented by the self-evaluation forms of the code clubs (prepared by the participating teachers) providing detailed description in free text of their experience. Furthermore, the self-evaluation reports document that the requirement to use and adapt appropriate work-sheets to support autonomous learning and personalization during the code club activities, although it was initially confronted with doubts, proved to be extremely effective. Furthermore, the approach to focus on self-contained projects instead of artificial examples and small programming exercises promoted student engagement and contributed to the creation of an atmosphere of meaningful learning.

Student evaluation of the training programme

Apart from teacher evaluation, as already mentioned, student evaluation was also done using an initial and a final questionnaire. The initial questionnaire contained 4 sections: (1) Demographic data; (2) Likert scale questions regarding the information about the code club, reasons for participation and expectations; (3) Likert scale questions about previous knowledge in programming; (4) Likert scale questions regarding attitude towards programming. The final questionnaire contained 5 sections: (1) Demographic data; (2) Likert scale questions regarding the evaluation of the code club; (3) Likert scale questions on the Python programming language and its possible use in secondary education; (4) Likert scale questions on the attitude towards programming; (5) Open-ended questions on the strong and weak points of the code club. By analysing the answers of these questionnaires (465 for the initial questionnaire and 358 for the final one) it is evident that the code clubs had a very positive impact on students in terms of developing programming skills and positive change in their attitude towards programming which is now seen as an important professional pathway. Table 4 below demonstrates these findings presenting only the change in students' attitudes using the corresponding questions of pre- and post-questionnaires. Due to lack of space we do not present an analysis of the other questions leaving it for a future presentation.

Table 4. Attitudes of students towards programming before and after their participation in the local code clubs (Npre=465, Npost=358). Percentages are shown for likert scale values.

Variable	1	2	3	4	5	mean
I believe that:						
Girls and boys are equally competent in coding	09.9	15.3	65.4	03.4	06.0	**2.80**
	07.8	12.6	65.6	06.4	07.5	**2.93**
I can collaborate with others when coding	06.5	08.4	24.3	26.7	34.2	**3.74**
	05.0	07.0	21.8	29.1	37.2	**3.87**
I can code by myself without help	12.5	23.7	24.9	20.2	18.7	**3.09**
	05.3	15.4	27.9	30.4	20.9	**3.46**
It is probable that I follow a computing profession	13.5	17.8	23.0	17.8	27.7	**3.23**
	10.9	12.3	21.8	21.2	33.8	**3.55**
Only future computer professionals should code	34.4	26.9	19.6	09.0	10.1	**2.34**
	27.7	24.9	25.7	11.7	10.1	**2.52**

6 Conclusions and future plans

The paper describes an approach and a concrete pilot experience with respect to teacher training in Python. The proposed training scheme is based on meaningful self-contained programming projects that are undertaken by students in coding clubs. A transfer was observed between knowledge acquired by the teachers and use of this knowledge with their own students. Evaluation was based on questionnaires and self-assessment for teachers and a pre-, post-questionnaire evaluation of the students that participated in the coding clubs established. Focusing on engaging programming projects rather than relying on artificial exercises addressing the syntax and the structure of Python (as it is in many cases the approach in traditional classroom teaching) highlights a wide range of higher-level concepts ranging from functional abstraction and problem-solving strategies to artificial intelligence.

Participating teachers were supported through distant-learning facilities offered by an appropriate learning platform (Coursevo) to study special material on the Python programming language and thus gain confidence in using an alternative, engaging methodology which can serve as a springboard for exposing their students to Computer Science practices and concepts. Participants, working together in regional groups, used these resources (a) to familiarize themselves with the Python programming language, (b) apply these resources in coding clubs employing pedagogically sound learning scenarios and (c) critically evaluate these resources and develop their own, based on the experience they acquire while applying them. The blended-learning approach followed promoted and enabled effective communication between tutors/trainers and trainees, as well as cooperation among trainees and access to courseware and learning resources.

A specific guideline to practitioners that wish to design and deliver effective training to teachers is to understand the importance of combining distant learning facilities with local cooperation of teachers and practical use of the acquired knowledge in organizing learning activities with students. This combination puts the general idea of blended-learning into a context that is highly effective providing motivation for teachers and increasing the positive impact to students.

Future work will address the transfer of the material and the methodologies reported in other domains (e.g. mathematics [15]) that could use coding to make learning playful and more engaging. Furthermore, we plan to explore the integration of visual interfaces to Python capitalizing promising results with respect to the performance gain that those interfaces can offer to novices [16].

Acknowledgments

The work reported in this paper was partially supported by the Google CS4HS programme (https://www.cs4hs.com/).

References

1. Fischer G. "Understanding, fostering, and supporting cultures of participation". interactions. vol. 18. 2011. pp 42-53.
2. Myers, B., Ko, A., J., Burnett M. (2006). Invited research overview: end-user programming. CHI'06 extended abstracts on Human factors in computing systems. 2006. pp 75-80.
3. Ko, A., J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., Scaffidi, C., Lawrance, J., Lieberman, H, Myers, B. (2011). The state of the art in end-user software engineering. ACM Computing Surveys (CSUR). 43(3), 21.
4. Aleksic, V., Ivanovic, M. (2016). Introductory Programming Subject in European Higher Education. Informatics in Education, 15(2), pp. 163-182.
5. Balanskat, A., Engelhardt, K. (2014). Computing our future: Computer programming and coding-Priorities, school curricula and initiatives across Europe. European Schoolnet.
6. Smith, N., Sutcliffe, C., Sandvik, L. (2014). Code club: bringing programming to UK primary schools through scratch. In Proceedings of the 45th ACM technical symposium on Computer science education. pp. 517-522.
7. Ovcin E., Cerato C., Smith D., Lameras P., Moumoutzis N. (2011). The pSKILLS Experience: Using Modern Educational Programming Languages to Revitalise Computer Science Teaching. International Conference on The Future of Education, Florence, Italy. Available online at: http://conference.pixel-online.net/edu_future/common/download/Paper_pdf/ITL70-Ovcin,Cerato.pdf
8. Smith, D., Lameras, P., Moumountzis, N. (2010). Using Educational Programming Languages to Enhance Teaching in Computer Science, EDGE 2010 Conference on the Use of Technologies in K-12 and Post-Secondary Education, Newfoundland & Labrador, Canada.
9. Peppler, K., Kafai, Y. (2005). Creative Coding: Programming for Personal Expression. 8th International Conference on Computer Supported Collaborative Learning (CSCL), Rhodes, Greece. Available online. Retrieved September 11, 2017, from https://goo.gl/pAHtQA
10. Moumoutzis, N., Christoulakis, M., Pitsiladis, A., Sifakis, G., Maragkoudakis, G., Christodoulakis, S. (2014). The ALICE experience: A learning framework to promote gaming literacy for educators and its refinement. International Conference on Interactive Mobile Communication Technologies and Learning (IMCL), 2014, pp. 257-261.
11. Zimmerman, E. (2009). Gaming Literacy – Game Design as a Model for Literacy in the Twenty-First Century. In Perron, B. and Wolf, M.J.P. Eds. The Video Game Theory Reader 2, New York; London. Routledge.
12. Pappas, N., Arapi, P., Moumoutzis, N., Mylonakis, M., Christodoulakis, S. (2011). The Multimedia Open Learning Environment (MOLE). Never Waste a Crisis!, 76.
13. Mylonakis, M., Arapi, P., Pappas, N., Moumoutzis, N., & Christodoulakis, S. (2011). Metadata management and sharing in multimedia open learning environment (MOLE). Metadata and Semantic Research, 275-286.
14. Pappas, N., Arapi, P., Moumoutzis, N., Christodoulakis, S. (2017). Supporting learning communities and communities of practice with Coursevo. Global Engineering Education Conference (EDUCON) 2017 IEEE, Athens, Greece, pp. 297-306.
15. Lameras, P. and Moumoutzis, N. (2015) Towards the Gamification of Inquiry-Based Flipped Teaching of Mathematics: A Conceptual Analysis and Framework. 2015 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL): pp. 343-347

16. Kyfonidis, C., Moumoutzis, N., Christodoulakis, S. (2017). Block-C: A block-based programming teaching tool to facilitate introductory C programming courses. Global Engineering Education Conference (EDUCON) 2017 IEEE, Athens, Greece, pp. 570-579.