Can Creative Computing foster Growth Mindset?

Michael Lodi

Department of Computer Science and Engineering
Alma Mater Studiorum - Università di Bologna & INRIA Focus, Italy
michael.lodi@unibo.it

CEUR-WS.org/Vol-2190/TACKLE_2018_paper_3.pdf

Abstract

Teacher training in computational thinking (CT) is becoming more and more important, as many countries are introducing CT at all K-12 school levels. Introductory programming courses are known to be difficult, and some studies suggest they foster an entity theory of intelligence (fixed mindset), reinforcing the idea that only some people have so-called "geek gene". This is particularly dangerous if thought by future primary school teachers. We analyzed the effects of an introductory course about *computational thinking and creative computing with Scratch*, and observed a statistically significant increase of pre-service teachers' growth mindset while observing a statistically significant decrease in their computer anxiety. The structure of the course is detailed, with particular emphasis on some characteristics that may have determined growth mindset increase. Limitations of this exploratory study are discussed, and future work is depicted.

1 Introduction and Motivations

[1] In the last decade, *computational thinking* (CT) has been recognized as a fundamental skill for everyone, not just computer scientists [Win06]. Many countries in the world are making efforts to include it in the school curriculum, at all K-12 levels [GP13].

In Italy, the recent school system reform explicitly states that it is mandatory to develop student's digital skills, with particular care to the development of *computational thinking*.

These pushes to teach CT (that very often is realized by teaching programming - or "coding") give rise to the necessity of an urgent plan for teacher's training, both for pre-service and in-service ones, and especially for primary school teachers. In Italy, in facts, primary teachers are not trained to teach CS fundamentals (and only since 2002 they need a Primary education degree to teach). Moreover, Italian primary teachers are mostly female, and therefore possibly subject to stereotypes about women and CS.

To non-computer scientists, learning to program may appear as a too challenging goal, achievable only from those having the so-called "geek gene" [AL13, PBCE16]. Moreover, stereotypes lead some people to identify computer scientists with singularly focused, asocial, competitive, male figures [LAY16].

Students and teachers have different personal ideas ("implicit theories") about their intellectual abilities. Some believe that their intelligence is a fixed trait (like eye color or height when adult), and they cannot do much to change it: they have an *entity theory of intelligence*, otherwise stated a *fixed mindset*. Some others believe instead that intelligence can be developed with study and effort (like muscles can be trained): they have an *incremental theory of intelligence*, also called a *growth mindset*. Mindset theory is a fundamental result of Carol Dweck's research [Dwe00]. In many studies, she showed that student's mindset could predict their achievement, in particular in Math and Science, and their ability to cope with challenges [BTD07, Dwe08]. Moreover, female students with a growth mindset showed less susceptibility to the harmful effects of stereotypes about women and math [GRD12]. In [MT08] it is argued that growth mindset can be particularly important in CS educa-

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: A. Piotrkowicz, R. Dent-Spargo, S. Dennerlein, I. Koren, P. Antoniou, P. Bailey, T. Treasure-Jones, I. Fronza, C. Pahl (eds.): Joint Proceedings of the CC-TEL 2018 and TACKLE 2018 Workshops, co-located with 13th European Conference on Technology Enhanced Learning (EC-TEL 2018), 03-09-2018, published at http://ceur-ws.org

[1] Parts of introduction, motivations and background were already described in [Lod17]

tion. Teachers' growth mindset is strongly necessary (although not sufficient) to foster a growth mindset in their students: teachers must create a growth mindset environment where growth messages are sent, but also teach students new strategies to cope with failures and to master the material [Dwe17].

Since 2014, at the University of Bologna, a laboratory course on "computational thinking and creative computing with Scratch" has been taught to preservice primary teachers. During the first two years of teaching, instructors collected oral reports from students. At the beginning of the course, many of them reported anxiety and low self-efficacy about learning to program. Some of them described themselves as "not a computer science/technology person". On the contrary, at the end of the course, instructors received very good feedback. Some students spontaneously thanked them because they did not feel any more like they were not "computer science people" and felt empowered to be creative with technology and to teach it to their future pupils.

Based on these anecdotal pieces of evidence, we decided to explore changes in mindset and computer anxiety before and after the fall term course of the academic year 2016-2017. The test on mindset was replicated in the spring term course.

2 Background

2.1 Computational Thinking

As known, the term *computational thinking* was first used in 1980 by Seymour Papert in Mindstorms [Pap80] and then brought to the attention of our community by Jeannette Wing [Win06] in 2006. In this decade, a body of literature has been produced to search for a better definition of this concept, and to provide tools and frameworks to introduce and assess CT in K-12 education [GP13]. While there is no agreement between authors, a lot of proposed definitions recognize CT is not only about technical methods and practices, but also about mental processes and transversal skills like *creativity, collaboration, tolerance for ambiguity, resilience,* and more [CLN17].

2.2 Scratch and Creative Computing

Scratch is a visual programming environment to create interactive media-rich projects, like video games, interactive stories, interactive art, and so on [MRR+10]. Scratch was developed at MIT Media Lab by the Lifelong Kindergarten group. It is built on constructionist ideas of Papert's LOGO [Pap80]. Some key features of Scratch, relevant for this work, are:

- *liveness* - the program is constantly running, and its behavior changes immediately when parts of the code are added, edited or removed;
- *tinkerability* - the program lets you experiment with blocks, in a very bottom-up, trial and error approach;
- *no error messages* - like when you play with LEGO®bricks, either blocks snaps together, or they don't; moreover, if your program is not correct, it runs anyway, so you don't feel too frustrated, and then you can try to figure out why it doesn't behave as expected.

Scratch's main goal is to teach digital fluency as a means of self-expression rather than as a tool for future careers [RM+09]: often students can use technology as passive users, but few of them have the opportunity to be active creators through technology.

Scratch belongs to MIT's vision about creative learning, and was specifically designed to help young people grow up as creative thinkers (through the "creative learning spiral" - an iterative approach to creativity - and through four main ingredients: Project, Peers, Passion, Play [Res14]).

These ideas are implemented in the Harvard's "Creative Computing Online Workshop"[2] and "Scratch curriculum guide" [BBC14] and in the "Learning Creative Learning" course at MIT[3]. Materials from these initiatives represent the primary sources for the course presented in this paper.

2.3 Growth Mindset

Dweck's studies on growth mindset are based on three decades of research [Dwe00]. Students with growth mindset show learning-oriented goals (not afraid to ask questions and make mistakes, in order to learn) and mastery-oriented responses (greater effort and new strategies) to challenges and setbacks, while students with fixed mindset show performance goals ("appear intelligent", so avoiding difficult tasks) and helpless response to challenges (e.g. giving up or blaming the teacher for their failure). As said in Section 1, growth mindset is positively correlated with grades and achievements and can be useful to reduce gender disparities in STEM.

Growth mindset can be positively conveyed by some interventions [Dwe08]: explicitly teaching students about mindsets, brain plasticity and the idea that intelligence can be trained with effort; portraying challenges, effort and mistakes as highly valued; praising process and effort, and give constructive feedback rather than praising the person or being judgmental.

Specific suggestions to stimulate a growth mindset in Math includes also [Boa13]: giving rich open tasks,

[2] `https://creative-computing.appspot.com/preview`
[3] `http://learn.media.mit.edu/lcl/`

oriented to learning, requiring effort and reasoning; teaching for patterns and connections; teaching creative and visual Mathematics.

Teachers' conceptions are crucial: in a study described in [Dwe08], adults were asked to behave as teachers - after a fixed or a growth mindset about Math had been taught to them. The "growth" group was more supportive with students, giving encouragement and suggesting positive strategies to deal with problems; by contrast, the "fixed" group subjects gave students simple comfort and fixed messages (e.g., "*Not everyone is a math person!*") and helped boys significantly more than they did with girls.

2.4 Computer Anxiety

Computer anxiety can be defined as "*a fear of computers when using one, or fearing the possibility of using a computer*" [SON05], and differs from negative attitudes toward computers. In facts, it involves a more affective response: "*resistance to and avoidance of computer technology are a function of fear and apprehension, intimidation, hostility, and worries that one will be embarrassed, look stupid, or even damage the equipment*" [HGK87].

Computer anxiety has been correlated with math anxiety and gender [HGK87, Mau94]. Females were found to have higher computer anxiety. By contrast, previous exposure to computers is correlated with the low level of anxiety. As [Mau94] suggests, females have less exposure to computers than males due to stereotypes, so previous exposure should be taken into account.

A study, referenced by Dweck herself, correlates computer anxiety and self-theories. It showed that computer anxiety decreased in participants of a basic computer training course who were taught incremental conceptions of ability, while did not change in participants to whom fixed entity conceptions of ability were taught [Mar94].

To assess computer anxiety, the "Computer Anxiety Rating Scale (CARS)" was developed and validated in [HGK87].

3 Previous Work

As opposed to other scientific disciplines, only a few studies have been conducted on the relationship between an introductory computer science/programming course and a growth mindset. In a survey administered to CS faculty members of a U.S. institution [Lew07], more than three-quarters of them disagreed on the fact that "Nearly everyone is capable of succeeding in the computer science curriculum if they work at it". Carol Dweck herself describes computer science as a discipline that requires a growth mindset [CCD+10].

Like math, computer science can induce a fixed mindset, as some authors [MT08, CCD+10] suggest. By contrast, we think some intrinsic characteristics of CS/CT (at least if taught as a creative subject - e.g., open/real/authentic projects, iterative approach, debug, trial and error, collaboration rather than competition) can foster a growth mindset. In other fields, for example Engineering, similar results were found: during the first year of University, students tend to move towards a fixed mindset. However, introducing open-ended engineering design projects into the curriculum may tend to lessen or eliminate the shift toward fixed mindset [RF14].

Only a few studies have been conducted to assess or alter the student's mindset before and after a programming course.

- Simon et al. [SHM+08] tried a small intervention in CS1 classes to change the mindset of students from CS Majors and Minors, but they obtained mixed results.

- Cutts et al. [CCD+10] performed three structured interventions into an introductory programming course, gaining significant improvement in growth mindset level of students and also a positive correlation in their test scores.

- On the contrary, Flanigan et al. [FPSS15] analyzed (without intervention) changes in CS1 (CS-major, other STEM-Majors, but also Arts and Business Majors) students across the semester, finding a significant increase in a fixed mindset and a significant decrease in a growth mindset.

All the cited experiments were conducted among CS1 students. Authors of the present paper did not find any study investigating correlations between growth mindset and CT courses.

4 The study

We decided to observe growth mindset and computer anxiety changes between the beginning and the end of the laboratory course.

We decided to not teach explicitly about growth mindset or brain growth, and the instructor (a computer scientist with a background in education) paid particular attention in avoiding explicit mentioning of Dweck's research and ideas in lessons/suggested reading material to avoid influencing the subjects and to test if CT and creative learning could influence growth mindset.

No active intervention was made to influence computer anxiety.

4.1 The Context

Currently, to become a pre-school and/or a primary school teacher in Italy, you have to get a 5-year (Single cycle/Combined Bachelor and Master) Degree in *Primary teacher education.* When graduating, students also get a "Pre-school and Primary school teaching license" that allows them to teach in Italian public schools. For historical and sociological reasons, in Italy primary teachers are mainly female and this is reflected in the fact that *Primary teacher education* students are almost all female (for instance, 91% in a.y. 2016/17 in our University).

At the University of Bologna, primary teacher education students take a mandatory "General Education and Educational Technologies" exam during the first year, and follow a practical 24 hours (3 credits) "Educational Technology Laboratory" during the fourth year. For that course, they can choose between several topics, from the use of interactive white-board, to stop-motion storytelling techniques and many others, all with the aim to learn how to use technology as a tool for better teaching. To allow students to be supported by instructors and to work with technologies actively, each thematic-laboratory has a maximum of 32 students. In the context of this "multi-track course," since the academic year 2014-15, students can choose the "Laboratory of creative computing and computational thinking", to learn the basics of computational thinking and creative computing with Scratch. To give the opportunity to take the course to more students (of the same academic year), the laboratory is replicated (same instructor, schedule, location, contents) in the fall term and in the spring term.

4.2 The Course

The course was designed as an introduction to creative computing and computational thinking with Scratch. It was made up of 6 lessons, 4-hours each. The course plan is now described. Many activities are taken from MIT/Harvard materials (see sec. 2.2).

Lesson 1.

- Brief introduction to creative computing and computational thinking;
- experiments with Google Presentations: the teacher creates a shared presentation with writing rights, then she asks students to add a new page and to write something about themselves, putting on a photo, and so on. This activity is initially messy, but soon students learn in a very bottom-up fashion to use the tool and to avoid modifying peers content;
- free exploration of Scratch;
- mini challenge to make something happen with it ("Scratch surprise" [BBC14]);
- guided tutorial to create a simple video game that contains a lot of computational concepts[4].

Lesson 2.

- Ten blocks challenge[5];
- free artistic project with just the simple hint to use the "pen" and "looks" categories blocks;
- debug exercises: students had to choose some debug exercises from [BBC14], remix them, find the bug and comment out how they found it and what they did to correct it.

Lesson 3.

- Witness from an invited primary school teacher using unplugged activities and Scratch in her teaching;
- examples of Scratch projects to be used with pupils[6];
- "about me" [BBC14] free project: create a project to introduce yourself and the things you do and love.

Lesson 4.

- Exploration of Code.org and comparison with Scratch: try some activities, find out pros and cons of the different platforms and approaches;
- advanced features (cloning and webcam): try to reproduce a "snowing-like" behavior with cloning and catching the clones with the hand through webcam-sensing[7].

Lesson 5.

- Scratch and the physical world (Makey Makey[8] and Lego WeDo[9]) demos;
- time to work in small groups on the final project.

[4] A simplified version of Carmelo Presicce's "Under the sea": https://scratch.mit.edu/projects/14759947/

[5] https://creative-computing.appspot.com/unit?unit=4&lesson=13

[6] e.g. from https://scratch.mit.edu/studios/1918506/

[7] e.g. https://scratch.mit.edu/projects/129283065/

[8] e.g. the classical "human chain" and "whack a mole": https://scratch.mit.edu/projects/43681296/

[9] e.g. simple sensor/motor use with Scratch described in: https://www.youtube.com/watch?v=qBhIcb-Ipmw

Lesson 6.

- Public presentation of final projects: design of an activity with Scratch for Primary School in the light of creative learning's 4 Ps. Students were advised to be particularly careful to not create a game or a project on which their pupils would have been passive consumers, but rather design a creative activity where their pupils should be free to create different projects but in the context of some Primary school teaching objectives of one or more subjects.

Each student had her own PC, but they were allowed to work in pairs/small groups, to get up and move around the laboratory and to communicate with each other. The teacher was always available to offer help.

The laboratory was mainly hands-on: students were assigned projects with a broad theme (e.g., "a project about you") and given time to freely create with Scratch, experimenting, getting help online or asking the instructor. Sometimes, more structured exercises were given, but they were chosen not to be mechanical or repetitive. Instead, they were explicitly posed as challenges to have fun with, while learning. No theoretical lectures about programming or Scratch were given. However, some tips and quick demos, always *after* they worked a while on projects or problems, were given.

Homeworks consisted in realizing other projects at home and writing a page in a shared online notebook (Google Presentation), reflecting on difficulties, achievements, and learning process. The instructor gave feedback as comments, and students were invited to comment at least two of their mate's pages each week.

An online virtual class was set up with a Google+ community, where students could ask for help (to mates and to the teacher) and discuss. The instructor posted interesting videos/articles and stimulated comments.

The exam was pass/fail, with no grades. At the beginning of the course, it was clearly stated that students would have been evaluated for their effort (measured with the presence in class, participation, shared projects) rather than on the quality of their works. The final presentation of a group project was also mandatory to pass the exam. Students were in particularly encouraged to share their projects even if they were buggy or incomplete, and ask for help.

All students passed the exam in both terms. Projects and journals were not graded but were checked by the instructor, and written feedback was given.

No explicit reference to growth mindset and brain growth theories were made. However, other growth mindset strategies (see 2.3) were put in action: in particular it was carefully paid attention to give *growth mindset feedback* (both oral and written in the comments to projects or posts) and it was *praised process and outcome* ("You worked a lot to create that!", "Very good project!") rather than the person ("Bravo!", "You are very good at it!").

The instructor established a good class climate, where errors were not stigmatized but seen as a powerful tool for learning, helping students reflect on them with guided questions rather than simply "tell the solution". This was eased by the tool: Scratch helps you not be frustrated by errors and instead motivates you to figure out how to fix your bugs.

Scratch tinkerability helped also to encourage students to not give up, moving forward by trial and error and feeling empowered by their learning and successes.

4.3 Data Collection

An identical survey was administered at the beginning (pre-survey) and at the end of the course (post-survey). It was dived into two sections (Growth Mindset and Anxiety) in the fall term course, while had only the Growth Mindset section in spring term's one.

The first section was intended to assess students' mindset through the Implicit Theories of Intelligence Scale (see Appendix A.1 for the full scale). It included eight Likert-type items, described in [Dwe00, p. 285]. Students were asked to rate from 1 ("completely disagree") to 6 ("completely agree") eight statements about intelligence (in Italian in the survey), four reflecting an incremental theory, like "*No matter who you are, you can significantly change your intelligence level*" and four reflecting an entity theory, like "*You have a certain amount of intelligence, and you can't really do much to change it*". During the analysis, the latter were reverse scored, so that high points were associated with a growth mindset, while low points with a fixed mindset.

The second section was intended to assess student's anxiety through the Computer Anxiety Rating Scale (see Appendix A.2 for the full scale). It included nineteen Likert-type items, described in [HGK87] (we used an Italian translation of the slight variation [SON05] of the original statements). Students were asked to rate from 1 ("completely disagree") to 5 ("completely agree") proposed statements (in Italian in the survey), half of them reflecting a high anxiety (like "*I have avoided computers because they are unfamiliar and somewhat intimidating to me*" or "*I do not think I would be able to learn a computer programming language*") while half of them reflecting a low level of anxiety (like "*Learning to operate computers is like learning any new skill, the more you practice, the better you become*"). During the analysis, the latter were reverse scored, so that high points were associated with high anxiety, while low points with low anxiety.

The pre-survey was administered right at the beginning of the fall term course, when students knew only the title and a very brief description of the course from the website. The post-survey was administered at the end of the course. Five weeks passed between the two administrations.

The questionnaires were anonymous, but answers of the same subject to pre and post-questionnaire were linked with a randomly generated code unknown to the researcher. The questionnaires were administered with an online form (Google Form).

This process was repeated for the spring term course, but with mindset questions only.

A total of twenty-three students ($N = 23$), all females, aged from 21 to 29 ($M = 23$, $SD = 1.95$, $MODE = 22$) completed both the pre-survey and the post-survey of the fall term course.

Moreover, a total of twenty students ($N = 20$), all females, aged from 22 to 28 ($M = 23.05$, $SD = 1.82$, $MODE = 22$) completed both the pre-survey and the post-survey of the spring term course.

4.4 Data Analysis

The data were analyzed with the R programming language and RStudio environment.

4.4.1 Growth Mindset

For each subject, the initial and final growth mindset level was calculated. Growth mindset level is a value from 1 (fixed mindset) to 6 (growth mindset), calculated as the mean of the eight answers (with entity items reverse scored, as stated) of each subject.

A paired-samples t-test was conducted to compare growth mindset at the beginning and at the end of the fall term course. There was a statistically significant difference in the mindset scores between the pre-test ($M = 4.62$, $SD = 0.78$, $\alpha = 0.86$) and the post-test ($M = 4.90$, $SD = 0.76$, $\alpha = 0.86$): $t(22) = -2.35$, $p = 0.028$ (< 0.05). In particular, growth mindset has increased from the beginning to the end of the course. (see Fig. 1, where mean GM of all subjects is represented as a black diamond).

Moreover, a paired-samples t-test was conducted to compare growth mindset at the beginning and at the end of the spring term course. Again, we found a statistically significant increase in the mindset scores between the pre-test ($M = 4.08$, $SD = 0.80$, $\alpha = 0.76$) and the post-test ($M = 4.44$, $SD = 0.83$, $\alpha = 0.90$): $t(19) = -2.50$, $p = 0.022$ (< 0.05) (see Fig. 1).

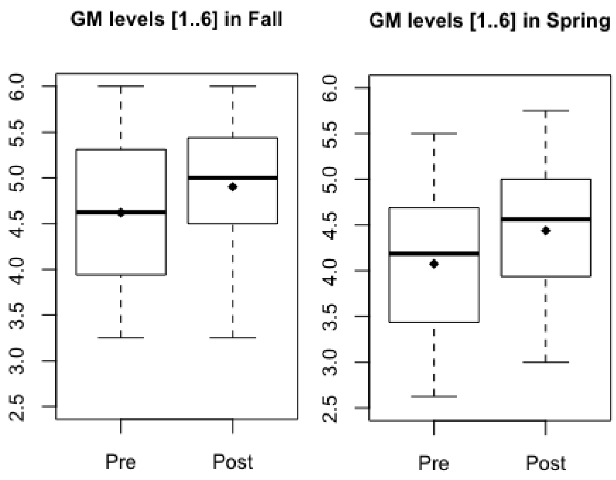

Figure 1: Distribution of growth mindset levels before and after fall and spring term course

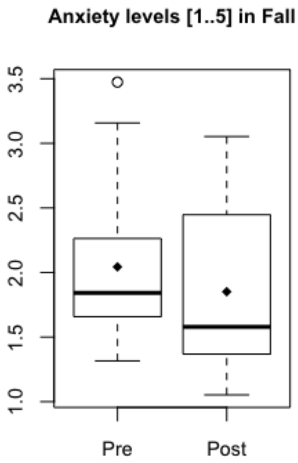

Figure 2: Distribution of anxiety levels before and after fall term course

4.4.2 Computer Anxiety

For each subject, initial and final computer anxiety level was calculated. Anxiety level is a value from 1 (low anxiety) to 5 (high anxiety), calculated as the mean of the nineteen answers (with some items reverse scored, as stated) of each subject.

A paired-samples t-test was conducted to compare computer anxiety at the beginning and at the end of the course. There was a statistically significant difference in the anxiety levels at the beginning ($M = 2.04$, $SD = 0.58$, $\alpha = 0.90$) and at the end ($M = 1.85$, $SD = 0.60$, $\alpha = 0.92$): $t(22) = 2.98$, $p = 0.007$ (< 0.01). In particular, computer anxiety has decreased from the beginning to the end of the course (see Fig. 2).

4.5 Data Validity

All answers showed an high internal consistency (see *Cronbach's alphas* in data analysis).

To be valid for a paired t-test, distribution of the differences between the two related values of each subject should be approximately normally distributed. Differences from both growth mindset and anxiety scores passed the Shapiro-Wilk normality test, recommended for small samples.

Finally, both measures are resistant to test-retest [HGK87, DCH95].

4.6 Limitations of the Study

Since it is a pre-experimental design, this study is afflicted by some limitations. In particular:

- there is no control group, and so we don't know if the course is the *only* or the *main* cause of the difference between results, or if external factors may have intervened; as a positive observation, anyway, both fall and spring groups registered an increase in mindset;
- the sample was relatively small and not randomized: it was made up of students that decided to attend the class;
- changes may be influenced by the experience of taking the test itself;
- *regression towards the mean* may have influenced the results.

5 Results, Conclusions, and Further Work

Despite the limitations of the study, we found a statistically significant increase in participants' growth mindset (result replicated in a following identical course) and a statistically significant decrease in participants' computer anxiety.

Initial growth mindset was already high. This is hardly surprising because it is crucial for teachers not to have fixed views about intelligence and almost certainly this has been taught them in the previous years of their degree. However, this clearly contrasts with oral reports about their "CS confidence" collected at the beginning of the course. We suspect they may have a high growth mindset in general, but hold fixed ideas about CS in particular: it is known one can have different mindsets in different areas [Dwe17].

More surprising is their medium/low initial anxiety. This may be due to their young age: they grew up in a world where technologies are everywhere, so they rapidly get used to them. It may be the case to construct an updated anxiety scale, which considers this diffusion and tests anxiety about more profound skills related to computer science rather than computers themselves or use specific CT/CS anxiety scales. It would also be interesting to measure CS self-efficacy.

Most interventions taught explicitly about brain growth to influence self-theories about intelligence. Even though we recognize this is crucial, we aim to foster a growth mindset mainly with teaching innovations and fundamental aspects of computer science. "Explicitly teaching" interventions can be further positive boosters for a growth mindset.

The aim of our exploratory study was mainly to test whether our insights about CT, creative computing and growth mindset were correct. Now we have to:

design a proper experiment to confirm these preliminary data; investigate more deeply what factors of CS/ CT/ creative computing are the most significant to foster growth mindset: we believe that teacher's feedback, iterative approach, open projects and challenging exercises were crucial; define and investigate specific "computer science mindset", rather than general ideas about intelligence; evaluate the relationship between (CS) growth mindset and actual learning of CT concepts.

Acknowledgements

Many thanks to all the students of the courses; to Simone Martini, for supporting my research and for his useful suggestions; to Alessandro, for the help with statistics; to Carmelo and all CoderDojo friends, for making me discover Scratch and Creative learning.

References

[AL13] A. Ahadi and R. Lister. Geek genes, prior knowledge, stumbling points and learning edge momentum: Parts of the one elephant? In *Proc. of ICER 2013*, pages 123–128, New York, NY, USA, 2013. ACM.

[BBC14] K. Brennan, C. Balch, and M. Chung. Scratch curriculum guide, 2014. http://scratched.gse.harvard.edu/guide/.

[Boa13] J. Boaler. Ability and mathematics: the mindset revolution that is reshaping education. In *Forum*, volume 55, pages 143–152. Symposium Journals, 2013.

[BTD07] L. S. Blackwell, K. H. Trzesniewski, and C. S. Dweck. Implicit theories of intelligence predict achievement across an adolescent transition: A longitudinal study and an intervention. *Child Development*, 78(1):246–263, Jan 2007.

[CCD+10] Q. Cutts, E. Cutts, S. Draper, P. O'Donnell, and P. Saffrey. Manipulating mindset to positively influence introductory programming performance. In *Proc. of SIGCSE 2010*, pages 431–435, New York, NY, USA, 2010. ACM.

[CLN17] I. Corradini, M. Lodi, and E. Nardelli. Conceptions and misconceptions about computational thinking among italian primary school teachers. In *Proc. of ICER 2017*, pages 136–144, New York, NY, USA, 2017. ACM.

[DCH95] C. S. Dweck, C. Chiu, and Y. Hong. Implicit theories and their role in judgments and reactions: A word from two perspectives. *Psychological Inquiry*, 6(4):267–285, 1995.

[Dwe00] C. S. Dweck. *Self-theories: Their role in motivation, personality, and development.* Psychology Press, 2000.

[Dwe08] C. S. Dweck. Mindsets and math/science achievement. *The Opportunity Equation*, 2008.

[Dwe17] C. S. Dweck. *Mindset (Updated Edition).* Robinson, 2017.

[FPSS15] A. E. Flanigan, M. S. Peteranetz, D. F. Shell, and L. Soh. Exploring changes in computer science students' implicit theories of intelligence across the semester. In *Proc. of ICER 2015*, pages 161–168, New York, NY, USA, 2015. ACM.

[GP13] S. Grover and R. Pea. Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1):38–43, Jan 2013.

[GRD12] C. Good, A. Rattan, and C. S. Dweck. Why do women opt out? Sense of belonging and women's representation in mathematics. *Journal of Personality and Social Psychology*, 102(4):700–717, 2012.

[HGK87] R. K. Heinssen, C. R. Glass, and L. A. Knight. Assessing computer anxiety: Development and validation of the computer anxiety rating scale. *Computers in Human Behavior*, 3(1):49–59, Jan 1987.

[LAY16] C. M. Lewis, R. E. Anderson, and K. Yasuhara. "I Don'T Code All Day": Fitting in Computer Science When the Stereotypes Don'T Fit. In *Proc. of ICER 2016*, pages 23–32, New York, NY, USA, 2016. ACM.

[Lew07] C. Lewis. Attitudes and beliefs about computer science among students and faculty. *ACM SIGCSE Bulletin*, 39(2):37, Jun 2007.

[Lod17] M. Lodi. Growth mindset in computational thinking teaching and teacher training. In *Proc. of ICER 2017*, pages 281–282, New York, NY, USA, 2017. ACM.

[Mar94] J. J. Martocchio. Effects of conceptions of ability on anxiety, self-efficacy, and learning in training. *Journal of Applied Psychology*, 79(6):819–825, 1994.

[Mau94] M. M. Maurer. Computer anxiety correlates and what they tell us: A literature review. *Computers in Human Behavior*, 10(3):369–376, Sep 1994.

[MRR+10] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond. The scratch programming language and environment. *Trans. Comput. Educ.*, 10(4):16:1–16:15, November 2010.

[MT08] L. Murphy and L. Thomas. Dangers of a fixed mindset. *ACM SIGCSE Bulletin*, 40(3):271, Aug 2008.

[Pap80] S. Papert. *Mindstorms: Children, Computers, and Powerful Ideas.* Basic Books, Inc., New York, NY, USA, 1980.

[PBCE16] E. Patitsas, J. Berlin, M. Craig, and S. Easterbrook. Evidence that computer science grades are not bimodal. In *Proc. of ICER 2016*, pages 113–121, New York, NY, USA, 2016. ACM.

[Res14] M. Resnick. Give P's a chance: Projects, Peers, Passion, Play. In *Proceedings of Constructionism and Creativity Conference*, 2014.

[RF14] K. J. Reid and D. M. Ferguson. Do design experiences in engineering build a growth mindset in students? In *2014 IEEE Integrated STEM Education Conference*, pages 1–5, March 2014.

[RM+09] M. Resnick, J. Maloney, et al. Scratch: Programming for all. *Commun. ACM*, 52(11):60–67, November 2009.

[SHM+08] B. Simon, B. Hanks, L. Murphy, S. Fitzgerald, R. McCauley, L. Thomas, and C. Zander. Saying isn't necessarily believing: Influencing self-theories in computing. In *Proc. of ICER 2008*, pages 173–184, New York, NY, USA, 2008. ACM.

[SON05] H. Sam, A. Othman, and Z. Nordin. Computer self-efficacy, computer anxiety, and attitudes toward the internet: A study among undergraduates in unimas. *Journal of Educational Technology & Society*, 8(4):205–219, 2005.

[Win06] J. M. Wing. Computational thinking. *Commun. ACM*, 49(3):33–35, March 2006.

A Questionnaires

A.1 Implicit Theories of Intelligence Scale

It included eight Likert-type (1 to 6) items, described in [Dwe00, p. 285]. We used an Italian translation. Questions with * indicate a fixed mindset, so they were reverse scored (so that high agreement corresponds with a growth mindset for all questions)

Q1* You have a certain amount of intelligence, and you can't really do much to change it.

Q2* Your intelligence is something about you that you can't change very much.

Q3 No matter who you are, you can significantly change your intelligence level.

Q4* To be honest, you can't really change how intelligent you are.

Q5 You can always substantially change how intelligent you are.

Q6* You can learn new things, but you can't really change your basic intelligence.

Q7 No matter how much intelligence you have, you can always change it quite a bit.

Q8 You can change even your basic intelligence level considerably.

A.2 Computer Anxiety Rating Scale

It included nineteen Likert-type (1 to 5) items, described in [HGK87] (we used an Italian translation of the slight variation proposed in [SON05]). Questions with * indicate a low level of anxiety, so they were reverse scored (so that high agreement corresponds with high anxiety for all questions)

Q1 I feel insecure about my ability to interpret a computer printout

Q2* I look forward to using a computer on my job

Q3 I do not think I would be able to learn a computer programming language

Q4* The challenge of learning about computers is exciting

Q5* I am confident that I can learn computer skills

Q6* Anyone can learn to use a computer is they are patient and motivated

Q7* Learning to operate computers is like learning any new skill, the more you practice, the better you become

Q8 I am afraid that if I begin to use computer more, I will become more dependent upon them and lose some of my reasoning skills

Q9* I am sure that with time and practice I will be as comfortable working with computers as I am in working by hand

Q10* I feel that I will be able to keep up with the advances happening in the computer field

Q11 I would dislike working with machines that are smarter than I am

Q12 I feel apprehensive about using computers

Q13 I have difficulty in understanding the technical aspects of computers

Q14 It scares me to think that I could cause the computer to destroy a large amount of information by hitting the wrong key

Q15 I hesitate to use a computer for fear of making mistakes that I cannot correct

Q16 You have to be a genius to understand all the special keys contained on most computer terminals

Q17* If given the opportunity, I would like to learn more about and use computers more

Q18 I have avoided computers because they are unfamiliar and somewhat intimidating to me

Q19* I feel computers are necessary tools in both educational and work settings