Engaging High School Students Using Chatbots

Luciana Benotti
Logic, Interaction and Intelligent Systems Group
FAMAF, Universidad Nacional de Córdoba/CONICET
Ciudad Universitaria, Córdoba, Argentina
benotti@famaf.unc.edu.ar

María Cecilia Martínez
Facultad de Filosofía y Humanidades
Universidad Nacional de Córdoba/CONICET
Ciudad Universitaria, Córdoba, Argentina
cecimart@gmail.com

Fernando Schapachnik
Fundación Dr. Manuel Sadosky and
Depto. de Computación, FCEyN, UBA
Buenos Aires, Argentina
fschapachnik@fundacionsadosky.org.ar

ABSTRACT

Chatbots have been used in different scenarios for getting people interested in CS for decades. However, their potential for teaching basic concepts and their engaging effect has not been measured. In this paper we present a software platform called Chatbot designed to foster engagement while teaching basic CS concepts such as variables, conditionals and finite state automata, among others. We carried out two experiences using Chatbot and the well known platform Alice: 1) an online nation-wide competition, and 2) an in-class 15-lesson pilot course in 2 high schools. Data shows that retention and girl interest are higher with Chatbot than with Alice, indicating student engagement.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer science education

General Terms

Education

Keywords

Computer science K-12 outreach, chatbot, experimental evaluation, engagement, gender

1. INTRODUCTION

Argentinean universities graduate only 3500 Computer Science (CS) students a year (compared to 10000 in Law and 15000 on Economics) while the national industry needs to hire twice that amount. Part of the problem is that CS is not taught at K-12 level. This lack of early CS education influences career choices as students may not be selecting CS simply because they do not know what CS is [6]. We found through student surveys that although more than 90% of our Argentinean K-12 students use computers as consumers, most of them believe that programming means "installing programs".

This context is not unique to Argentina, many developed countries share the same problem (e.g., [27, 13, 1]). The industry and the government are tackling this problem with several initiatives. One of them is a programming contest based on the well-known tool Alice [8, 9] called Dale Aceptar, described in Section 3.2, that despite having attracted more than 27000 students, faces the issue of low female participation and low retention rates. In an effort to improve these issues, the Sadosky Foundation,[1] the institution running the competition, partnered with the Universidad Nacional de Córdoba to develop Chatbot.

This article documents the findings of that experience and makes the following main contributions:

- We present the educational tool Chatbot which is an open source software that hides the complexity of chatbot programming. It provides a simple interface that helps students learn basic Computer Science concepts such as variables, conditionals, and finite state automata, among others.
- We propose a pedagogical strategy to increase student engagement by using it in a specific way: a gamified, inquiry oriented, structured task where students program a chatbot and get automated progress feedback. It is gamified as a mystery game called "Alibi".
- We develop a 15-lesson pilot classroom learning experience and report on the engagement and difficulties encountered by K-12 students while learning basic CS concepts using Chatbot and Alice in both the classroom and the online contest Dale Aceptar.

The rest of the article is organized as follows. It describes the Chatbot platform (Section 3) and both the outcome of its introduction in the Dale Aceptar contest and in a pilot study specially designed to test it in a classroom environment (Section 4). Before that, Section 2 surveys previous work in the area of engaging K-12 students in CS learning. Some final remarks and future research agenda conclude the article in Section 5.

[1]The Manuel Sadosky Foundation is a public/private institution whose goal is to promote stronger interaction between ICT industry and the scientific-technological system. Its Chairman is the Minister of Science, and the Vice-chairmen are the chairmen of the most important ICT chambers of the country.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ITICSE'14, June 21–25, 2014, Uppsala, Sweden.
Copyright 2014 ACM 978-1-4503-2833-3/14/06 ...$15.00.
http://dx.doi.org/10.1145/2591708.2591728.

2. PREVIOUS WORK

The typical K-12 student in Argentina almost never encounters CS topics during its school years. Computer Science is just not taught at school, not even as an optional course. The curriculum in Argentina focuses on ICT training classes rather than CS content. In these courses, computing entails little more than learning how to use a word processor, a spreadsheet or create an online blog. Students often get bored in their ICT classes and outperform their own teachers. This context is not unique to Argentina, many developed countries share the same problem [27, 1, 13, 5]. There are some exceptions such as Israel, among others, where CS has been taught at high schools for many years now [28].

The need to interest more K-12 students in CS has been addressed by government institutions, companies, universities and teachers around the world. The work is extensive and varied and we we will not be able to make a complete survey here. We just comment on some representative examples. Many excellent initiatives have bloomed, like Alice [8, 10], CS Unplugged [2], the Computer Science Teachers Association and Code.org, just to mention a few. As a result, several outreach programs exist that are interested in how to best enthuse more students. Many activities and different approaches to teaching CS are being evaluated. Doran et al [12] developed and tested a curriculum for video-game design and evaluated its impact on student engagement and performance in other fields, such as Math and English.Other studies concentrate in analyzing teachers' opinions [26, 5].

Despite advances in the last decades, it is still a matter of debate how to interest students, and specially girls, in CS. Moreover, there is no consensus on what CS concepts should be included in the school curriculum (e.g., [1, 14, 22, 7]). Documenting experiences of teaching CS in high school will contribute to the discussion of what and how CS concepts could be taught at high school in order to keep the students interested.

There has been much research on how to foster interest in CS by teaching how to program 3D video-games. Alice [8, 9] is a pioneer in this regard whose engagement potential has been thoroughly investigated. Such research lead to the development of Alice Story Telling [16] that was specially designed to interest girls in CS. In Section 4 we compare Chatbot with Alice.

Chatbots have been used in different scenarios for getting people interested in CS for decades. In most cases chatbots are used as tutoring systems [17, 11]. Several programming clubs have proposed the programming of chatbots as a method for interesting their students in computer science. Shaw [25] used chatbot programming to teach computing principles in introductory CS courses. Keegan et al [15] presented Turi, a chatbot software for secondary and primary schools. Bigham et al [4] used low level chatbot programming to inspire blind high school students to pursue Computer Science. Bigham et al study showed that this was a successful tool in spite of technical difficulties such as the fact that the produced chatbots could not connect to the more used social networks. In spite of their repeated use, to the best of our knowledge, the potential of chatbot programming for teaching basic CS concepts and its engaging effect has not been measured. In this paper we not only evaluate the engagement shown by students with no previous interest in CS but we also compare it to animations and video game programming with Alice. For this study, engagement means student cognitive investment on learning and completing the task. The main indicators for engagement are amount of student participation (task completion and attendance), intensity of concentration, enthusiasm, and expressed interest [20].

3. CHATBOT

Chatbot is an educational software tool whose design goal is to motivate students to learn basic CS concepts through the construction of chat automata. It has a mode of operation where it can connect to social networks (such as Gtalk and Facebook) and reply to chat conversations automatically. The chatbots can be programmed to answer in different ways depending on who it is talking to, what the person is saying, which topic they talked about before, etc.

By programming their chatbots, students learn basic CS concepts such as variables, conditionals, finite automata, recursion, randomness, regular expressions, among others. Chatbot can also be used to explore more advanced concepts such as the Turing test and Natural Language Processing concepts (e.g. lemmatization and syntactic analysis). Chatbot is open source and is available at `bit.ly/1iglAf6` .

3.1 Teaching Basic CS Concepts with Chatbot

In this section we illustrate how Chatbot can be used to teach the basic Computer Science concepts of **variables** and **conditionals**.

Chatbots are programmed in Chatbot by writing sets of **(pattern, effect)** pairs. The chatbot responds with the *effect* when the *pattern* matches the stimulus received by the chatbot. Patterns are simple regular expressions that may include wildcards and variables, and effects may include variables and conditionals (among more advanced structures) as illustrated in Figure 1.

In this example the chatbot programmed is the suspect of a murder who is talking to the leading detective. The following dialogue is given to the students and they are asked to implement a single (pattern, effect) pair to program a chatbot that can answer like this suspect.

Detective: Do you think that the cook is the murderer?
Suspect: it's possible that the cook did it
Detective: or the photographer is the murderer?
Suspect: no, I am sure he didn't do it
Detective: what if the gardener is the murderer?
Suspect: it's possible that the gardener did it

A possible correct answer to the exercise is shown in Figure 1. In the figure the pattern includes the wildcard * that can match any number of words and the variable [person] that stores the value of the word that comes right before the phrase "*is the murderer?*". The effect is a conditional expression that, depending on the value of the variable [person], may give two different answers (one of them uses the content of the variable).

Pattern: * [person] is the murderer?
Effect: {**if** [person] = photographer}
no, I am sure he didn't do it
{**else**} it's possible that the [person] did it

Figure 1: Sample (pattern, effect) pair that uses variables, wildcards and conditionals in Chatbot.

3.2 Open-Online Contest for Teenagers

Dale Aceptar (Spanish for "just hit OK" but also "go with it") is a free online competition organized by the Sadosky Foundation (`www.daleaceptar.gob.ar`). Based on Alice, it is performed annually with the aim of interesting more students into pursuing CS-related careers. The competition is atypical, in the sense of being aimed at students with no prior background in programming, who sign in because they see the commercials on national TV and feel like having fun, win a prize or both. While they participate, students are also exposed to short pieces of information about CS and its advantages as a career choice.

The site offers 23 short-video lessons on Alice, going from the basics up to building a turn-taking, timer-based game. Multiple fora provide support for Q&A. The competition has attracted more than 27000 students along its three editions.

Besides proposing working with Alice, the 2013 edition added a "gamified" alternative: students could participate in "Alibi", a murder story based on Chatbot. Five funny suspects, a corpse and a detective are left alone in a mountain. The detective keeps a log book with his findings and speculations, which is weekly made available to participants along with an interrogation questionnaire from the detective for each suspect. Participants download the file and are supposed to program their Chatbot so that it answers the questions properly (a structured task). Chatbot confesses guilt if it cannot find a matching rule, and flags an answer as incorrect if there is a rule but the output does not match the (encrypted) regular expression that the questionnaire file has for identifying correct answers. Students must keep their bot from confessing but also from flagging answers as incorrect. Based on how well the bot answers a score is calculated. It reaches 100% if all questions are answered properly. A jury of experts then picks winners among the top ranked bots, which must be programmed using concepts such as variables and finite state automata. As in the case of Alice students are supposed to learn by watching the five online Chatbot tutorial videos and using the support fora.

Figure 2: "Alibi" suspects lining up for questioning.

3.3 Comparative Pilot Study at High School

At the same time "Alibi" was launched we conducted a pilot study using both Chatbot and Alice in two public high schools in the city of Córdoba (Argentina) through a 15-lecture course. The expectation was to compare Alice and Chatbot without the bias of self-learning that Dale Aceptar has. We also wanted to know how students from diverse and specially poor context, with no previous interest on CS, engage in programming using Chatbot.

Introducing Chatbot in the context of public high schools also promoted the platform dissemination. We agree with Pears [23] that researchers often spend a great amount of time developing a teaching tool, but very little effort disseminating its use. Tools need customization and pedagogical work before other educational institutions can adopt them.

The Alice course was designed to teach students how to program animations and the Chatbot course was designed to teach students to program chatbots that played the role of a suspect in a short version of "Alibi". Tutors visited the schools once a week to teach Alice in the first place and Chatbot in the second. The rationale was presenting students with a platform that could develop a familiar product such as animations first, and then move to less known products such as chatbots. It is possible, though, that if we had started with Chatbot, our results would have been different.

The lesson design for teaching both Alice and Chatbot followed a discovery based approach [24]. All lessons had four different segments. 1) Motivational. Aimed at interesting and challenging students to create the need to use some CS concept. In this segment, we presented students with a short challenge, such as moving an object in Alice in a particular way or reproducing a short dialogue in Chatbot. 2) Short lecture. Which consisted in a brief introduction to a CS concept that can solve the problem. E.g., conditionals. Intentionally, the tutor does not solve the problem leaving room for student discovery. 3) Exploration and production. Students explore the platform combining the right concepts that solve the challenge. The purpose of the segment is exposing students to experimentation for gaining understanding [21]. 4) Show and assess. In the last part of each lesson students share their progress on their animations or chatbots with other students. Student construction, presentation and evaluation of their products seems to improve learning computer programming [19].[2]

The pilot course was attended by 47 students, the average age was 15.4 years old. 55% of the students were female and 45% were male. Students from both schools had similar socio-economical situations: most students came from impoverish families. The course was mandatory and taught during school hours, however students were not evaluated and did not get extra credits for the course. All students completed the evaluation survey of the course, whose results are reported in Section 4.2.

An assistant made classroom observations and both tutor and assistant filled in post observations notes after each lesson. Both the assistant and the tutors had previously designed the lesson plans. In addition, students were given a pre-test, mid-test (when Alice module concluded) and a post-test at the end of the Chatbot module. We triangulated qualitative and qualitative data to increase the validity of the study and to better describe our findings adding a pedagogical dimension [3].

4. FINDINGS

In this section we report the outcomes of both introducing Chatbot in Dale Aceptar and the comparative pilot study in high schools.

4.1 Open-On line Contest for Teenagers

Besides the issue of lack of female participation, we also wanted to address engagement: although many students sign

[2]Chatbot offers the obtained score and incorrect question flags as a self evaluation mode that provides students with feedback on the quality of their rules.

	Alice Start	End	*%End*	Chatbot Start	End	*%End*
Female	1022	16	*1.57%*	337	27	*8.01%*
Male	7480	93	*1.24%*	1117	75	*6.71%*
%Female	*12%*	*15%*		*23%*	*26%*	
Total	8502	109	*1.28%*	1454	102	*7.01%*

Table 1: Comparison of 2013 participants that registered (Start) vs those that uploaded their work to the competition web page (End).

in, only a few are self-motivated enough to complete the task of designing a game or animation and participate until the end of the competition. Although it is well-known that retention rates are low in online courses [18], a specific note about retaining teenagers in that setting is in order.

As argued in [20], in the classroom engagement is the product of three main factors, 1) the need for personal competence (which varies among socio-economic status) 2) the types of tasks students are required to do (mechanic, fun, authentic), and 3) the school environment (support, care, fairness, academic status). Positive school and classroom environment include teachers providing personal support to avoid frustration when difficulties arise, and caring about students as individuals in a context where academic expectations are clear and school success is promoted for all.

How does that work in the online world? What do you do to retain students, who have not logged in to your site for a while, so that they can come back? How do you reach out proactively to offer help? Some years ago, the email address provided when registering would have been the way, being a communication channel that was independent of whether the user visited back the site or not. Nowadays teenagers do not use email. They communicate through Facebook, and that platform, partly to avoid spam, do not allow sites ("fanpages" in Facebook jargon) to initiate communications, so a good part of retention tools are lost. We believe that this lack of strategies to reach back to students that had not logged in for a while explains the small percentage of participants that stayed in the competition long enough to present some finalized work piece ("End" column in Table 1).

As can be seen in Table 1, more people decided to participate with Alice than with Chatbot (8502 vs 1454). We attribute the difference to the fact that most teenagers do not know what a chatbot is while Alice was presented as a tool to program video games and animations, two concepts very familiar to them.

Two interesting observations can be made from Table 1. First, although Dale Aceptar attracts mainly male students (partly because the prize of the competition is a gaming console), the percentage of female registration is twice higher with Chatbot (23%) than with Alice (12%), this difference is statistically significant (Chi-square test, p=0.01). We do not know what motivations students have for choosing one platform over another, but similar female preferences were seen in the classroom (see next Section), and the evidence collected there points into the direction of highlighting a gamified task such as "Alibi" that requires the use of language and dialogue, as valuable to increase girls interest. Second, despite the challenges for retention in online courses, engagement, as measured by number of students who completed their work in the competition, reaches 7.01% in Chatbot while only 1.28% in Alice, this difference is statistically significant (Chi-square test, p=0.01). We believe that the difference might be attributable to "Alibi" providing both a more structured task, with more clearly defined goals and periodic updates, in a platform such as Chatbot that provides feedback in the form of a score, as opposed to the more open and unstructured task of designing and creating video games or animations.

4.2 Comparative Pilot Study at High School

We conducted a pilot study in high schools to collect more detailed information on students behaviour using both tools, to learn about how students use Chatbot in the classroom and to analyze its potential to engage students in programming. After the pilot classroom course we applied a post-test to the students asking the following questions that provided us with indicators of student engagement. All questions are based on a scale ranging from 0 (meaning "not at all") to 10 (meaning "very much").

C1) How **interesting** was learning Chatbot for you?
C2) Do you want to **learn more** using Chatbot?
C3) How **easy** was learning Chatbot for you?
A1) How **interesting** was learning Alice for you?
A2) Do you want to **learn more** using Alice?
A3) How **easy** was learning Alice for you?

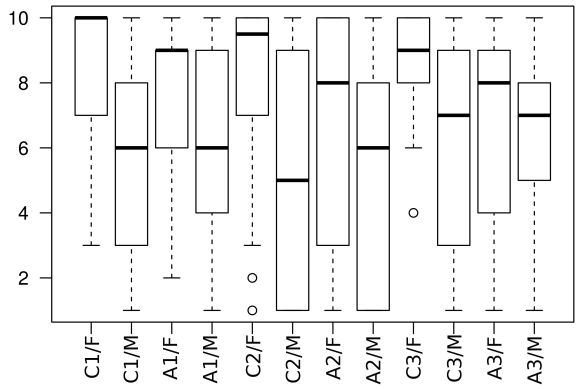

Figure 3: Results of engagement indicators obtained in the high school pilot study, by gender.

In Figure 3 we show a box plot of the results obtained in the post-test. Girls' self-reported interest was higher than boys' both for Chatbot and Alice. However, the difference is statistically significant for Chatbot (Chi-square test, p=0.01) but not for Alice. Girls' interest with Chatbot had a median value of 10 over 10 and girls' interest with Alice had a median value of 9 over 10. After finishing the course, girls want to learn more using Chatbot (median 9.5) while half of the boys don't (median 5), the difference is statistically significant (Chi-square test, p=0.01). In terms of easiness, the differences are not statistically significant, neither by gender, nor by tool. However, one can observe more variation in Chatbot for boys and Alice for girls than vice versa.

4.2.1 Interest with Alice and Chatbot

To increase the validity of our study we triangulated self reported student data with tutors perceptions documented in post lesson observations. While the tutors reported that

Platform	"Interesting"	"Fun"	"Engaged"
Chatbot	5	6	10
Alice	5	0	8

Table 2: Frequency of keyword appearance in post-lesson observations.

most lessons were engaging for students, upon a closer examination of their discourses we discovered subtle, but yet interesting differences, which mirror the findings on the items C1 and A1, students self reported interest.

In general, according to our tutors and research assistants, Chatbot resulted more engaging for our students. When describing the attitude of students working on their lesson they used the words "engaged", "interested" and "fun", all indicators of engagement according to Newmann [20]. Table 2 shows the number of times each word was repeated in the post lesson observations. Tutors seemed to observe that students have "fun" more often in Chatbot than in Alice. Also, the word "hook" appears more often in Chatbot. One possible explanation is that programming with Chatbot was part of playing the game "Alibi". As with the online experience, we believe the "gamification" of the task could have provided a source of fun. For example, based on post lesson reflections, one lesson included collective testing of some interesting pre-made chatbots (the psychologist and a chatbot that chats about his birthday). As those worked well, students showed interest in seeing how they were programmed. After that segment of the lesson, "Alibi" was presented. Students got hook into the characters, and all of them preferred to start creating their own "Alibi" chatbot instead of trying to build one of the topic of their choice.

4.2.2 *Explaining "Easiness"*

The second emerging theme in the qualitative analysis was that most CS concepts tackled either with Alice or Chatbot resulted "easy" for the majority of the students. Tutors and assistants reported in their observations that students learned "easily", solved most the challenges and discovered new rules or instructions to develop their products. For example one reflection mentioned: "*I asked them to write something that required a conditional and gave the class time to find the right tool to solve the problem. In particular, one of the students found the option 'create a conditional rule' and solved the challenge.*" Other classroom observations considered for this theme reported that students could solve challenges "rapidly". Tutors observed that students easily understood and applied conditionals.

However, some concepts did result harder for our students. As an example, in the Chatbot module, in one school, the tutor reported students had difficulties understanding finite state automata, despite eventually being able to apply it into their chatbots.

One finding emerging from the analysis of classroom observations is that some concepts students can discover or learn more intuitively than others. For example, variables and conditionals were intuitive concepts that students discovered when exploring the platform. Some other concepts, such as dummy objects in Alice and finite state automata in Chatbot seemed to require much more thought, practice and analysis. In spite of this, students reported that they found both platforms easy to learn.

5. CONCLUSIONS

In this article we document an experience on using the well-known educational tool Alice and Chatbot, a chatbot programming platform, both in an online competition and in classroom environments. The purpose of creating the tool was increasing student retention and engagement, specially in girls, while teaching basic CS concepts, both as a way of promoting interest towards CS-related careers and contributing to the increasingly important discussion of what CS concepts should be taught at school and how they should be approached.

We found that most indicators of engagement (task completion, participation, enthusiasm and self reported interest) increased when using Chatbot in comparison to Alice. With some differences between girls and boys, in the online experience, task completion and participation rates grew by a factor of 5 on Chatbot vs Alice. In the in-classroom pilot course, girls' self-reported interest was much higher than boys' as was their willingness to learn more using Chatbot.

The data seems to align well with previous reports stating that girls engage more in verbal oriented activities such as developing chatbots. Our mix has other ingredients as well: besides having a platform in common, both the online and classroom experience were structured in the format of a game called "Alibi" (gamification). Teaching materials such as the questionnaires, online videos and Chatbot immediate feedback, as well as the discovery-based lesson, were carefully designed to promote student engagement. We believe that more structured tasks and immediate feedback are important, and future work should include collecting evidences to prove or disprove their significance as individual drivers of engagement.

For instance, work with Alice could also be made more structured, focusing on building a particular type of game instead of each student choosing their favorite, and requiring to follow some sort of schedule where each week a particular aspect of the game is tackled, providing feedback to students on the quality of what they are doing. Last issue is key: providing support and feedback on the their programming can be done in class, while an online contest would need an immense amount of resources to provide the same level of support and feedback.

The difference on engagement could be due to the fact that the programming concepts covered are not exactly the same. For instance, Alice has data types, methods and parameters, which are key to CS and somehow harder to understand. If the difference in engagement could be attributed to Chatbot being somehow "incomplete", a teaching strategy could be depicted for girls: start with more engaging albeit "incomplete" tools, get them to the "want to learn more" state (Figure 3) and only then move to more powerful platforms.

Despite the scholarly interest of researching the separate significance of each of the these separate variables (platform and teaching approach), we highlight their combined value, and future agenda includes digging deeply into integral approaches on how to best enthuse and engage students in CS.

6. ACKNOWLEDGMENTS

This work was partially supported by grants ANPCyT-PICT-2008-306, ANPCyT-PICT-2010-688, ANPCyT-PICT-2012-712, the FP7-PEOPLE-2011-IRSES Project MEALS, two Google RISE awards, and grants by the Argentinean

Ministry of Science, Technology and Productive Innovation's Manuel Sadosky Foundation. The authors wish to thank the teams at Universidad Nacional de Córdoba and the Manuel Sadosky Foundation for their support and collaboration.

7. REFERENCES

[1] T. Bell, P. Andreae, and L. Lambert. Computer science in New Zealand high schools. In *Proceedings of the Twelfth Australasian Conference on Computing Education - Volume 103*, pages 15–22, 2010.

[2] T. Bell, P. Curzon, Q. Cutts, V. Dagiene, and B. Haberman. Introducing students to computer science with programmes that don't emphasise programming. In *Proceedings of the 16th ACM annual joint conference on innovation and technology in computer science education*, pages 391–391, 2011.

[3] A. Berglund, M. Daniels, and A. Pears. Qualitative research projects in computing education research: An overview. In *Proceedings of the 8th Australasian Conference on Computing Education - Volume 52*, pages 25–33, 2006.

[4] J. P. Bigham, M. B. Aller, J. T. Brudvik, J. O. Leung, L. A. Yazzolino, and R. E. Ladner. Inspiring blind high school students to pursue computer science with instant messaging chatbots. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*, pages 449–453, 2008.

[5] J. Black, J. Brodie, P. Curzon, C. Myketiak, P. W. McOwan, and L. R. Meagher. Making computing interesting to school students: Teachers' perspectives. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, pages 255–260, 2013.

[6] L. Carter. Why students with an apparent aptitude for computer science don't choose to major in computer science. *SIGCSE Bulletin*, 38(1):27–31, 2006.

[7] Computing at School Working Group. *Computer Science: A Curriculum for Schools*. Computing at School Working Group, 2012.

[8] S. Cooper, W. Dann, and R. Pausch. Teaching objects-first in introductory computer science. In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, pages 191–195, 2003.

[9] W. Dann, S. Cooper, and D. Slater. Alice 3.1 (abstract only). In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, pages 757–757, 2013.

[10] W. Dann, D. Cosgrove, D. Slater, D. Culyba, and S. Cooper. Mediated transfer: Alice 3 to java. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, pages 141–146, 2012.

[11] O. V. Deryugina. Chatterbots. *Scientific Technical Information Processing*, 37(2):143–147, Apr. 2010.

[12] K. Doran, A. Boyce, S. Finkelstein, and T. Barnes. Outreach for improved student performance: A game design and development curriculum. In *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education*, pages 209–214, 2012.

[13] S. Furber. Shut down or restart? The way forward for computing in UK schools. Technical report, The Royal Society, London, 2012.

[14] O. Hazzan, J. Gal-Ezer, and L. Blum. A model for high school computer science education: The four key elements that make it! *SIGCSE Bulletin*, 40(1):281–285, 2008.

[15] M. Keegan, R. D. Boyle, and H. M. Dee. Turi: Chatbot software for schools in the turing centenary. In *Proceedings of the 7th Workshop in Primary and Secondary Computing Education*, pages 153–154, 2012.

[16] C. Kelleher and R. Pausch. Using storytelling to motivate programming. *ACM Communications*, 50(7):58–64, July 2007.

[17] A. Kerly, P. Hall, and S. Bull. Bringing chatbots into education: Towards natural language negotiation of open learner models. *Knowledge Based Systems*, 20(2):177–185, Mar. 2007.

[18] D. Koller, A. Ng, C. Do, and Z. Chen. Retention and intention in massive open online courses: In depth. *Educause Review*, 2013.

[19] T. L. Naps, G. Rössling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger, and J. A. Velázquez-Iturbide. Exploring the role of visualization and engagement in computer science education. *SIGCSE Bull.*, 35(2):131–152, 2002.

[20] F. Newmann. *Student Engagement and Achievement in American Secondary Schools*. Teachers College Press, 1234 Amsterdam Avenue, New York, 1992.

[21] J. O'Kelly and J. P. Gibson. Robocode & problem-based learning: A non-prescriptive approach to teaching programming. *SIGCSE Bull.*, 38(3):217–221, 2006.

[22] H. Park, S. Khan, and S. Petrina. Ict in science education: A quasi-experimental study of achievement, attitudes toward science, and career aspirations of korean middle school students. *International Journal of Science Education*, 31(8):993–1012, 2009.

[23] A. Pears, S. Seidman, L. Malmi, L. Mannila, E. Adams, J. Bennedsen, M. Devlin, and J. Paterson. A survey of literature on the teaching of introductory programming. *SIGCSE Bull.*, 39(4):204–223, 2007.

[24] M. J. Prince and R. M. Felder. Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Journal of Engineering Education*, 95(2):123–138, 2006.

[25] A. Shaw. Using chatbots to teach socially intelligent computing principles in introductory computer science courses. In *Proceedings of the IEEE 2012 Ninth International Conference on Information Technology*, pages 850–851, 2012.

[26] M. M. Voyles, S. M. Haller, and T. V. Fossum. Teacher responses to student gender differences. In *Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, pages 226–230, 2007.

[27] C. Wilson. *Running the Empty: Failure to Teach K-12 Computer Science in the Digital Age*. Association for Computing Machinery, 2010.

[28] I. Zur Bargury. A new curriculum for junior-high in computer science. In *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education*, pages 204–208, 2012.