Youth Apps Challenge Builds STEM Interest

Peter Gruenbaum
President
SDK Bridge
1035 NE 94th St
Seattle, WA 98115 USA
peter.gruenbaum@sdkbridge.com

Earl Bergquist
Computer Science Teacher
Garfield High School
400 23rd Ave
Seattle, WA 98122 USA
embergquist@seattleschools.org

Karen Manuel
STEM Outreach Director
Technology Alliance
1301 5th Ave., Ste. 1500
Seattle, WA 98101 USA
karen@technology-alliance.com

Abstract

The Youth Apps Challenge builds students interest in computer science and entrepreneurship through an app development contest. In its inaugural year, the contest received 75 entries from 302 middle and high school youth in Washington State. The Technology Alliance trained 40 teachers who were interested in teaching app development to middle and high school youth—most with little or no programming experience. The day-long training consisted of an introduction to the curriculum as well as hands-on exercises using Scratch, idea generation, paper prototyping, wireframe development, and App Inventor. The Technology Alliance focused on recruiting teachers from diverse, high-need schools. The 42-hour curriculum was taught to students with little or no programming experience via in-school courses and after school programs. Although a formal evaluation is yet to be conducted, initial informal data show increased student interest in taking additional programming classes.

About the Youth Apps Challenge

In May and June of 2014, the Technology Alliance held its first ever Youth Apps Challenge, a statewide competition designed to build student interest in computer science education and careers. The Challenge was open to middle and high school students in Washington State. The Technology Alliance received 75 app submissions from 19 schools and non-profit teams, with 302 students from across the state participating.

The Technology Alliance is a non-profit organization that supports the growth of Washington's high-impact industries through programs, events, data analysis, and policy activities.

The Curriculum

The curriculum, which is based on Apps for Good's (www.appsforgood.org) curriculum is comprised of 42 one-hour sessions that includes idea generation and selection, user interviews and profiles, paper prototypes, wireframe designs, and creating a pitch video. In addition, the curriculum included sessions to introduce students to computer science concepts, such as working with Scratch (http://scratch.mit.edu) and the Hour of Code from code.org (http://code.org). To learn App Inventor, students were led through some of the App Inventor tutorials, and then were given several sessions to build their app. Of

the 42 sessions, 5 sessions covered introduction to computer science concepts, 3 sessions covered idea generation and selection, 3 sessions covered understanding users, 7 sessions covered prototyping, 3 sessions covered users, 18 sessions were devoted to App Inventor, and the final 4 sessions were used for preparing for the contest submission. The curriculum was mapped to the standards.

A day-long training was provided to 40 teachers who were interested in teaching app development to middle and high school youth; and most of them had little or no programming experience. The training consisted of an introduction to the curriculum as well as hands-on exercises using Scratch, idea generation, paper prototyping, wireframe creating, and App Inventor. The Technology Alliance focused on recruiting teachers from diverse, high-need schools.

Contest Details

The contest was open to all middle and high school students in Washington State, but had several requirements. Submissions could be either general (just including information about the app idea and design) or technical (including a working app). Technical submissions did not have to be fully functional, but could be prototypes that demonstrated what a fully functional app would look like. Although the curriculum used App Inventor, any technology was accepted for the technical submissions. Students were encouraged to work in teams, but it was not a requirement.

Students submitted a presentation describing their development and research process, a one minute pitch video, and a wireframe design (i.e., visual design). In addition, for technical submission, they submitted both the finished app and their code. For App Inventor submissions, this meant submitting an .aia file. The presentation needed to include a problem statement, descriptions of user personae, an image of a paper prototype, and an image of the wireframe design.

The submissions were rated by judges on a scale of 1 to 4 in several areas. For all submissions, the judges were asked to rate how well the app solved a real-world problem, how well the submission illustrated how the app would work, how feasible the concept is in terms of data and content, and how well the team has described the app's users and market. Technical submissions were judged on how well the app functioned, how usable it was, how much technical complexity it contained, whether it used good coding practices such as comments and abstraction. The judges primarily came from the Technology Alliance's Ada Developers Academy (http://adadevelopersacademy.org/), which is a year-long intensive training program for women transitioning into software development. A web portal was created where teachers could sign up for accounts and submit their students work. Judges could then log in to rate each submission.

Eastern and Western Washington have very different cultures; where Western Washington has a very strong technology industry, and Eastern Washington is much more agricultural and racially diverse. In order to provide opportunities for all, separate awards, including tablets, were given for Eastern and Western Washington. In addition, separate awards were given for middle and high school submissions, technical and non-

technical submissions, and for the level of experience each team had with computer science.

Contest Results

We received a total of 75 submissions, with 17 technical and 58 non-technical. Feedback from teachers was that they often ran out of time before students could get to creating the actual apps. We need to adjust the timing and the curriculum so that we make sure they have time in the future. Of the technical submissions, 13 were mobile apps, and 3 were games; also, 13 used App Inventor, one was a web application built in Python, one was a Windows Phone application built in C#, and one was a game built in C# using the Unity Game Engine.

Some technical submissions were simply translations of the wireframe design into App Inventor, providing an experience like going through static web pages, and not making full use of what App Inventor is capable of. However, other apps used features such as timers, sounds, variables, data storage, GPS, and texting. The most ambitious use of App Inventor was an app to send out an emergency text if you were trapped somewhere. It displayed your position on a Google map, and also stored emergency numbers using the TinyDb feature, as well as sent actual text messages with the GPS position.

Teacher Experience

Teachers implemented the curriculum either in in-school or after school environments. Garfield High School teacher, Earl Bergquist, integrated teaching App Inventor in his single semester Introduction to Computer Science / Creative Programming class. He observed his first year teaching that a subset of his students, especially higher needs students, did not think that they would be using personal computers in their future career or life and were not very motivated to learn programming on classroom PCs. Developing apps using tablets and App Inventor engaged students. He observed the focus on making mobile apps combine with the tactile use of physical tablets gave the exercises more relevance and stimulated kinesthetic learning.

Students were able to transfer their programming skills from Scratch to the App Inventor environment easily. To assist and provide scaffolding for their progress, Earl created AIA template files to show the alternative approaches to accomplish page follow and pass information between screens. He looks forward to using App Inventor next year, and will be following the Mobile Computer Science Principles course (https://sites.google.com/a/css.edu/mobilecsp/) this summer to enhance his semester class and prepare for the full CS Principles class in the future.

Almost all of the teachers that taught apps development this year plan to offer the curriculum next year. The teachers offering Youth Apps curriculum after school plan to start their program earlier in the school year to allow more time for app development and contest preparation. The Technology Alliance plans to evaluate the apps program next year to gain a better understanding of how the initiative impacts students' STEM interest and offer additional teacher training.