What's Next for Computing Education

Michael J. Lee
New Jersey Institute of Technology,
Informatics, Assistant Professor
mjlee@njit.edu

University of Washington(Ph.D)
UC Berkeley(M.S)
UC San Diego(B.S)

관심분야: 코딩교육, 인간-컴퓨터 상호작용, 가상현실, 머신러닝

Introduction

Computer programming is increasingly being recognized as an important 21st century skill. In recent years, computing education related resources have seen huge growth online. Potential learners have many resources to choose from, including massive open online courses (MOOCs), online tutorials, and educational games. Annual events such as The Hour of Code, provides curated links to these different types of online resources, attracting and encouraging millions of learners all ages to learn more about programming.

However, with all this interest in computing education, there is still a lot more we need to learn. As more people pursue computer programming skills, there are many challenges, as well as opportunities, to use and develop new methods for educating the masses. This may include the use of new technologies and media, both inside and outside of the formal classroom setting, that extend and complement the methods used today. As we continue to innovate, we have the potential to change how we teach both computer science and other topics in compulsory and discretionary settings.

Personalized Learning

As more people turn to online resources to learn, there will be an increasing need for systems to understand and adapt to the needs of their users. Personalization can include technologies such as customized interfaces, intelligent tutors, auto-generated curricula, and adaptive curricula. As we learn about, develop, and improve these technologies, we will have the opportunity to adapt them for classrooms as well.

Customized Interfaces

One major factor to consider in teaching effectively is to engage learners. One way to do this is to customize the learning interface and experience to the user. With the large amounts of data people generate on the web (e.g., sharing content on social media; purchasing products; product reviews; history of web activity such as videos watched), systems can detect and classify types of users into a general user model (e.g., personality traits).

Personality traits have shown to be a suitable general user model as it characterizes a person's thoughts, feelings, social adjustments, and behaviors, which subsequently influences their expectations, self-perceptions, values, attitudes, and

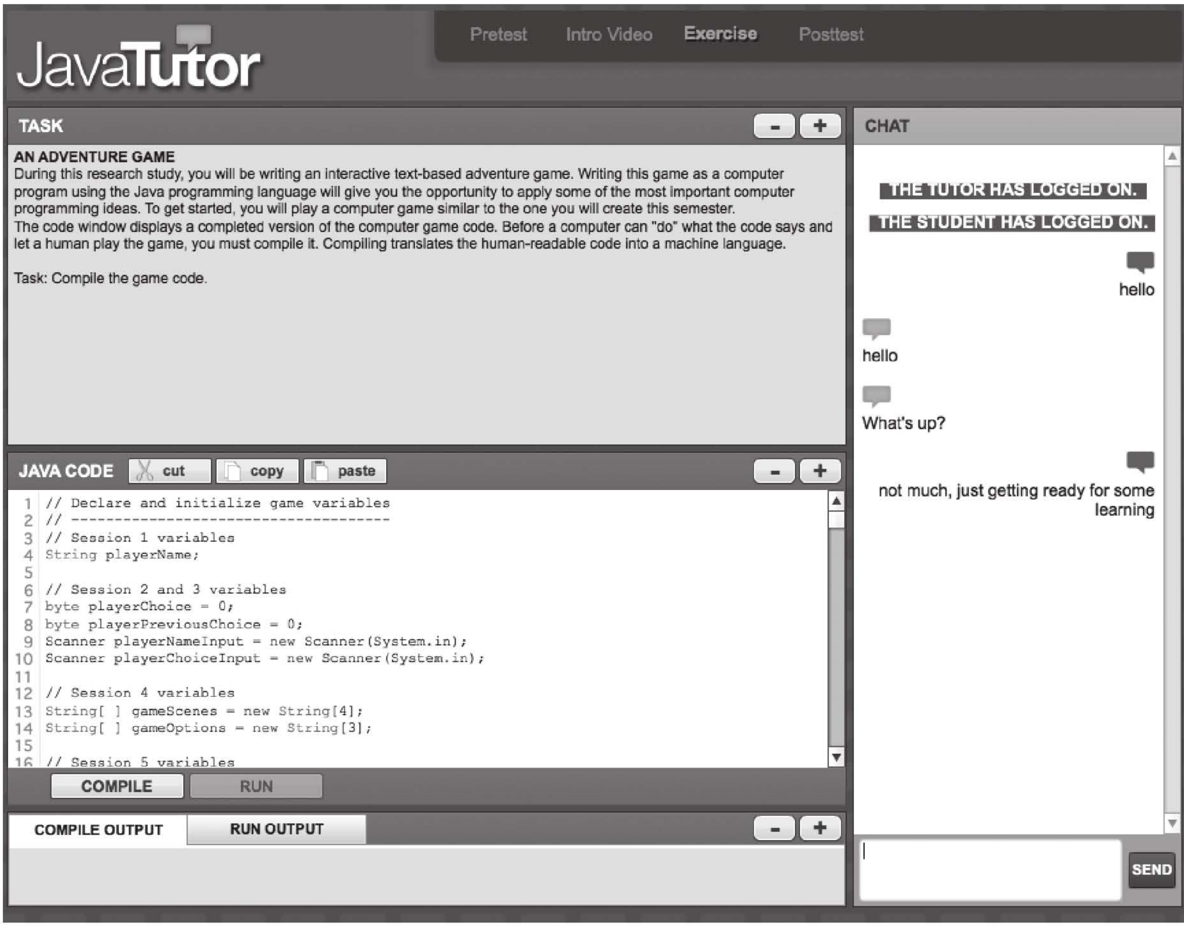

Fig. 1 Screenshot of JavaTutor, a natural language dialogue, intelligent tutoring system for novice programmers.

their reactions to others, problems, and stress. We can use these user models to customize the experience for learners, making the systems engaging, easier, and more enjoyable for them to use.

Intelligent Tutors

Intelligent tutoring systems (ITSs) are designed to model human tutors using artificial intelligence to engage students in sustained reasoning activity and to interact with the student based on a deep understanding of the students' behavior. An ITS can give customized feedback and content to learners based on their actions or questions navigating through the subject material (see Figure 1). Moreover, these systems can detect such behavior such as disengagement (or frustration) and provide interventions to keep learners engaged when necessary. Studies show that educational systems incorporating ITSs lead to positive learning outcomes for students in diverse topics such as computer programming, algebra, medicine, law, and reading.

There may never be enough teachers online to teach the millions of people wanting to learn new subjects such as programming. ITSs can fill this void by providing

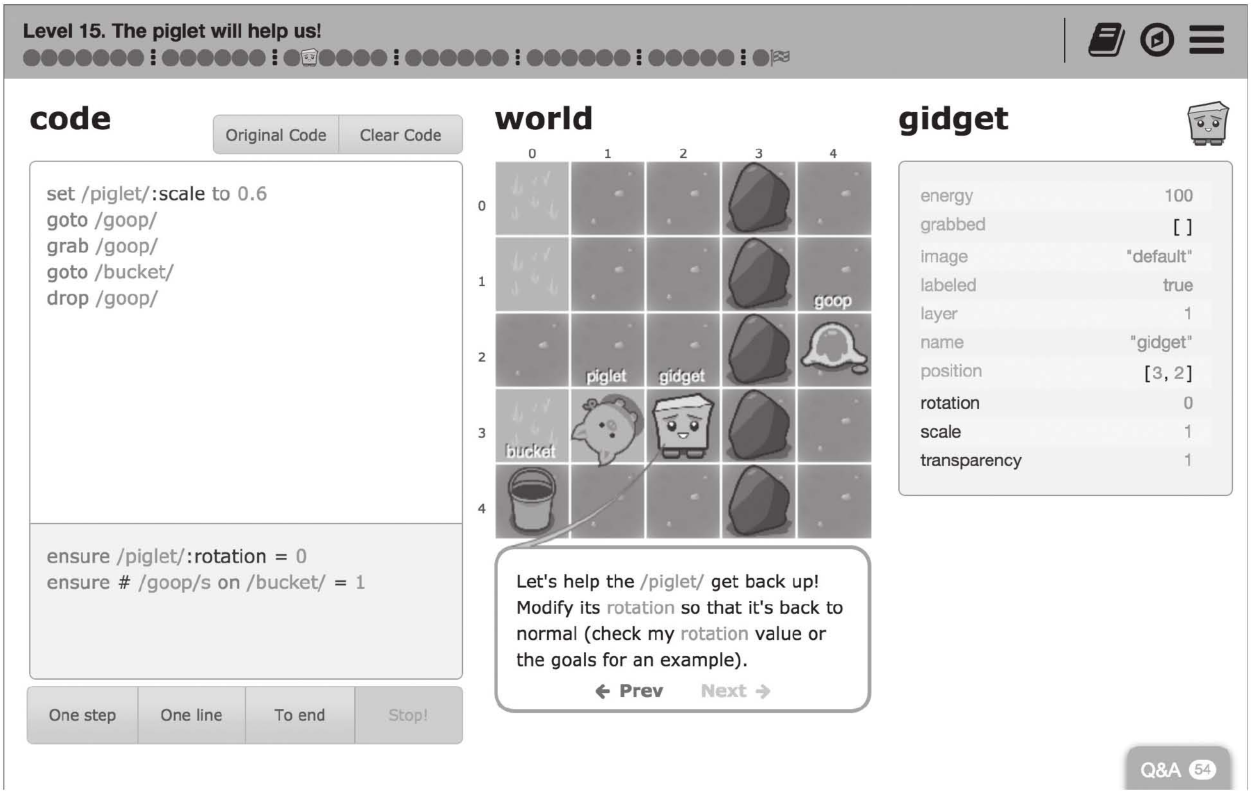

Fig. 2 Screenshot of Gidget, where players learn programming by solving debugging puzzles (www.helpgidget.org).The gameprovideslearners withadditionallevelsformorepracticewhennecessary.

immediate, just-in-time feedback, automatically generate content for additional practice (see Figure 2), and help when learners need extra support. Additionally, these types of systems continue improve over time, as new user data adds more examples for ITMs to draw from. Combining these capabilities with more information (e.g., general user profiles as mentioned in the previous section, and big data sources) can improve the accuracy and utility of these systems even further.

Adaptive Curricula

Finally, structured courses, such as MOOCs and those in schools, can benefit by adapting content for their students. For example, Teach2One is an existing program in the USA that consists of open-space middle school classroom, where each students' daily mathematics performance is measured against a skills and concept map by an external team of educators, who then modify the contents of each students' mathematics schedule for the following day.

Here, the on-site teachers teach collaboratively, supporting more students than they would individually, facilitating students' learning through multimodal instruction at different working stations. This system helps to ensure that a student has the necessary knowledge to move on to more advanced or related content, and to teach the necessary information in cases where he/she does not have it.

What is Next?

The technologies and techniques described above present exciting potential to computing education and education in general. As these technologies become more mature and widely adopted, we may see a better educated and informed world.

The Future Classroom

K-12 teachers will continue to have an important role in compensatory education. Future teachers will have more data about their students than ever before.

As the use of computing devices (such as touch tablets) in the classroom becomes more ubiquitous, ITS or ITS-like systems will use the continuous stream of student-generated data about their assignments and exams (and possibly other measures such as what time of they are most active, which assignments or exam questions took the longest for them to complete, etc.) to inform teachers and students immediately about where the latter is excelling and failing, and give directed feedback/suggestions about how to address these issues.

This automated assessment of students' needs, along with an adaptive curriculum as described in the Teach2One example above, can lead to individual, customized learning even within large groups of students. These adaptive curricula may help students identify and pursue the subjects they are more interested in, better preparing the next generation of people for the 21st century and beyond.

Conclusion

However, before we can reach the future classroom, we need to address many of the issues currently facing computing education (and more generally, STEM education) to make sure we are providing equal access and opportunities to everyone.

For example, many countries are now requiring computing education in their K-12 (primary and secondary school), but there are currently many issues such as: teaching credentials (e.g., who are going to be teaching the computer science courses? How will we train them?), cost (e.g., how are we going to pay for the space, equipment, and teachers?), and curricula (e.g., what do we need to teach?).

In addition, we need to think beyond K-12 and classrooms. How do we attract more minorities and women into computing and STEM? How can we dispel some of the negative preconceived notions about computing? How do we include people who may not have ready access to computing? How do we teach adults? How do we teach most effectively online?

Although we do not have answers to many of these questions yet, there are many educators, researchers, academics, politicians, and other leaders currently exploring these issues and how to solve them. With every new advance and piece of knowledge, we get a little closer to the ideal of teaching the people of the world (any subject), both inside and outside the classroom, in new and exciting, data-driven ways that are both engaging and effective.