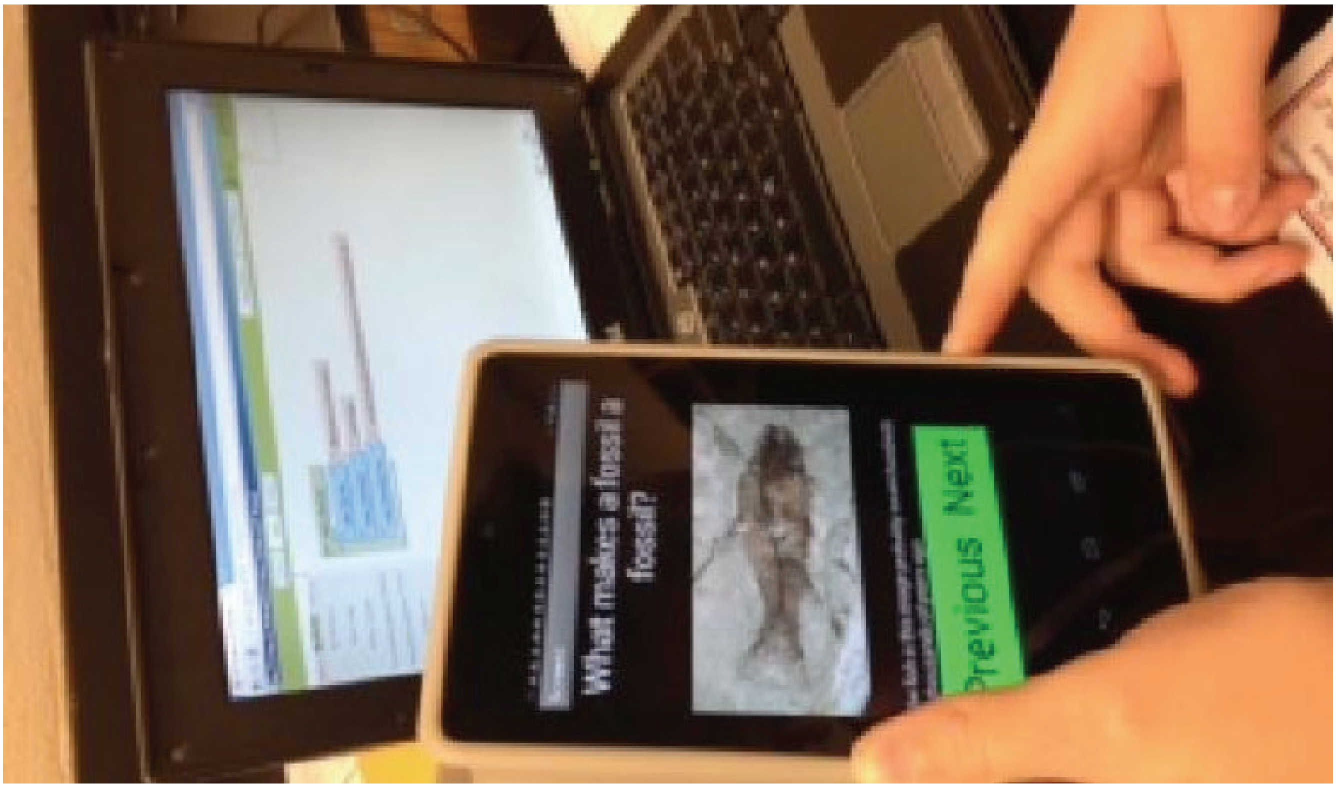
Adolescents and App Development in Middle School Classrooms

Danielle Herro, D. Matthew Boyer and Christina McCune-Gardner, Clemson University
dherro@clemson.edu, dmboyer@clemson.edu, gmccune@clemson.edu



Example URL: http://www.workingexamples.org/example/show/630

Our research involves investigating the viability of teaching computational thinking (CT) to middle school students through application (app) development. Computational thinking is a way of devising, decomposing, and designing ways to solve problems. Many computer scientists and educational researchers consider CT foundational skills for everyone, believing they complement core elements of computer science, and attend to human-computer interactions involving creativity, innovation, collaboration, aiding in structuring and solving problems efficiently (Papert, 1996; Wing, 2006). New guidelines for computer science now include measures to consider CT. Computational thinking has no firm definition, but it is broadly defined as a set of thinking practices characterized by conceptualizing ideas, engineering solutions, and thinking at multiple levels of abstraction (Wing, 2006). It is supported through teamwork, creativity, human interaction with computers, increased visualization, and it aims for global impact – to make a difference in the lives of others (The College Board, 2011). For this research, we draw on Barr and Stephenson's (2011) definition of computational thinking as a set of techniques that include: problem decomposition, pattern recognition, pattern generalization to define abstractions or models, and algorithm design. These skills are markedly different than traditional computer science skills of decades past that were narrowly focused on knowledge and skills related to technical programming (CSTA, 2005).

The fundamental need for computer science knowledge in an information-based society is being reframed as a need for everyone to learn computational thinking (Yadav, Mayfield, Zhou, Hambrusch, & Korb, 2014). Ostrachan (2012) portrays coding as a metaphor for computational thinking, detailing the ways writing and interpreting code is integrated in everyday life. Coding has recently gained popularity as a way to encourage computational thinking (CSTA, 2012), as evidenced by large-scale initiatives such as Code.org which hosted CS Education Week (http://csedweek.org/) in December, 2013, introducing code to more than 31 million students.

The App Development Platform

MIT App Inventor (http://appinventor.mit.edu/explore/) is an open source, visual programming platform that allows users to create Android apps by snapping blocks in place. Conceived as an easy entry into intuitive, incremental and logical programming, it consists of two main elements: a Design view used to select components of an app, and a Blocks Editor to program behaviors (Pokress & Veigra, 2013). Google originally developed and maintained the software in 2010; it is now hosted by Massachusetts institute of Technology (MIT).

MIT App Inventor (also referred to as "App Inventor") was created with educators and learners in mind, with a goal of increasing interest and skills in computational practices (Pokress & Veigra, 2013). It is novel, and thus has limited but growing research exploring its capacity in college classrooms (Wolber, 2011; Abelson, Morelli, Kakavouli, Mustafaraj, & Turbak, 2012) with teachers in summer camps (Hsu, Rice, & Dawley, 2012) or its potential to bring computational thinking (CT) to K-12 students (Morelli et al., 2011).

Seed

Tell us about your idea or project. What's your vision?

Four prevailing realities inform our vision: (1) the current demand for those skilled in computer science and CT outpaces student training and graduates by roughly 2:1 (US Department of Education, 2012); (2) middle schools typically do not offer computer science curriculum due to low student interest and lack of resources (CSTA, 2011, p. ii.); (3) recent guidelines have shifted to include CT practices supported through teamwork, creativity, human interactions with computers, increased visualization, and societal impact (The College Board, 2011); and (4) outside of school youth regularly participate in media-rich production spaces emulating some of the abovementioned CT practices (Jenkins et., al, 2006). Additionally, a wealth of research supports programming and designing with games and apps as a means to foster complex problem solving, logic and reasoning, systems thinking, and creativity (Gee, 2003; Ketelhut, Dede, Clarke, & Nelson, 2006; Klopfer, Osterweil & Salen, 2009; McClarty et al., 2012; Williamson, Squire, Halverson, & Gee 2005).

Providing and researching opportunities for students to hone CT, and for teachers to develop ways to embed CT practices in curriculum, offers a way to meaningfully address this need.

What problem are you trying to solve and why does it matter?

Our goal is to offer middle school students an avenue to practice CT while investigating the following questions:

(1) What are teachers' perceptions regarding app creation to teach computational thinking?

(2) What are students' attitudes and beliefs about computational thinking? Do they change after participating in app-design curricula?

(3) What evidence links app development to the CT skills of problem decomposition, pattern recognition, pattern generalization and algorithm design?

Project collaborators include professors of Digital Media and Learning and Computer Science, and participating classroom teachers across three school districts. The final phase our project (detailed below) will include researchers from the Educational Testing Service (ETS). Our initial challenge is navigating the demands and logistics of schooling while providing a worthwhile experience, and collecting meaningful data to inform our project moving forward. This working example will discuss both the process and challenges of our endeavor.

What challenges might pop up?

Working Examples: Feedback and Support from the Community

We seek feedback from the community regarding the viability of this enterprise. We anticipate challenges creating game-like assessments (see Phase 3 in Sprout) and appropropriately defining "indicators" to measure CT practices. Thus, we are especially interested in feedback and support from those who have experience in CT and app development, or have created in-game assessments.

Using the "Worked Example" model (Gee, 2010) we present media-rich (video, images and hyperlinks) curriculum examples, data, and our early findings. We propose conditions necessary to garner support when implementing app-based curricular platforms in school by revealing processes, successes, barriers, and failures. It is in "building plausibility arguments via proof-of-concept implementations" or exemplars, that a broad interdisciplinary discussion can ensue, and serve to unify and inform research and practice (Barab, Dodge & Gee, 2009, p. 18).

Sprout

Tell us about your process and how your idea is evolving throughout the project.

Three phases encompasses our goals in this project.

Phase 1: A pilot project was conducted in two at-risk 8th grade classrooms helping us better understand the viability of using MIT App Inventor in schools. Technical requirements, logistical concerns and curriculum integration were addressed, and data were collected to gauge teacher and student perceptions of the experience. Measures included pre and post-test surveys from 17 boys, 18 girls, and 2 participating teachers, observations, focus group interviews, and student artifacts. Data were analyzed and triangulated describing teacher and student perspectives and practices. The pilot experience suggested (1) students' had limited knowledge of CT before the intervention; (2) students' understanding of, and attitudes towards, CT improved after the unit; (3) students' believed the unit motivated them to continue developing apps, and (4) boys showed greater gains in understanding and attitudes. Teachers' perceived the unit as valuable and intend to offer additional classroom opportunities. They suggested teacher training, scaffolded lessons, and addressing logistical concerns are necessary for effective practice.

Rethinking challenges and failures. Two challenges and "failures" surfaced during this phase. The working environment for students posed numerous challenges including network issues related to Wi-Fi and dated school-owned laptops. Behavior issues were exacerbated by the room configuration and additional distractions created waiting for technical issues to be resolved. Our team, and participating teachers, agreed the curriculum needed revision to better scaffold lessons. Students struggled moving from simple to complex app creation. Simple apps involved fewer components and design elements, complex apps typically had a number of variables requiring increased decomposition, pattern recognition and logic in coding. To address the challenges moving forward, we spent more time on the "back-end", ensuring technical support, specifying minimum hardware and software requirements and testing connectivity. Many of our challenges were remedied by better communication. We also re-thought scaffolding between difficulty levels of apps, adding components and variables at a gradual pace.

Phase 2: The second, in-progress, phase is being conducted in two regular education classrooms. It entails examining 8th grade students' and teachers' perceptions of CT before and after an intervention designing apps, and further studying whether evidence exists linking four pre-determined CT to student learning. Specifically, we are investigating indicators of CT by student engagement in problem decomposition, pattern recognition, pattern generalization to define abstractions or models, and algorithm design. Indicators of CT are currently being coded and mapped to themes from pre and post-test surveys, focus group interviews, observed activity, and student artifacts our research team independently to reach consensual validation (Eisner, 1991, as cited in Creswell, 2007, p. 204).

Describing the current intervention. Within the 4-week, daily fossil unit App Inventor was taught 3 times per week. Students used their fossil research to plan and design a story-telling app including pictures and text supporting their essential question, with buttons programmed to move from page-to-page on the app. A desktop sized mat emulating the various panels and components in App Inventor was used to plan and storyboard learning. A typical class period included a 10-20 minute mini-lesson using the mats or other learning resources, followed by 25-30 minutes of students working in App Inventor at their individual computers. Towards the end of the unit, students spent a majority of their time at their computers working on coding, playtesting, and problem solving. All (n=57) students completed a working app. Figures 1 and 2 depict representative student work.

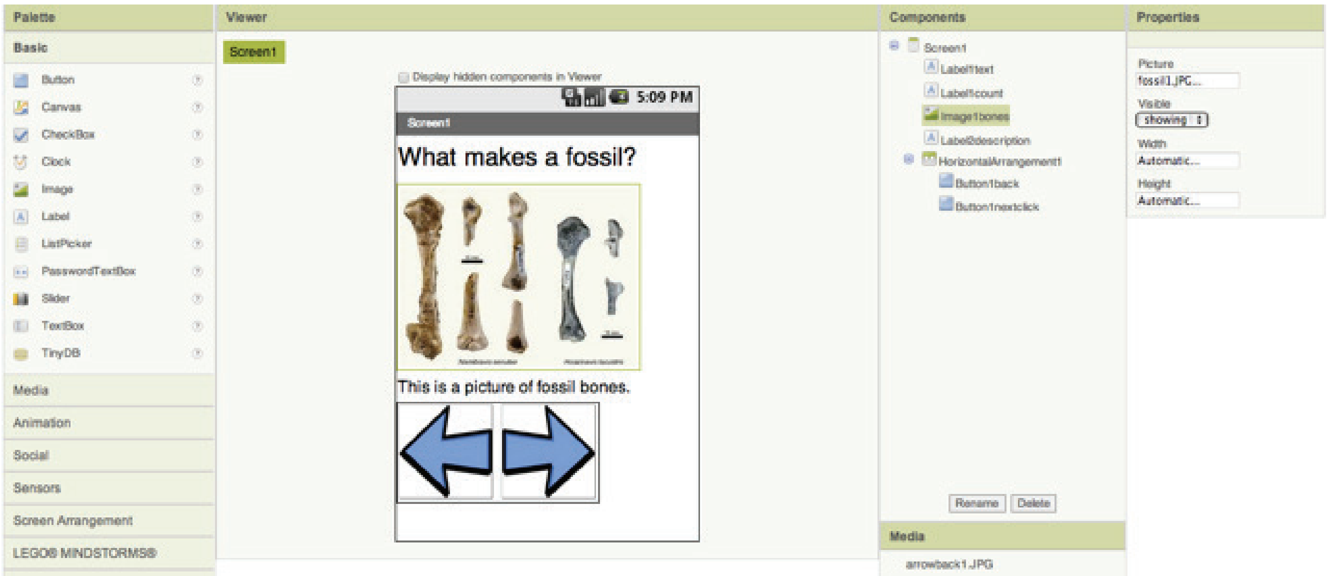


Figure 1: Screenshot of student's app in Designer view using MIT App Inventor

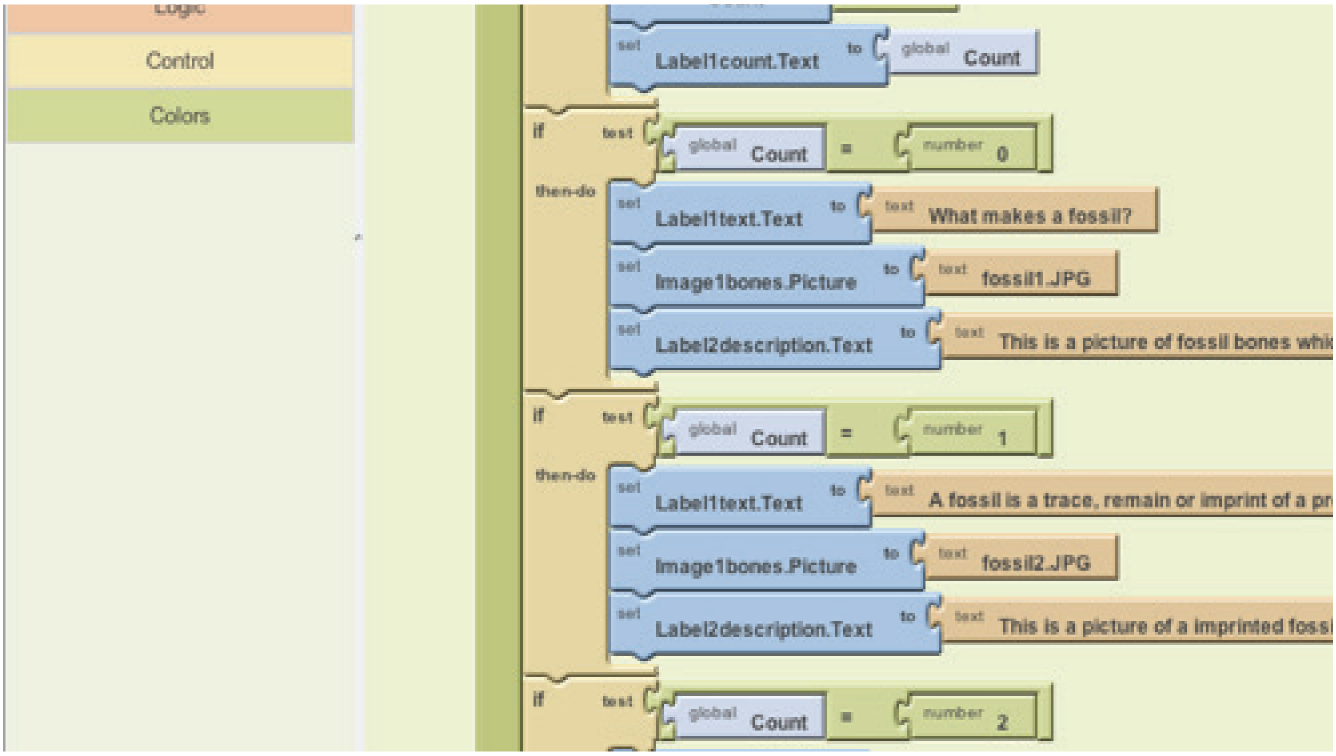


Figure 2: Student coding in Blocks Editor view of MIT App Inventor.

Phase 3: The third phase will consist of expanding app development units to include an assessment measuring CT using a game-like platform. We are partnering with research scientists from ETS, and pursuing external funding; our research proposes creating a game-like model with an embedded test administered before, and then after, students participate in an app creation unit.

Working Examples: Feedback and Support from the Community

We seek feedback from the community regarding the viability of this enterprise. We are especially interested in feedback and support from those who have experience in CT and app development, or have created in-game assessments.

Using the "Worked Example" model (Gee, 2010) we present media-rich (video, images and hyperlinks) curriculum examples, data, and our early findings on http://www.workingexamples.org/. We propose conditions necessary to garner support when implementing app-based curricular platforms in school by revealing processes, successes, barriers, and failures. It is in "building plausibility arguments via proof-of-concept implementations" or exemplars, that a broad interdisciplinary discussion can ensue, and serve to unify and inform research and practice (Barab, Dodge & Gee, 2009, p. 18).

In this context, the curriculum design provides a model towards app design to further CT practices in school. This invitation for scholarly conversation and critique of curriculum components, its appeal to youth and potential next steps - including studying achievement indicators within the curricular interventions using game-like assessments, create a common foundation for collaboration across disciplines adding to the plausibility of the thesis.

References

Abelson, H., Morelli, R., Kakavouli, S., Mustafaraj, E., & Turbak, F. (2012). Teaching Cs0 with mobile apps using app inventor for android. *Journal of Computing Sciences in Colleges,* 27(6), 16-18.

Barab, S., Dodge, T., & Gee, J.P. (2010, April). *The worked example: Invitational scholarship in service of an emerging field.* Paper presented at the annual meeting of the American Education Research Association, San Diego, CA.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. *ACM Inroads, 2*(1), 48-54.

Computer Science Teachers Association (2005). The new educational imperative. Retrieved from http://csta.acm.org/Communications/sub/DocsPresentationFiles/White_Paper07_06.pdf

Computer Science Teachers Association (2011). CSTA K-12 Computer Science Standards. Available at: http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf

College Board (2012). Computer science principles: Computational thinking. Big ideas, key concepts and supporting practices. Supported by the National Science Foundation, grant CNS-0938336.

Creswell, J. W. (2007). Qualitative inquiry and research design: *Choosing among five approaches* (2nd Ed.). Thousand Oaks, CA: Sage Publications.

Hsu, Y.-C., Rice, K. and Dawley, L. (2012), Empowering educators with Google's Android App Inventor: An online workshop in mobile app design. *British Journal of Educational Technology,* 43: E1–E5. doi: 10.1111/j.1467-8535.2011.01241.x

Gee, J. P. (2003). *What video games have to teach us about learning and literacy.* New York, NY: Palgrave Macmillan.

Gee, J. P. (2010). *New digital media and learning as an emerging area and "Worked Examples" as one way forward.* The John D. and Catherine T. MacArthur Foundation Reports on Digital and Media and Learning. MIT Press.

Jenkins, H., Clinton, K., Purushotma, R., Robison, Al., & Weigel, M. (2006). *Confronting the challenges of participatory culture: Media education for the 21st century.* The MacArthur Foundation: Chicago, Illinois.

Ketelhut, D. J., Dede, C., Clarke, J., & Nelson, B. (2006, April). A multi-user virtual environment for building higher order inquiry skills in science. Paper presented at the 2006 AERA Annual Meeting, San Francisco, CA. Retrieved from http://muve.gse.harvard.edu/rivercityproject/documents/rivercitysympinq1.pdf

Klopfer, E., Ostewil S., and Salen K., (2009). *Moving learning games forward: Obstacles, opportunities and openness.* Cambridge, MA: The Education Arcade.

McClarty, K. L., Orr, A., Frey, P. M., Dolan, R. P., Vassileva, V., & McVay, A. (2012). *A literature review of gaming in education.* Pearson Research Report.

Morelli, R., de Lanerolle, T., Lake, P., Limardo, N., Tamotsu, E., & Uche, C. (2011). Can Android App Inventor bring computational thinking to K-12. In Proc. *42nd ACM technical symposium on Computer science education* (SIGCSE'11).

Ostrachan, (2012). Code as a metaphor for computational thinking. Retrieved from https://www.cs.duke.edu/csed/talks/vatech/vtech2012.pdf

Papert, S. (1996) An exploration in the space of mathematics educations. International Journal of Computers for Mathematical Learning, Vol(1)1. pp. 95-123.

Pokress, S. C., & Veiga, J. J. D. (2013). MIT App Inventor: Enabling personal mobile computing. Retrieved from http://lib-arxiv 008.serverfarm.cornell.edu/pdf/1310.2830.pdf

U.S. Department of Education, Office of Instructional Technology (2010). *Transforming American education: Learning powered by technology.* Washington D.C.

Williamson, D., Squire, K., Halverson, R., & Gee, J. P. (2005). Video games and the future of learning. *Phi Delta Kappan, 87*(2), 104-111.

Wing, J. (2006, March). Viewpoint: Computational thinking. *Communications of the ACM.* (49) 3.

Wolber, D. (2011, March). App inventor and real-world motivation. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 601-606). ACM.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. ACM *Transactions on Computing Education* (TOCE), 14(1), 5.