Cal State LA integrates Applied CS into mobile programming course offering

CAL STATE LA
CALIFORNIA STATE UNIVERSITY, LOS ANGELES

At a Glance

What they wanted to do

- Enhance an existing CS mobile programming course offering by incorporating Applied CS workshops as 6-week introduction

What they did

- Adapted a pre-existing mobile programming class for seniors and graduate students to create a hybrid course with 6 weeks of Applied CS workshops followed by 4 weeks of group app development
- Gave students the opportunity to practice programming on projects under tight time constraints

What they accomplished

- Provided a comprehensive, hands-on, for-credit Android programming experience for more than 50 students
- Enabled students to leverage algorithms and data structure skills to build Android apps

Challenge

California State University, Los Angeles is located in the heart of Los Angeles. It offers undergraduate and graduate courses for more than 27,800 students. The rapidly growing computer science (CS) department currently serves more than 500 students, offering BS and MS degrees.

Given the rapid – even accelerating – changes in computing software and technologies, the department is always looking for ways to offer its students access to the most up-to-date technologies and employment skills, including in subjects like mobile development. Computer Science professors recognize that there is sometimes a trade-off between theoretical or foundational education versus offering hands-on coding training, simulating an industry coding environment, in the classroom. There typically isn't enough time in a semester to teach or develop new projects and curriculum to address both aspects to programming.

Solution

With the support of CS Department Chair Professor Raj Pamula and Professor Mark Sargent, the Cal State LA CS department adapted their 2015 mobile development course for Spring 2016 to take advantage of the Applied CS with Android content from Google. They divided the course into two portions – the first six weeks used five of the Applied CS units plus the "Code Sprint," and the final four weeks were devoted to having students work in small teams to design, scope, build and launch Android apps as the final course project.

> *"We had students work in teams under the mentorship of Google engineers to deliver something under a time crunch. This helped them appreciate what urgency looks like in a real-world setting."*
> *—Mark Sargent, Professor, California State University, Los Angeles*

In the first half of the class, students met with Professor Sargent and industry engineers for four hours each week to build the Applied CS activities in Android Studio. The 2015 course had spent the first half learning Android Studio SDK, so the added benefit of project-based work and implementing algorithms and data structures (even some that were new to the students) from Applied CS made this version of the course more dynamic and robust. During the "Code Sprint," students worked in small groups to create a simple game app using a data structure/algorithm from the Applied CS curriculum. This time-constrained and team-focused project eased students into working on full app and product design in the second portion of the course, and helped them connect the dots between knowing what they want to achieve and knowing what CS tools they need to build it.

About California State University, Los Angeles

- Founded in 1947 in the heart of Los Angeles, Cal State LA has more than 27,800 students
- Cal State LA is ranked #35 on the US News & World Report's 2017 college rankings list. The College of Engineering, Computer Science, and Technology is ranked 7th in the nation among non-doctoral public universities
- Cal State LA was ranked #1 in the nation for upward mobility by the recent Equality of Opportunity project www.equality-of-opportunity.org

www.calstatela.edu

Benefits

Hybrid course model

By integrating the Applied CS workshops into the mobile development course, the class adopted a hybrid model: structured workshop activities early in the course, all building the same Applied CS activities on Android, and more open-ended app development group work during the second part. Since the Department already offered a course in mobile computing, it was easy to adapt that course's framework to reflect the new course structure using Applied CS.

Additionally, instead of making a series of Applied CS workshops supplementary to coursework or adjunct to the 2015 course, integrating it into the course itself drove stronger attendance, active participation and completion. The students said the exercises were fun and made them better programmers.

Creating industry-modeled coding environment in the classroom

One of the biggest benefits students got from the course was training in writing code fast and working within a short development cycle. Most working programmers have tight deadlines and other constraints, such as having to use existing code bases and tools. For some projects, students got no more than four hours to finish the assignment. "We encouraged students to work together to deliver something under a time crunch," says Professor Sargent. "This helped them appreciate what urgency looks like in industry versus in the classroom."

Students also learned to write more efficient code, which is often a key requirement in the industry. Often, students create programs however they can, crafting long sequences of code that use brute force to solve a problem. But these solutions will create challenges and might not work out when run-time efficiency matters. When students are not pushed to consider computational efficiency, they might not use the most efficient algorithms or options – even though they may have learned about computational complexity in the algorithms course.

Because the professors invited industry engineers as teaching assistants to help students learn how to build Android games, students got a chance to see how professional engineers solve coding problems, write efficient code and debug mistakes – giving them valuable insight into the high standards for coding in the tech industry. "The (Applied CS) projects were great because the students had a chance to interact with real programmers," says Professor Sargent.