Embedding Creativity in the University Computing Curriculum

Ed Currie [1] and Carl James-Reynolds[2]

[1] Middlesex University, London NW4 4BT, UK
[2] Middlesex University, London NW4 4BT, UK
lncs@springer.com

Abstract. We explore the need for embedding creativity in the UK Higher Education computing curriculum and some of the challenges associated with this. We identify some of the initiatives and movements in this area and discuss some of the work that has been carried out. We then describe some of the ways we have tried to meet these challenges and reflect on our degree of success with respect to the goal of producing graduates who are fit for the myriad of job opportunities they will come across in a rapidly changing technology landscape. Finally, we make a number of recommendations.

Keywords: Creativity, Computer Science, STEAM.

1 Introduction

Encouraging Higher Education computing students to develop their creativity is important to enable the UK economy to gain competitive advantage from early adoption and development of new technologies in areas such as business, entertainment, education etc.

Pink [1] explored the importance of imagination and creativity in employment and suggests that technology needs more than functionality to sell. Successful designs such as the iPhone provide evidence that creativity and design are important in engineering and can provide the USPs that enable a product to stand out against competitors' products.

In order to achieve this we need to equip our graduates with the appropriate knowledge, skills and creative flair that will enable them to tackle existing problems as well as identifying new opportunities for solutions. Capraro [2] explores STEM Project Based Learning (PBL) and believes that "PBL provides the contextualized, authentic experiences necessary for students to scaffold learning and build meaningfully powerful science, technology, engineering, and mathematics concepts supported by language arts, social studies, and art."

Traditionally, HE practitioners tend to classify and compartmentalize, keeping course content strictly within the discipline of study. There is also the temptation to homogenise educational content for the purposes of quality and adherence to standards, which also facilitates transfer of content between educators and the globalisation of material, ensuring that all content can be taught anywhere by anyone from the field of study, as exemplified by MOOCs and the National Curriculum.

This might be seen as a positive step, allowing for easy access to resources and facilitating quality audits. However Eisner [3] states that "The more we feel the pressure to standardize, the more we need to remind ourselves of what we should not try to standardize." Standardization carries a risk of stifling innovation and limiting creativity.

In the rest of this paper, we examine trends and initiatives for creativity in technical subjects generally and computer science in particular, and reflect on some of the attempts made to introduce more creativity into the curriculum in the Department of Computer Science at Middlesex University. We endeavour to address questions such as why we should teach creativity to computer scientists, how we can teach creativity and indeed, is creativity teachable?

2 Some motivations for technical creativity

The separation of arts and science is a relatively recent phenomenon and the study of the arts and a liberal education are seen less of a right and more as a pastime for those who do not need to earn a living. In part this has been encouraged by the commoditisation of education.

Many leading scientists throughout history have been involved in the arts from Einstein who played violin, to Da Vinci who often designed machines in his artwork. In education it has emerged that the meaningful integration of arts with the sciences can provide rewarding educational experiences that focus on creativity as playing a key role in problem solving and allows left and right brain skills to be integrated.

Mishra et al [4] discuss the synergies between the creative processes in physics and those in music, through the experiences of Ludwig Boltzmann, a leading 19th century physicist who was also a talented musician. He compared reading James Clark Maxwell's work on the theory of dynamic behaviour of gases, the interplay of the various equations and formulae, with the experience of listening to the interplay of the various instruments in the performance of a musical composition. It is interesting to speculate whether this is merely an interesting analogy, or whether there is some common thought process or mental mechanism that engages in each of these apparently different areas of activity. Mishra et al state that "…great thinkers in the areas of math and science often relate their efforts to music or the arts, highlighting the aural and visual experiences of their work, much more than the logical or formulaic." They cite a number of other examples of creative thinkers being inspired by the connections between their own discipline and many others, and furthermore argue that this should have a profound effect on how we approach teaching and learning; that we should be attempting to break down the boundaries between disciplines to nurture this process.

Scoffham and Barnes [5] discuss engaging children in "personally meaningful activity" and state that "Teachers also frequently affirm that the most effective learning happens when children are fully engaged creatively"

The importance of creativity in engineering is well documented and applies to computer science and multi-dimensional systems design, as we will discuss below. Indeed, the term 'functional creativity' was coined to express the idea of meeting functional requirements in novel ways [6]. A case study by Daly et al [7] found that some aspects of creativity were present in the engineering courses they considered, whereas others were not as prevalent. In particular, the need for convergent thinking such as analysis and evaluation was common, but divergent thinking such as exploration and generation of ideas was not.

Even in the domain of 'pure' computer programming, experts find it easy to identify so-called 'elegant' solutions, which differ from more crude solutions not in how they meet the functional requirements of the system, but in their simplicity and economy of style. This is also sometimes, but not always, a function of how well they meet well-known design criteria that enable understanding, modification and extending of systems.

Students' perception of what is required in the real world and relevant to their subject areas has been shaped by their courses at school and popular media representations of the subject area and associated professionals. Graham and Latulipe [8] in their exploration of recruitment and retention of female Computer Science students discuss the 'geek with a monitor tan' stereotype that most women wish to avoid and also that women have a stronger interest in real world problem solving that will benefit people, which they do not associate with computing. Art and design based subjects are seen as creative whereas Sciences are often viewed as areas requiring the remembering of lots of facts and equations, with little room for creativity. The growth of creativity in science should help to redress the gender imbalance in student recruitment.

3 Creativity and collaboration

Intuitively, we tend to think that collaborative learning is 'good for' students, although it raises many issues where assessment is concerned. Our intuition is supported by the fact that such learning has been shown to have a positive influence on critical thinking. A study by Gokhale [9] found that student performance when tested on so-called "drill and practice' activities was not significantly different whether they had studied individually or collaboratively. However, those who had engaged in collaborative learning performed significantly better on tests involving critical thinking than did those who had learned individually. These results supported the learning theories proposed by proponents of collaborative learning.

Gokhale states that "… it can be concluded that collaborative learning fosters the development of critical thinking through discussion, clarification of ideas, and evaluation of others' ideas", and for effective collaborative learning, "The instructor's role is not to transmit information, but to serve as a facilitator for learning. This involves creating and managing meaningful learning experiences and stimulating students' thinking through real world problems." The importance of collaboration is also shared by Steiner (10) who discusses creativity as involving collaboration or co-creation.

There has been a large increase in the number of on-line courses in recent years, and these present further challenges that might be addressable through the introduction of creativity in the curriculum. Levy and Ramin [11] investigated the skills needed for student success on e-learning courses, which have notoriously high non-completion rates. It is possible that a contributory factor to the high drop out rates in e-learning could be the sense of isolation and lack of a learning community to motivate and inspire students in this mode of study. We hypothesize that this could be mitigated in computing programmes by introducing some element of creativity into the learning process.

By their very nature, on-line computing programmes tend to somewhat rigid in content and involve less divergent computational thinking; students are often not encouraged to take an exploratory approach to problem solving. However, microcontroller kits can be very cheap and are easily sent through the post and collaborative creative activity using such devices is possible through on-line study groups.

4 Technical creativity in schools

Foster [12] states that "perhaps the two educators who had the greatest influence on the genesis of what is now known as technology education were Lois Coffey Mossman (1877- 1944) and Frederick Gordon Bonser (1875-1931)". As early as the 1920s, they identified problems in technical education, such as not relating work to the real world, poor motivation, not taking account of individuality and the emphasis of the product over the educational process. These issues, recognized so long ago, are still prominent in the teaching of technology in higher education today.

When computers were first introduced to schools, children used LOGO and turtle graphics to draw patterns and explore computing concepts such as loops and conditionals. Computers were also used to support artistic endeavours such as story writing. Henderson and Miner [13] observed that "Computers used effectively can become tools to build the minds and imaginations of future generations of creative thinkers".

STEM was an American initiative from 2006 to try to improve student performance by adopting an interdisciplinary approach, primarily in schools and colleges. This has since become STEAM with an acknowledgement that art plays a critical role in applications of scientific knowledge and that creativity is fundamental to solving problems. These initiatives have been adopted throughout the world and are increasingly finding their way into Further and Higher education.

Introducing creativity into a curriculum is not always easy; students are often risk averse and the National Endowment for Science, Technology and the Arts (NESTA) point out that innovation is urgently needed for economic growth and that risk taking is essential to innovation. They also point to lost opportunities for encouraging risk taking in STEM subject areas [14]. Csikszentmihalyi [15] records that Madeleine L'Engle stated "But we're allowed to learn from our mistakes and from our failures". Encouraging risk taking is possible, but it is important to have sufficient time to fail and learn from mistakes before assessment takes place.

Many schools now have cross curriculum days where they tackle problems using a wide range of skills. They also make use of STEM ambassadors who volunteer to assist in schools and provide examples of real world problems and careers. Some initiatives, such as BBC Microbits, are allowing children to play with embedded technologies and explore these in a wider context than just computing.

With the increased need for teachers to meet grade targets for their pupils to ensure career progression, there is an incentive for teachers to be risk averse and avoid making changes unless they can see these being rewarded. Whilst the syllabi try to facilitate flexibility, the exams and coursework together with limited resources available often significantly limit freedom in teaching. From a school perspective, literacy and numeracy are seen as being more important than other subjects as perceived by OFSTED and these subject areas are given priority in order to ensure the school's status is retained or improved. This is not a new phenomenon; Craft [16] discussed performative and creativity agendas co-existing and "the tendency among educators and policy makers, even then, to reduce creativity to the arts"

5 Technical creativity in HE

The issues of poor motivation, lack of individuality and emphasis of the product over the educational process recognized in school teaching so long ago by Mossman and Bonser [12], are still prominent in the teaching of technology in higher education today. The emphasis on the product manifests itself at the level of individual module assignments and also in the bigger picture of the purpose and intended outcomes for technical degree programmes. In the former, students are rewarded according to how well their constructed product meets the specification, with little regard to the processes the student has engaged with along the way. In the latter, our programmes are increasingly designed to be 'vocational' which is often interpreted to mean that they should anticipate and teach the same technologies that will be used in future employment, making them into glorified training courses. We hypothesize that nurturing polymaths requires a much more creative approach to technology higher education. That does not mean that repetitive practice of technique is not important, and indeed this is as vital a part of the training of artists and musicians as it is in computer programming or mathematics. Csikszentmihalyi [15] states "A musician must learn the musical tradition, the notation system, the way instruments are played before she can think of writing a new song".

However, we believe that this needs to be placed in a different context, in which such work is complemented by creative individual and collaborative activities that help to produce more rounded individuals.

The shift from a 'silo' approach to subjects in school is now being explored in higher education and in the way we approach teaching and curriculum design. Employers often look for creativity in employees and see this as a key skill. Integrated creative STEM approaches have often been seen in particular areas such as engineering and product design where strong technical and design skills are equally important. However good practice does not always spread from one area to another.

It is difficult to measure creativity, although a number of techniques have been tried. Bennett et al [17] explore the use of Computation Creativity in game design and use divergence from the tutorial norm as an indicator of creativity. They showed very different results for three different classes and it is possible that the tutor's style and perceived attitude has significant impact on the amount of risk that a student is prepared to take. It is interesting to note that an online course showed far less divergence. All students produced variations of the same game. It becomes more difficult when all students produce different games, as there is no tutorial norm. However, such coursework encourages a creative approach from the start and the student essentially is specifying their own game criteria as a contract with the tutor.

6 Embedding technical creativity in our computing curriculum

How can we teach creativity to computer science students? Indeed, can it be taught? A very eminent professor once told one of us that computer programming could not be taught. While this might be a somewhat extreme view, we believe it does have some factual basis, inasmuch as learning programming is a voyage of exploration for the student, in which the role of the teacher is to facilitate the journey and help students overcome the obstacles along their way. If this is true of what is the core subject of computer science, then it seems the learning of creativity should follow a similar trajectory.

Whether creativity can be taught is a moot point. In the arts, great emphasis is placed on developing technique, with the understanding that mastery of such technique allows individuals' innate natural creativity to blossom. Within technical disciplines such as computer science, technique is also heavily emphasised (computer programming, mathematical ability etc.) but sometimes what follows is merely assessment to prove that the technique has been assimilated.

A dominant idea in the discipline is that of computational thinking in problem solving. Computational thinking is in itself a highly creative process. However, the creativity required by computer scientists goes well beyond this, as computer scientists are required to work with experts in many other fields when developing software. The ability to communicate with others is key, as modern development methods involve continuous dialogue with clients and their involvement in the development process. This might be in the field of business, entertainment, infotainment or the arts. The ability to adapt and adopt is vital in such an environment. Many university computer science programmes follow a traditional pattern that does not emphasise the development of such skills.

Beaubouef [18] discusses the importance of communication skills for computer scientists, in order to succeed as computing professionals. It is important to have a wider knowledge beyond the discipline, or at least the ability to quickly acquire a grasp of an unknown domain, in order to model it. This emphasizes the role of the instructor in teaching students how to learn, as they will be required to continue learning throughout their careers. The ability to communicate with experts in other fields in a productive way to develop requirements etc. is key in modern software development. Extending this concept beyond the obvious areas of the sciences, medical, engineering, accounting, finance and economics, the ability to understand and appreciate the significance of the arts not only prepares students for careers involving the application of computer science in creative industries, but also develops creative and critical abilities that facilitate creativity in more conventional forms of system development. All of this supports the assertion that collaborative and perhaps multidisciplinary learning is important for the development of computer scientists.

The concept of abstraction is central to computer science; it is concerned with the ability to view a computational structure or code at different levels of detail and is vital when dealing with complex systems and development in teams. Defining and working with abstractions is another highly creative activity in the subject.

At Middlesex University, we have tried to encourage the development of students' creative abilities through individual and group project work that requires students to think beyond their current level of knowledge, to work with incomplete information and with existing systems, in order to produce new artefacts. [19] First year students work with electronics including programmable microcontrollers and sensors to model physical systems such as games and traffic lights. They extend this work to projects that involve programming bespoke robots with on board Raspberry Pi computers [20], infrared and bump sensors etc. Such projects are more motivating than more conventional screen and keyboard programming exercises, particularly as students can choose their own ideas for projects. Students are assessed according to how they demonstrate a number of key observable skills during the process of conducting these projects, rather than on the final product itself. This means that they can express themselves more creatively and take more chances, safe in the knowledge that they will not be penalised if their artefact is not fully functional. This addresses the issue discussed in section 5, of assessment rewarding the final product rather than the quality of the process followed by the student and also assists in embedding Csikszentmihalyi's trait of embedding playfulness and discipline [15].

This kind of work does not involve direct teaching of creativity. Rather, the role of the staff is to provide the environment and motivation to allow creativity to flourish. Therefore, staff are learning and creativity facilitators rather than deliverers although, of course, a few well-chosen words of encouragement or advice here and there can sometimes prod the creative process back into gear.

This approach enables students to build a portfolio of interesting completed projects which they can then share with potential employers, which should be more impressive than a CV alone. We often use a blog-style portfolio as assessment and allow the students some flexibility in adding material that may not be assessed, but does contribute to the portfolio in order that the portfolio may also be shown to potential employers. This portfolio may contain videos and links to software or other artefacts. Some students will choose to use this as an opportunity to demonstrate their web skills and other may use common blogging platforms.

Coursework does not have to be tightly specified; open-ended design tasks allow students to build and code devices and encourage them to explore the problem and solution spaces. A recent example was a challenge to build a useful phone app that used an accelerometer. The students were allowed to develop a range of possible apps, for example, to recognize falls or to recognize walking activity or to synchronize music to walking or running speed. One of our first year exercises requires students to investigate fast app development tools online, choose one of them and build an app. This is often just a drag and drop type of activity, which enables them to explore the limitations of the software and identify where this type of development might be useful and where it would be more appropriate to use native source code.

Another interesting question is: how do we measure creativity? This does not just mean how creative is the final solution, but also how creative was the path followed in developing that solution. Measuring the extent to which a solution is elegant and well-crafted is not difficult for an experienced computer science teacher, but assessing the creativity of the process followed is more problematical.

Perhaps a more fundamental question we might ask is what should be the purpose of a degree in computer science? We would not dispute that the content should prepare students for careers in the area, but how that should be achieved is an open question. Some would argue that the student should be equipped with a long list of technologies with which they can claim to be familiar, to impress potential employers. We would argue that the inclusion of creative projects in a programme will enhance the students' critical thinking abilities, self-reliance and communication skills, which are vital assets for a successful software engineer.

However, we also believe that the purpose of a degree, even in a technical subject, should not be just to prepare students for specific careers, but to expand their minds and make them into well-rounded, educated human beings. Not all computer science graduates pursue careers in computer science, but the skills they learn should stand them in good stead in many possible careers.

Research Through Design (RTD) is an approach to the acquisition of knowledge through the application of a design process. In this approach, the mode of thinking imposed by following a design process leads to greater understanding. Blythe [21] discusses the deployment of prototypes in Research through Design and the idea of "Design Fiction", whereby imaginary stories, films, artefacts or prototypes may be used for evaluating the potential of novel designs.

RTD has been used to some extent to facilitate creativity in the computing curriculum at Middlesex University. For example, students in their first two weeks are expected to design a fictional technology-based product that it might be possible to build in the near future. They then develop a short video to market their product. We have seen wide-ranging examples, from trainers that change colour to match different outfits, to pens containing Wi-Fi hubs and storage for a personal cloud. These exercises serve a number of functions; they allow students to get to know each other in the early stages of their course, they allow them to explore some of the difficulties faced with time management when they complete future assignments and to try out ideas in a non-assessed safe environment.

Students also write short fiction stories based on the interplay of emerging technologies. They are expected to extrapolate from the present using trends such as Moore's law and are expected to develop characters and use their imagination. The range of quality and scope of their stories is broad and students are often reluctant at first to engage with story writing. However, the process allows technology to be explored in a societal context that is important for developing reflective practitioners.

Research through Art and Design gives rise to the concept of the artist-researcher who is typically a practitioner. Educators in the arts have their own portfolios and interact with communities to display their work. We would argue that it is equally important that educators in the sciences are seen to be interacting with communities and producing work. The saying "He who can, does. He who cannot, teaches" from Bernard Shaw’s Man and Superman, can be dismissed when students see that those who are teaching are also doing. This message is even stronger when staff and students collaborate on pieces of work and bring them to wider audiences. Sometimes the students might be involved in stages of the work such as testing or dissemination, but they get insights into the creative processes required to produce work and can also contribute to future developments. Examples of these at Middlesex are STEM activities with local schools and the New Scientist Live show, where undergraduate students have displayed their own prototype MIDI controllers to create unusual sound effects, alongside laser harps that were designed as collaborations between the lecturer and MSc students. Examples of staff work used in first year classes include an augmented reality application [22], EEG controlled model racing cars used for demonstrating how to connect Arduino to Bluetooth and the use of pulse width modulation (PWM) to control motor speed in digital systems.

These activities allow students to become familiar with the importance of bringing ideas and artefacts to an audience, develop their confidence and enable collaboration with staff and other departments, leading to more diverse projects.

There are other opportunities to display work and engage with the outside world; for example many courseworks are now submitted on Social Media platforms in the form of Blogs and Videos. This not only helps develop an entrepreneurial approach and some understanding of marketing, but also allows the student to show their ideas to a wider audience, resulting in increased student motivation.

Students carry out individual and group projects with physical artefacts that often involve working with and modifying/ extending existing program code. Their research involves understanding the purpose and structure of the existing code, how it might be deployed in solving their specific design problems, how it can be modified etc. Students develop the ability to work with incomplete knowledge and understanding in a number of domains (coding, physical artefacts etc) and to incrementally and systematically acquire new knowledge and skills.

At Middlesex, we have a specialized MSc Creative Technology that enables students to tackle a range of interesting collaborative projects. These have ranged from laser harps to VR games with motion and gesture sensing. Some projects involve building a device from an existing design in order to develop skills in constructing artefacts. Students are also expected to produce videos of their work and develop pitch presentation skills. There is an emphasis on students attending hackathons and developing team skills. We are now exploring ways in which the students can engage with the wider community such as charities and local schools through STEM events.

Physical Computing allows students a more hands-on approach and students explore a wide range of computing concepts in an interactive and exciting way that facilitates the exploration of design and generation of ideas. A coursework description might specify at least two input sensors and visual or audio output and a scenario in which a device might be used. Student responses have included, car park monitoring systems, MIDI sequencers, alarm systems, air conditioning control systems, and interactive play equipment for children. By including a fun factor element in the marking scheme and submission of a video, students are provided with a great deal of scope for developing their creativity. It also is a move towards assessing the process and not just a product. Digital literacy is also developed with the repurposing of Social Media as a platform for marketing ideas and oneself in place of the entertainment and communication channel for friends. The students already have these skills, but need to use them in a new context.

7 Discussion and conclusion

At undergraduate level it is not possible to leave the fostering of creativity to chance. Students are often strategic learners and will take routes that avoid areas with which they are uncomfortable. It is therefore necessary to embed creativity into the curriculum and assessment in a number of different ways.

What does a modern creator look like? It is important for students to have role models and examples of computational creativity. In the past this might have been more difficult but now many examples can be viewed on Ted Talks; innovators such as Elon Musk are high profile and generate media interest.

For collaborative creative activity, it is important that rooms are fit for purpose, with a range of facilities. At Middlesex, desks are designed for collaborative working and have plenty of space to allow students to work effectively in groups. Brookes [23] identifies that "having an appropriate space does make a difference to student outcomes." In some instances space is required outside of timetabled class hours, particularly with project based work and a range of available spaces is made available, including lounge areas with provision for laptops, labs with facilities for soldering and electronics and lockers for projects to keep them organised and easily accessible. Students who need workshop skills to complete their projects can have inductions and access to a wide range of resources including 3D printing and laser cutting.

The following is a list of recommendations for embedding creativity in the curriculum based on our own experiences:

- Spaces matter – the right type of space is important for collaborative working and laying out materials. We have found that students do not work well in groups when they are seated in rows facing the tutor.
- Coursework should be open-ended and as loosely defined as possible, with students agreeing their own ideas with their tutors. It is important to still have a well-defined marking scheme and core elements of the work specified so that student are comfortable with the boundaries of what they might produce. Keeping the coursework open-ended allows for better differentiation and allows more capable students to extend their work.

- Students need to work in groups – this is important for creativity and is also a real-world skill. This does not always need to be formally assessed; however very short vivas can be used to quickly determine whether group members understand what has been produced.
- A proportion of staff should be practitioners who show their work in the community.
- Students should collaborate with such staff to develop interaction with the community; this may involve working with local schools as mentors or developing projects for those with disabilities.
- Software can be designed and redesigned; a specification for a software project can be open-ended with hints as to how an existing program might be improved - this is also a creative process.
- Risk and originality should be rewarded in a transparent way by ensuring that this is reflected in marking schemes.
- Students can be practitioners by making their work available on social media platforms (with appropriate safeguards in place).
- We should be assessing the journey made by the student and not just the destination; this can be done with blogs and video documentaries of their work.
- It is important to collaborate with other departments and learn from them. We have achieved this by collaborating on programmes and also through collaborative community work.

One of the problems with embedding creativity in computing programmes is that of perception; students currently see themselves as ‘technical’ and do not believe that creativity is what they have signed up for when undertaking their degree course.

Perhaps an insight into a possible future might come from the case study by Henriksen [24] of the approach of a winner of the 2008 National Teacher of the Year, Michael Geisen, who was lauded for his inspiring arts-based teaching of the sciences in school, pioneering the evolution of STEAM-based approaches to teaching STEM subjects. When such learning becomes ubiquitous in schools, creativity should come naturally to computer science undergraduates.

8 References

1. Pink, D.: A whole new mind: why right-brainers will rule the future. New York: Riverhead Books, (2006).
2. Capraro, R., Slough, S.: Why PBL? Why STEM? Why Now? An Introduction to STEM Project-Based Learning: An Integrated Science, Technology, Engineering, and Mathematics Approach. M. Capraro, Ed., In STEM project-based learning: an integrated science, technology, engineering, and mathematics (STEM) approach, 2. Ed. Rotterdam [u.a]: Sense Pub, (2013).
3. Eisner, E.: Artistry in Education. Scandinavian Journal of Educational Research, vol. 47, no. 3, pp. 373–384, Jul. (2003).

4. Mishra, P., Henriksen, D.; A NEW Approach to Defining and Measuring Creativity: Rethinking Technology & Creativity in the 21st Century. The Deep-Play Research Group. TechTrends; Washington Vol. 57, Iss. 5, 10-13. (Sep 2013).
5. Stephen Scoffham & Jonathan Barnes (2011) Happiness matters: towards a pedagogy of happiness and well-being, Curriculum Journal, 22:4, 535-548
6. Cropley, D., Cropley, A.: Engineering creativity: A systems concept of functional creativity. In J. C. Kaufman & J. Baer (Eds.), Creativity across domains: Faces of the muse (2005). (pp. 169–185). Mahwah, NJ: Lawrence Erlbaum
7. Daly, S., Mosyjowski, E., Seifert, C.: Teaching Creativity in Engineering Courses: Teaching Creativity in Engineering Courses, Journal of Engineering Education, vol. 103, no. 3, pp. 417–449, Jul. (2014).
8. Graham, S., Latulipe, C.: CS girls rock: sparking interest in computer science and debunking the stereotypes, in ACM SIGCSE Bulletin, vol. 35, pp. 322–326. (2003).
9. Gokhale, A.: Collaborative Learning Enhances Critical Thinking. Journal of Technology Education vol. 7, no. 1, (1995).
10. Vera John-Steiner (2000) Creative Collaboration, Oxford University Press ISBN: ISBN-13 978—0-19-530770-2
11. Levy., Y. Ramim, M.: The e-learning skills gap study: Initial results of skills desired for persistence and success in online engineering and computing courses. in Proceeding of the Chais 2017 Conference on Innovative and Learning Technologies Research, 2017, pp. 57–68. (2017).
12. Foster, P.: The Founders of Industrial Arts in the US. Journal of Technology Education vol. 7, no. 1, (1995).
13. Henderson, A., Minner, S.: Computing For Creativity. Intervention in School and Clinic, vol. 27, no. 1, pp. 43–46, (1991).
14. Rolfe. H.: Learning to Take Risks to Succeed. NESTA June 2010 LTR/29 https://www.nesta.org.uk/publications/learning-take-risks-learning-succeed, last accessed 2018/5/15.
15. Csikszentmihalyi, M. 1997. Creativity: Flow and the psychology of discovery and invention. New York: HarperCollins.
16. Anna Craft Creativity and Education Futures Trentham Books 2011 ISBN: ISBN-978-1-8585-6462-3
17. Bennett, V., Koh, K., Repenning, A.: Computing creativity: divergence in computational thinking. In Proceeding of the 44th ACM technical symposium on Computer science education, pp. 359–364. (2013)
18. Beaubouef, T., Why computer science students need language. ACM SIGCSE Bulletin, vol. 35, no. 4, pp. 51–54, (2003).
19. Currie, E., James-Reynolds, C.: The Use of Physical Artefacts in Undergraduate Computer Science Teaching. 3rd EAI International Conference on e-Learning e-Education and Online Training (eLEOT 2016), Dublin, Ireland**,** August 31 – September 2, 2016**.**
20. Androutsopoulos K. et al: MIRTO: an Open-Source Robotic Platform for Education. In ECSEE'18 Proceedings of the 3rd European Conference of Software Engineering Education, Pages 55-62, Seeon/ Bavaria, Germany — June 14 - 15, 2018
21. Blythe, M.: Research through design fiction: narrative in real and imaginary abstracts. pp. 703–712. (2014).
22. Apparition Dornier 17 http://rafmuseum.mdx.ac.uk/dornier17/download-a-dornier/, last accessed 2018/5/15.
23. Brookes, W.: On creativity and innovation in the computing curriculum. pp. 17–24. (2018).

24. Henriksen, D. Full STEAM Ahead: Creativity in Excellent STEM Teaching Practices. *STEAM*, vol. 1, no. 2, pp. 1–9, Feb. (2014).