

Course Identifier: (e.g. TLC801) CSE 491-003<sup>1</sup>

Course Name: Mobile Application Development

Department: Computer Science and Engineering College: Engineering

Primary contact name, phone number, and email (*normally this will be the lead instructor*)

Charles B. Owen, 353-6488, cbowen@cse.msu.edu

Faculty and Staff Involved in Developing and Offering the Course *please list full name, position at MSU, email address, and project role for each person*

NAME	MSU Affiliation	PROJECT ROLE
Charles B. Owen	Associate Professor	Instructor/Course Developer

Which Competition Are You Entering (select one):

FULLY ONLINE COURSE (no required face to face component)

BLENDED/HYBRID OR FLIPPED COURSE (some face to face learning is replaced by online learning)

TECHNOLOGY-ENHANCED LEARNING INNOVATION (one specific technology innovation in a face-to-face, blended, flipped, or online course)

Semester(s) offered in 2012-2013 and number of students enrolled:

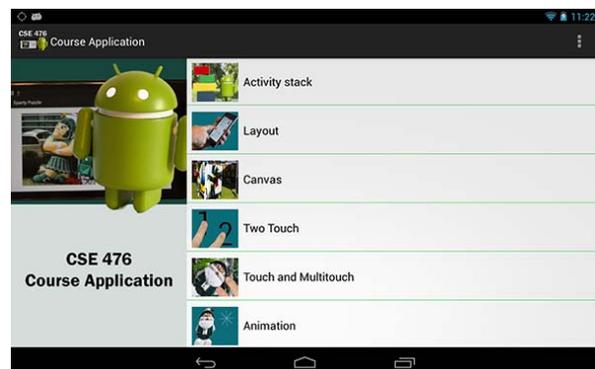
SEMESTER	# STUDENTS
Spring, 2013	40

Please address these categories:

#### I. Course Description (400 word limit)

*(broadly, what does the course teach, how is student performance assessed, what aspects of the course are online or technology-enhanced)*

This entry describes innovative technological enhancements that are being used to teach students the unique methods and requirements for programming mobile computing devices. These enhancements are a custom Android app, the *CSE 476 Course Application*, and an *advanced integrated online presence*. In 2012, the Computer Science and Engineering department committed to a new course in Mobile Application Development. The explosive growth of smart phones and apps has created a great demand for programming professionals with the

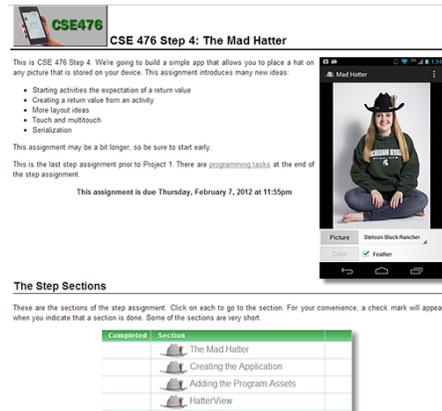


<sup>1</sup> This entry will refer to the course as CSE 476. CSE 491 is a number for courses in development and will be used until the final course number of CSE 476 is formally approved.

unique skills necessary to develop applications for these devices. I was approached to develop that course for a first offering in Spring, 2013 and an annual offering thereafter as CSE 476.

Mobile application development differs considerably from that of traditional desk-top applications. Programs must work seamlessly over a wide range of screen sizes, orientations, and aspect ratios. Touch and multi-touch are integral aspects of the application design and the devices have unique sensors such as accelerometers and gyroscopes. Battery life and power consumption are critical aspects of design. Applications can use GPS and other methods to be location-aware. Most applications are actually integrations of components running on the devices and the cloud. Students face a diverse array of programming languages and methods from XML layouts to Java application code to Java Server Pages, PHP, and SQL components on servers.

The course is programming-intensive, as the only way to develop the skills necessary to program these devices is through practical experience. Lecture sessions consist of problem solving, demonstrations, in-class assignments called *ToeTippers*, and the CSE 476 Course Application, an Android app students install from Google Play. *Step assignments* are online assignments consisting of a combination of guided tutorial and individual student. Students are guided through the construction of an interesting application and must complete key elements of the program themselves. In addition, there are three projects and a practical examination.



Since every student in the class has an Android device, they are able to utilize the Course Application, heretofore referred to as the "app", during the class period. The app provides demonstrations that tie into content presented in the lecture and problem-solving exercises that students can complete in class to help understand complex issues such as the activity stack, layout classes, and the drawing canvas. Since the app is installed via Google play, new activities can be added and updates pushed to student devices on a regular basis as the course is developed (Owen, 2013).

## II. Learning and Interaction Goals of the Course or Technology-enhanced Innovation (*what learning and interaction outcomes did you hope to achieve in your use of technology, why is this an award-worthy course or technology-enhanced learning innovation*)

The course makes extensive use of technology to create an experiential learning environment. Mobile application development is a cross-development activity. The development tools run on conventional desktop systems including Windows and Macintosh computers. Devices are connected via USB to these systems and applications are installed over that link for testing and debugging. Technological innovations in the course therefore exist in both the custom course application that runs in the device and the course website, which houses the interactive step assignments, along with all lectures and supplemental course material. The app on the device instructs students in device capabilities and can be used both in and out of the classroom, whereas the online presence instructs students in development methods and is used at home or in the labs.

The fundamental theoretical basis for these technological advancements is application of constructionism theory (Papert, 1980). There is significant evidence that students learn most effectively when they are active in creating tangible items in the real world. It has been shown

that even apparently theoretical concepts in Computer Science can be more effectively taught if students are producing a game that can apply or demonstrate those concepts (Korte et al, 2007). It has been my experience in general that students are much more eager to learn if they are building programs that they can relate to and use. I have employed these concepts extensively in the materials for CSE 335, an intermediate level programming course that moves students from the small, single page programs of the first courses to larger systems development. The course was often a stumbling block for students, as it required them to transfer a larger amount of classroom knowledge to practice. By employing a constructionism approach to that course, I have been able to reduced the drop and fail rate to small numbers. Students build a FarmVille-like city simulation game and a keyframe-based 2D animation program. Students also create a 30 second animation, so they actually have to use the program they have written.

In the mobile application development course, the tangible item can be as simple as a screen layout created in class or as large as the working apps produced in the step assignments and projects. Students are aware from the first day of class that they will be building working, complete applications (two of the step assignment projects can be installed as working apps from the Google Play Store, for example: Sparty Puzzle and The Mad Hatter). As concepts are covered in lecture, they immediately are able to apply them, either through the written ToeTipper assignment or using the CSE 476 Course Application.

The mobile application development course presented me with a new capability: every student in the class has an Android device and can utilize them during the class period. Though many students have laptop computers, their availability cannot yet be assumed, installing and updating applications on conventional laptops is problematic, and there is a wide range of platforms. Therefore applications are usually limited to pure web apps. As a hands-on tool running on the actual devices students are learning to develop for, the app provides both the chance to do exercises with immediate indication of results and experiment with how the Android operating system works so they can see how their designs will interact with the OS. Most existing courses in mobile application development either use a conventional classroom lecture format with outside work or are taught in dedicated labs (Fenwick, Kurtz, and Hollingsworth, 2011). Using the app in the classroom allows the hands-on advantages of a computer lab without the expense and seat limits.

### III. Points of Interest and Innovation

*(Please discuss course highlights and including URLs and/or screen shots of key components of the course or technology-enhanced learning innovation you want to bring to the attention of the judges. Possible outstanding aspects of the course might include student interaction, rich media content elements, interactive learning objects, assessment, effective incorporation of polling and surveys, facilitated teamwork, peer review, portfolio creation, etc.)*

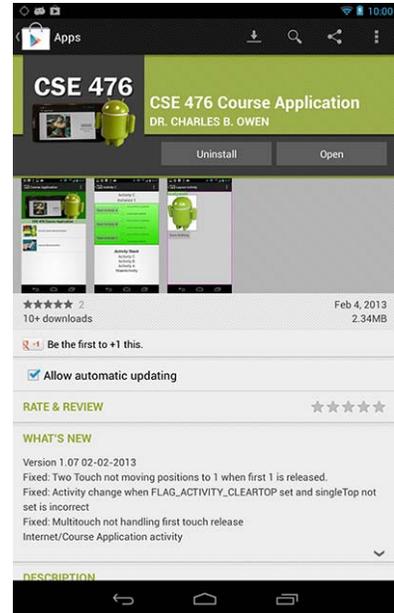
Typically, mobile application development courses are taught with a ratio of one device for every two to three students and the department invested in 17 Google Nexus 7 tablet computers to support the course assuming that model. However, the dropping prices of tablets, the rapidly growing popularity of Android phones, and the ability to supplement those who did not have devices with department-owned tablets has led to every student in the course having a device,

which is a significant pedagogical advantage. Students can be expected to have a device during the class period that they can use for the course<sup>2</sup>.

## The CSE 476 Course Application

During class, students use the CSE 476 Course Application, an application available to them for free from Google Play. The Play Store page for the application is shown to the right. Distribution via Google Play allows me to update the application as it evolves over the course of the semester. Updates are automatically pushed to student devices within 24 hours. The application is under continuous development during the semester. At this time there are nine different activities. Several additional activities are under development and will be available this semester.

The goal of the app is to provide demonstrations of functionality that complement lecture presentations and to provide tools for exercising ideas in class. Many of the concepts in mobile application development are new and unique. Demonstrations help to make those concepts real and exercises help students understand how to apply these concepts in their own programs.



The Android operating system assumes most user applications are used full-screen or in a fixed screen area. Each page of an application is called an *Activity* and the flow of the user interface is through the activities in the program. Developers must learn to manage the activity stack, an internal record of what activities are currently active. The process of invoking new activities includes flags that control the activity stack. The Activity Stack activity allows students to experiment with those flags and see how activities are managed. The program displays the current activity stack, something that is normally hidden from developers, and indicates what instance of an activity is current displayed.

A common source of frustration for new Android developers is not understanding that invoking an activity may create a new instance rather than returning them to a previous instance. This activity helps to make it clear how and why this occurs.

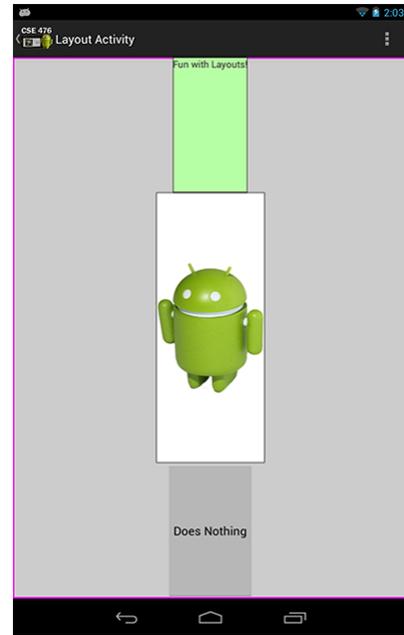
One of the most difficult elements of application design for mobile devices is effective screen layouts. Mobile devices have a wide range of screen sizes, aspect ratios, and applications typically must work in both landscape and portrait modes. To allow for flexible placement of user interface components, the Android operating system has a complex layout system including

<sup>2</sup> Given the declining prices of devices such as the Nexus 7 and upcoming offerings from Samsung, future offerings of the course will require students to provide their own device

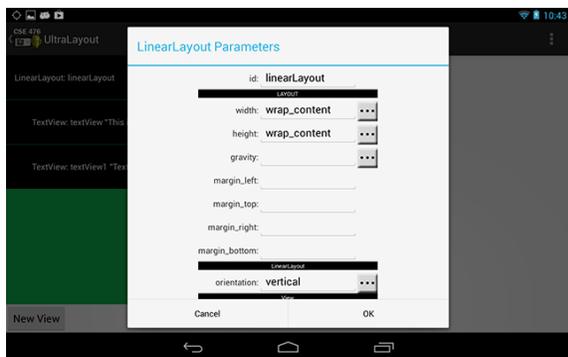
several ways to group components, device independent layout units, and a box model for the distribution of screen real estate.

The app has two different layout activities. The first activity presents students with a simple layout of three component and a *LinearLayout* grouping, the most common basic grouping. Dialog boxes allow students access to many of the layout parameters including size specifications, margins, padding, weights, and gravity. This is the first layout experience for most students. As they adjust parameters, the layout changes on the screen so they can see what the parameters mean. These concepts are then transferable to the interface in the Android Development Tools, where they can create layouts on their own.

Layout is a complex, confusing, and often frustrating task. The app provides two different activities that allow students to experiment with creating layouts. The basic layout activity was the first one created and has the advantage that options are limited. Students are not creating a layout, but rather manipulating the parameters for a simple fixed layout. The focus of experimentation can be entirely on the parameters and their unique meanings.

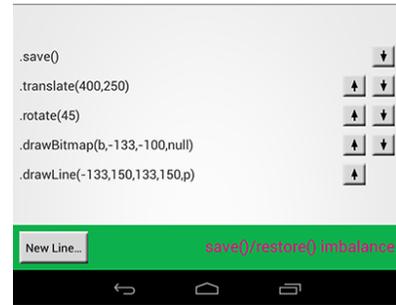


However, students also need to understand how to structure layouts. Android layouts are built as a tree consisting of grouping components such as *LinearLayout* and *RelativeLayout* and widgets that represent views and controls on the screen such as buttons, text, list boxes, or images. The *UltraLayout* activity allows students to create their own layout structure. In this illustration, the actual layout tree is on the left on the display and a layout area on the right. (the layout varies depending on the size, resolution, and orientation of the device).



Students can add new views including layout components to the tree on the left. Touches in the tree bring up a dialog box where they can edit the parameters in the same format as in the development platform. Changes are immediately evident in the display to the right. *UltraLayout* is much more rich and complex than the basic layout activity and an entire class period is planned for exercises using this activity.

The internal drawing component in the Android platform is the *Canvas* class. Canvas allows for drawing of a variety of primitives including pixels, lines, shapes, and images. In keeping with the device independent theme of mobile devices, the drawing tools are used in an environment where scaling, rotation, and translation are common and usual. The Canvas activity allows students to do experiments using the Canvas class, creating what is analogous to the code they will write in their application. They create lines of code that include the transformation tools, translation, rotation, scale, and skew, and code to draw bitmaps, lines, and rectangles.



Canvas uses a stack-based system for managing the effects of transformations on the drawing tools. The program allows students to use those tools and indicates when they have an imbalance in the save/restore sequence, as shown to the right.

As students begin to manipulate images and primitives in programs, they must learn to use the transformation components in the Canvas to construct complex composite operations. For example, the process of rotating an image around its center requires a translation of the center to the origin, the desired rotation, and a translation back to the original center point. This sequence of manipulations reduces mathematically to a single operation. These manipulations and their sequencing are not obvious to students and they must learn them, preferably through experience.

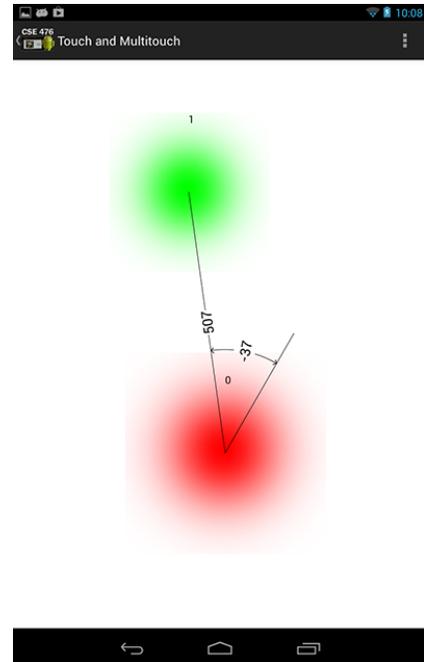


The Animation activity illustrates several basic transformation operations and how they are applied to animate primitives on the screen. This activity is used in conjunction with a lecture that presents four different ways primitives can be manipulated for animation purposes. The activity shows students the manipulation result and the Canvas activity allows them to experiment with the tools that are used to create the result. They build up programs beginning with spinning lines, moving to a spinning rectangle and text, and finally showing a more complex animation of an image the program calls the Pensive Gnome.

The lines are rotated mathematically, the box using Canvas transformations and the text requires a measurement operation. The Gnome is skewed and scaled and included mostly as a challenge to students, but also to help make the program fun. Check boxes enable each animation component individually, so students address one animation concept at a time.

Mobile devices have greatly benefitted from multi-touch technologies. There are two activities in the app that illustrate multi-touch. The first illustrates the decision process for two finger assignment and is not shown here. The second activity, shown to the right, illustrates important concepts of multi-touch application design.

As students touch the screen, a colored glow shows the touch location and the assigned pointer ID. Students can see touch locations up to the limit of their particular device. The size of the glow indicates the pressure. A settings box can enable features that show the distance between the touches and the angle of a two-touch rotation. This is used in conjunction with a lecture that presents the process for determining motion and rotation from a multi-touch manipulation. (Scale is not presented; students are required to deduce that operation themselves in a step assignment).



The process of moving, rotating, and scaling is using an image than students can directly manipulate, so they know how the multi-touch manipulation is supposed to work. Rotation of an image using multi-touch requires both computation of the relevant angles and rotation of the image corner around a touch point. The equations to solve this rotation are derived in class. The demonstration includes a crosshair that represents the image corner that must be manipulated numerically to make the rotation appear to be seamlessly done by touch.

The illustration to the right shows the activity when manipulating an image. Mathematically, the process consists of adding the differential rotation angle and rotating the image corner, shown here as a crosshair, around the rotation point, in this case the first touch (shown in red). The activity helps students to understand why it is that rotating an image actually changes its drawing location on the screen, even though the image may appear to be in the same place.



A parallel step assignment allows them to implement the process themselves including derivation of the scaling equations to complete the process.

Gestures are an important part of mobile and touch-based applications. Some common gestures are the fling, a rapid finger movement that can be used to indicate a discard or toss, single and double taps, and long presses. The gesture recognition components provide indications of these operations, but the internal representations are not always directly analogous to user experience. For example there are three events for a single tap and more than six for a double tap.



The Gestures activity creates a simple gesture area and allows students to experiment with the different gestures. The activity indicates the actual events a program receives from the gestures.

Another activity, not shown here, illustrates the process for acquiring data from an internet resource, the most recent Dr. Owen Community post on Facebook. The client-server components planned for the program are currently in development as the course moves from stand-alone application development into this next phase of application development.

Several additional activities are planned. A sensor activity will illustrate the output of the accelerometer, gyroscope, and magnetic field sensors and the use and tuning of filters to make this very noisy data useable. A range of activities will be created to illustrate the use of the client-server communication tools.

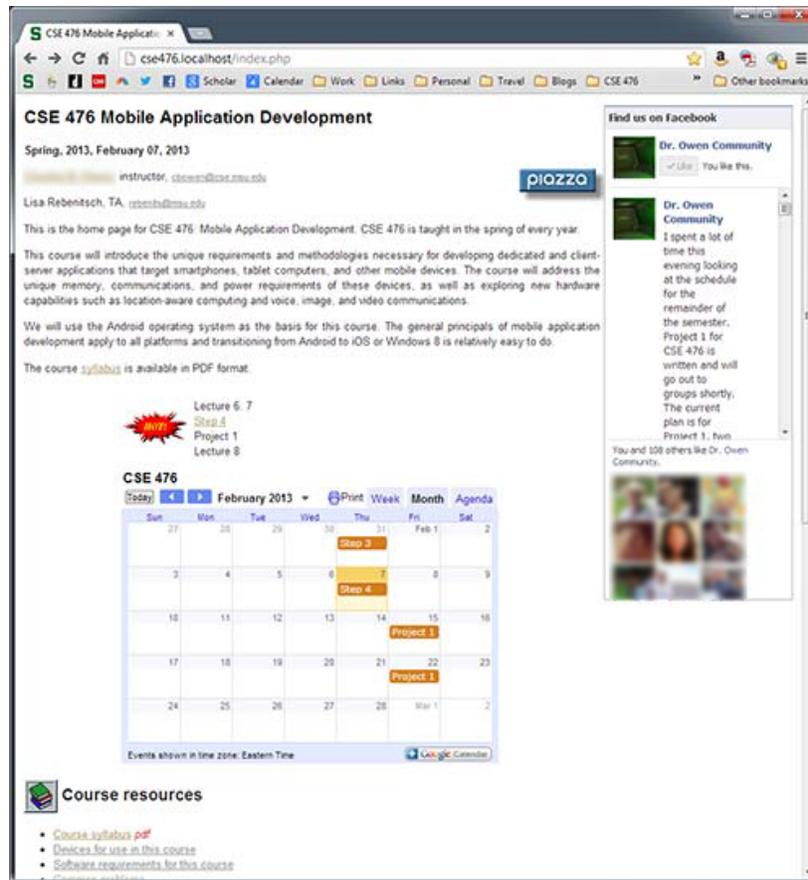
One of the elements of client-server development is database access and the use of Structured Query Language (SQL). While this coding normally executes on a server, Android devices have a fully functional MySQL database included. A planned activity will allow students to create SQL commands and execute them on the device. This will allow experiments with SQL in class.

The app is very new and the focus has initially been on producing useful tools for the classroom that operate independently in the device. However, there is a great potential for using the communications capabilities of the devices to make the classroom experience more interactive. As an example, it is planned that the app will include the ability to capture a screen and send it to a server. The goal is to allow students to submit their layout, canvas sequence, or other activity result to a server that I can then access and display on the projector. This has great potential to enhance the use of the devices in class.

The components in the course application have been carefully designed to work on a wide range of Android devices. Students in the course have devices ranging from small phones to ten-inch tablets. The activity layouts are designed to provide good functionality over this range of devices and to show students how applications can be designed to work over a range of devices. For example, the UltraLayout activity appears as two screens side by side on smaller devices, a sweep of the finger sliding between the two. On larger devices and different screen aspect ratios it divides the screen allowing the user to see both parts of the activity at the same time.

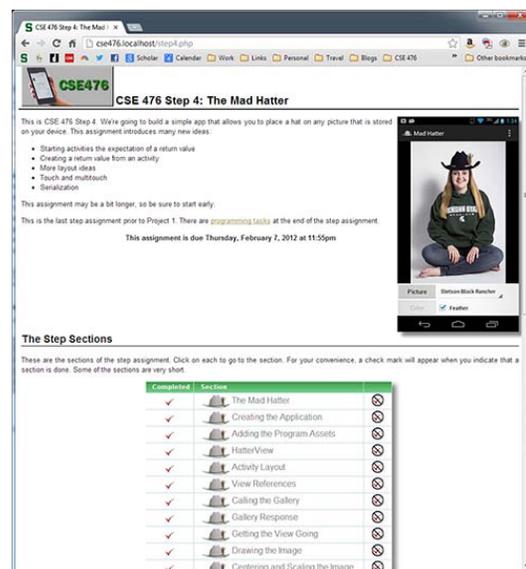
## The Course Web Site

The course web site includes all course materials including all lectures, handouts, additional material, and, of course, the Step assignments. It also includes a link for Piazza, a Facebook insert for the Dr. Owen Community, and a Google calendar that lists all pending due dates. As new content is added to the site, it is indicated in the *Hot* section.

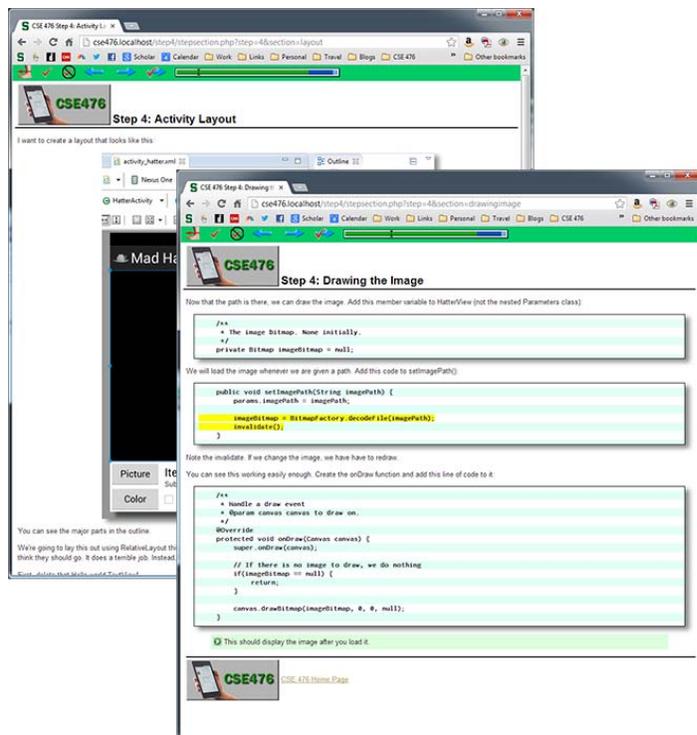


The web site is active and requires students to log in. It is backed by a database and knows the student when they do log in. This allows material to be personalized for students, which helps to prevent confusion. As an example, students are required to create a package name of the form: edu.msu.cse.studentid.MadHatter, where they replace studentid with their own id. The site presents this information with the actual user ID already inserted into the package name.

Step assignments are a key element of the course and what students most comment about as being what stimulated learning. A step assignment consists of a guided tutorial that builds up components of an application. The example to the right builds *The Mad Hatter*, a simple app that allows students to place a hat on a person's head in any picture. The step sections are intentionally very short, building one concept or part of the program at a time. The system keeps track of completed sections and includes a progress bar on the section pages and check marks in the main page to indicated completed sections or tasks.



Step sections are very media rich, including screen captures of the development tools, extensive code samples and highlighting, and pull-down sections that can provide additional information or clarifications when necessary.



The site also includes some resources that are only available to course staff. New content can be placed and reviewed by staff prior to distribution to students. Also, staff can see the current progress of all students during the step assignments.

### Additional Technology Enhancements

I have moved all course questions, answers, and announcements to Piazza and there is a link to the Piazza page on the course web site. Piazza is a major advance over the community features of Angel and is well suited to Computer Science courses. I have also made use of Skype as a tool for communicating with students in some courses.

#### IV. Accessibility

*(It is not a requirement that winning entries be accessible to learners with visual, auditory, mobility, and cognitive disabilities. However, if your course content or technology-enhanced learning innovation is accessible, or if it incorporates an innovative approach to accessibility, please describe.)*

While the course technology components have not been specifically designed or tested for accessibility, every effort has been made to accommodate disabilities. In particular, the Course Application includes the content descriptions and annotations necessary to support the Android accessibility mode, though the implementation of that mode is still evolving at this time.

#### V. Evidence of Effectiveness with Students

*(Please include evidence such as comparative test scores, SIRs results, short student letters of support, your own observations of project or group performance, etc.)*

#### Student Testimonials

As this is the first offering of this course, there are no SIRs results, yet. Some students were asked to comment on the effectiveness of the technological enhancements in this course and the Course Application in particular.

The CSE 476 course app allows students to play with concepts to see what happens when they do X. Experimentation is how we learn; we need to try things for ourselves. Reading textbooks, and listening to lectures can give students ideas, but how can they really understand it if they are not doing it or using it?

My personal favorite is the Touch and Multi-touch example as it lets me see the program keep track of touches and then how that data can be used to manipulate a garden Gnome. The concept is that you know where the user is touching the screen and where they move their touches on the screen. Well so what? How can I use that? Why should I care? The app answers that question I can zoom in on my gnome, rotate him and shrink him back down again.

The class application for CSE 476 is an irreplaceable asset, especially for particularly visually oriented students. It's the difference between a professor explaining what he/she assumes you are thinking and being able to explore your own thought process. In a way it's our textbook, but at a higher level of interaction. The app provides straight-forward and effective methods of implementing techniques that are otherwise foreign to students. For some topics, it's matter of messing around with parameters to understand their effects, and for others it's a window into an entirely unfamiliar process. Overall, the app is a wonderful and innovative addition to the learning process.

As a student in Dr Owen's android application development course, I feel that the app he developed specifically for us is invaluable. I am a visual learner so having the ability to test out the concepts discussed in lecture as he is going over them is very useful. There is an activity for every main concept addressed so students can test if they understand and ask clarifying questions before the material becomes more complicated. Based on the value I place on the course app I look forward to seeing it develop even further as the course progresses.

### **Other Evidence of Effectiveness**

I have been developing many of the technological enhancements in the course, particularly in relation to the course web site, for several years. The last two years have seen a transition from traditional web-based presentation of pages of content to a more interactive Web 2.0 experience including paced content, check marks and progress indications that allow students to gauge their experience, status pages that allow me to track class progress, and interactive components that present additional information and clarification on request.

Some anonymous SIRs comments from students in other courses using the same web code base:

*"I felt the step assignments were perfectly done, I learned a lot from them!"*

*"Step assignments were the bomb!"*

A survey was conducted in Fall, 2011, the first year the step assignments were transitioned to the page-based format. Students were asked if breaking the assignments into smaller tasks improved the assignment on a scale of 1 to 5, with 1 being Agree Completely and 5 being Disagree Completely, the survey mean was 1.1 with no vote less than 2. Students also agreed strongly (1.2) that check offs and a progress bar are useful elements of the assignment. An observation by course staff seems to indicate that students start the assignments earlier when they could see the number of tasks involved. The survey result does not directly support that hypothesis, with most students neither agreeing or disagreeing that they started earlier.

## VI. Plans for Sustainability

*(Describe future plans for your course or technology-enhanced learning innovation.)*

The mobile application development course is currently in application to become a permanent annual offering and I will continue to develop the app and web presence as I teach it. A consistent factor in Computer Science is the rapid evolution of technologies and the necessary evolution of courses that teach those technologies. This is particularly the case for cutting-edge technologies such as mobile devices. The Course Application will necessarily evolve over the coming years to keep up with technological advances. However, maintenance and updates will be a much more manageable task in the future than the large initial development this year.

The course web site is in a modular form that separates working components from content. All of the courses I teach utilize this code base and all benefit from continuing development. I have a professorial assistant working with me to further enhance my course web presence. My design course introduced a new peer-review component this academic year that appears to be very effective. I am also moving toward a flipped course structure, with more content online and more experiential activities in the classroom.

## References

James B. Fenwick, Jr., Barry L. Kurtz, and Joel Hollingsworth. 2011. Teaching mobile computing and developing software to support computer science education. In Proceedings of the 42nd ACM technical symposium on Computer science education (SIGCSE '11). ACM, New York, NY

Laura Korte, Stuart Anderson, Helen Pain, and Judith Good. 2007. Learning by game-building: a novel approach to theoretical computer science education. In Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE '07). ACM, New York, NY, USA, 53-57.

Charles B. Owen, The CSE 476 Course Application, Google Play,  
<https://play.google.com/store/apps/details?id=edu.msu.cbowen.courseapplication>.

Seymour Papert, Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc., 1980.