Enriching Engineering Curricula with a Course on Cutting-Edge Computer Technologies

Wenbing Zhao
Cleveland State University, wenbing@ieee.org

Abstract – **This paper reports the design and implementation of a special topics course on Kinect Application Development. The primary objective of the course is to enrich the Electrical and Computer Engineering curricula by teaching cutting edge computer technologies to keep our programs current and exciting to students. This course differs from traditional Electrical and Computer Engineering courses both in the content covered and in the way it was taught. This course covers both fundamental computer vision technologies that made Kinect possible as well as the rich Application Programming Interfaces provided by the Kinect Software Development Kit. Furthermore, the course consists of both traditional lecture-based instructions and active learning components with lab exercises and team-based projects. The course offers students an opportunity to practice real-world software engineering as well as solving multidisciplinary problems.**

Index Terms – Computer Technology, Kinect, Active Learning, Project-Based Course.

INTRODUCTION

In this paper we report the design and implementation of a special topics course on Kinect application development [1], and share our experiences and lessons learned in teaching this non-traditional course. The course was first offered in spring 2014 as a tech-elective for upper-division undergraduate students and graduates students in both the Computer Engineering and Electrical Engineering programs. The motivation for offering this course is three fold:

1) *Strong student interest in learning application development skills for cutting edge computer technologies,* in particular those used in mobile devices and revolutionary game controllers. Engineering students have strong curiosity regarding the technologies that make modern gadgets work and how to program them to solve real-world problems. The Microsoft Kinect sensor (or Kinect for short) is one of the latest additions to the list.
2) *Great opportunity to teach students about real-world software engineering, as well as solving multi-disciplinary problems*. This course would offer students the opportunity to develop innovative computer applications that help solve some real-world problems in areas such as healthcare and education.
3) *Enhanced breadth and depth of the Electrical and Computer Engineering curricula*. Except a course on iPhone Application Development the author created several years ago, no other course in our Department teaches students cutting edge technologies that they directly use on a daily basis. The offering of this course help our programs remain current and exciting to students.

This type of non-traditional courses on cutting edge computer technologies was pioneered by Stanford University. In winter 2009-2010, an iPhone Application Development course (as CS 193P) was introduced at Stanford with a huge student enrollment and many students had to be placed in the waiting list [2]. In fall 2010, the author created a similar course at Cleveland State University [3]. Several of our students were able to find jobs on mobile applications development soon after they took the course. Prior to offering the Kinect Application Development course, a 2-week boot-camp style short course on Kinect Application Development was incorporated in a senior design course in spring 2012 [4].

The course on Kinect Application Development is the first course dedicated on this subject nationwide to the best of the author's knowledge. Unlike the iPhone Application Development courses, which focus completely on the Application Programming Interfaces (APIs), the Kinect application development course covers more in-depth computer vision fundamentals so that students understand how the magic of motion sensing capability in Kinect was accomplished and that they learn how to develop custom Kinect extensions beyond what is available in the Kinect APIs. Furthermore, an advanced 3D development tool [5] was introduced to students so that they can develop useful applications that fully utilize the 3D data captured by Kinect.

COURSE DESIGN

Kinect is a hands-free game-controlling device that revolutionized the gaming industry. It is equipped with an RGB camera, a depth sensor, and a multi-array microphone. Unlike Wii and PlayStation, which require the players to carry a game controller physically, Kinect enables Xbox to offer a natural user interface to players so that they could use gestures and spoken commands to control the game console and play games (i.e., the player's body is the controller). While it is great to play games with Kinect, what is more exciting to many researchers and practitioners is the Kinect's potential to be used in areas far beyond its designated use, particularly in medicine, health care, young childhood education. The recent release of a Software Development Kit (SDK) for Kinect by Microsoft paved the way for widespread adoption of this revolutionary technology. So far, several startup companies have been funded to develop innovative solutions based on Kinect to facilitate more effective patient rehabilitations [6-8].

The course on Kinect Application Development aims to teach students this exciting new technology. This section discusses the course content design, and the teaching methodology used for the course.

I. Course Content Design

Unlike traditional courses that typically focus on a set of well-defined subjects with many textbooks to choose from, this course was designed to cover a wide variety of topics, including:

- Under-the-hood computer vision research that powers Kinect for motion sensing
- How to use Microsoft Visual Studio, the primary development platform for Kinect applications.
- C# primer. This is the language of choice for developing Kinect applications. Most students do not know C# at the time of taking this course.
- The APIs offered in the Kinect SDK, including those on Kinect color image stream, Kinect depth image stream, Kinect skeleton tracking, Kinect Microsoft array, and speech recognition.
- Unity3D, a popular tool for developing 3-dimensional game applications [5]. This tool is essential to fully utilize the motion data captured by Kinect.
- Gesture recognition. Several gesture recognition methods are introduced to students. This subject forms the foundation for developing powerful gesture-based applications that use the natural user interfaces instead of traditional input from mouse and keyboard.
- Computer vision programming with OpenCV. This subject empowers students to extend the current Kinect APIs for features that are not yet supported by Kinect SDK.

Several books (on Kinect programing, C# language, OpenCV programming, and Unity3D development) were used in this course, in addition to several online resources. Details of the resources can be found on the course Web page [9]. The course content design was also influenced by the Kinect-based research the author has been conducting in the past two years [10-12].

II. Teaching Methodology

The nature of the course calls for a hybrid teaching methodology that involves traditional lectures, mini lab sessions, and team-based projects.

Except for the presentation sessions, a guest seminar on entrepreneurship, and a project progress assessment workshop, every session (1 hour and 50 minutes) involves a short lecture (about 50 minutes) and a companion mini lab exercises. Each lab exercise is centered on learning to build a simple application using the APIs covered in the lecture. To accommodate the diverse background of the students who take the course (students were in the Bachelor of Computer Engineering, Bachelor of Electrical Engineering, Master of Science in Electrical Engineering, and Master of Science in Software Engineering programs), two sets of tasks were designed. The basic set is catered to Electrical Engineering majors who do not have extensive programming background, while the advanced set is designed to challenge Computer Engineering and Software Engineering majors.

The active learning method [13] was employed throughout this course because it fits very well with the hands-on nature of this course. In particular, the flipped classroom active learning method [14] was used in the project component as well as the advanced tasks in the lab exercises. In both cases, students are expected to learn additional APIs and tools that are not covered by the lectures. During the lab exercises, the instructor walked around the classroom to check the progress of each student and answer any questions that might be raised by students. The project progress assessment workshop was dedicated entirely to check the progress of the projects and answer students' questions. These activities facilitated the instructor to engage and to work with individual students in every session throughout the entire semester.

As an example, one of the lab exercises requires the students to build a hand-tracking Kinect application where the right hand of the user is tracked and displayed on the screen with a circle shape. The advanced tasks for this lab, once completed, would make the hand tracking application a drawing application. The tasks include:

- Show all traces of the hand movement. This task would require students to explore APIs to enable the display of hand traces.
- Add a reset button on the graphical user interface to clear traces to make a new drawing. In this task, students would have to learn the API to clean up the traces.
- Add a palette chooser so that the color of the circle (that represents the hand) can be changed. This is the most challenging task because it involves the construction of a group of graphical controls (e.g., a set of buttons) representing different colors, and a mechanism to enable the change of color without disrupting the current drawing. The easiest solution would be to add the tracking of the other hand. Alternative schemes without using the other hand are possible. For example, voice command can be added to enable and disable hand tracking so that one can use the current hand to change color using the palette chooser.

Towards the end of the semester, a local business owner was invited and gave students a seminar on career development and entrepreneurship. This was done to prepare students in the class for future development of the ideas they came up with.

COURSE OUTCOME

Due to overwhelming interest in the course from students, two sections were offered concurrently in spring 2014 with a total of 72 students enrolled (5 of them are undergraduate students and the remaining are graduate students).

Every student in the course completed 18 lab exercises (at least one application was developed in each lab exercise). Overall, students completed and demonstrated 25 projects. The following is a list of outstanding projects:

- *Smart Presenter*: A powerful, intuitive application that can be used to make PowerPoint presentations using hand gestures. This is by far the most impressive Kinect application developed in the class.
- *Presentation Kinesis*: Another presentation application that allows a user to control PowerPoint slides and to make annotations using gestures.
- *Fun Learning Games*: A set of three Kinect games designed for young children to learn shapes, spelling, and simple math using hand gestures.
- *Match Me*: Another Kinect game designed for young kids.
- *A Kinect Painting Application*: A painting application that allows a user to paint using hand gestures.
- *A Virtual Sword Combat Game*: A 3D game that allows two players to play a virtual sword contest.
- *A Virtual Boxing Game*: Another 3D game that allows two players to play a virtual boxing game.
- *A Kinect Map Application*: An application that allows a user to use Google Map via hand gestures.
- *A Family Media Player*: A media center application that allows a user to play media using hand gestures.
- *Kinect Softball Pitch Analyzer*: An application that can be used to capture the pitching motion of a baseball player and provide real-time feedback regarding the quality of the pitching.

During the semester, each student was required to submit a self-evaluation report regarding the progress of the project, as well as the individual role in the team and specific contributions to the project every two weeks. The report was adapted based on the individual evaluation rubric form [15] designed for the Engineering Projects in Community Service (EPICS) at Purdue [16]. Items evaluated include the following:

- Individual contributions to the project and future tasks planned.
- Critical thinking, i.e., whether one has good insight to the problem on hand and has a good plan to overcome any challenges, as well as on the issue of how to transfer skills learned from one setting to other settings.
- Teamwork and leadership, i.e., what role one plays in the team and how well one works with, and helps other team members.

For each item evaluated, students were asked to give themselves a letter grade from A to F.

COURSE EVALUATION

At the beginning of the semester, an informal survey was conducted regarding their previous programming experiences. Less than 20 students (from a total of 72) had substantial programming experiences prior to taking this course. At the end of the semester, 25 out of 30 teams were able to present and demonstrate a quality project. This shows that the far majority of students could meet the objective of this course.

In addition to the project-based objective assessment, a survey was designed and administered at the end of the semester. A total of 68 valid surveys were collected. The survey assesses three areas of the instructor's teaching:

(1) The utilization and success rates of various subjects in the project that have been covered in the class. For each subject, the student can choose one of the three options: attempted successfully; attempted with little success; never attempted. From the selection made by students, the successful rate (i.e., number of success/total number of attempted) and the utilization rate (i.e., number of attempted/total number of students) for each subject were derived. For several advanced subjects, a low utilization rate is considered normal. The successful rate is more concerned to the instructor. A low successful rate would mean that more time and effort are needed to cover the subject.

(2) Ratings on various aspects of the course design and implementation, including the value of teaching cutting edge technology, enhancing of programming skills, and the value of the guest seminar. The subjective ratings from students would help determine the value of this course and the value of the guest seminar.

(3) The overall rating on the quality of this course.

The survey results are summarized in Tables I and II The advanced subjects such as Unity3D and object recognition have less utilization rates, and for those who have attempted, the successful rates were less on those subjects as well. The subjective opinions from students regarding this course are rather positive. In the course evaluation survey, the item that received the worst rating is

TABLE I
UTILIZATION AND SUCCESS RATES OF THE SUBJECTS COVERED IN THE COURSE

In the project,	Attempted with success	Attempted with little success	Never attempted	**Success Rate**	**Utilization Rate**
I used Kinect's skeleton data	61	7	0	89.7%	100%
I implemented one or more gesture recognition algorithms	46	17	5	73.0%	92.6%
I used Unity3D together with Kinect	24	16	28	60%	58.8%
I used voice control	26	12	30	68.4%	55.9%
I used Kinect's depth data	41	9	18	82.0%	73.5%
I used computer vision algorithms to analyze depth data for object recognition	22	17	29	56.4%	57.4%

TABLE II
COURSE EVALUATION

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	**Average**
The opportunity of learning the latest technology in this course makes the degree program more current and relevant	51	16	1	0	0	**4.74**
I had a more engaged learning experience compared with traditional lecture-based courses I took	41	23	4	0	0	**4.54**
I significantly improved my programming skills in this course	28	28	11	1	0	**4.22**
I became more interested in further improving my programming skills	41	23	4	0	1	**4.51**
I enjoyed the guest seminar on entrepreneurship & career development	38	24	5	0	1	**4.44**
It is beneficial to be exposed to the subject of entrepreneurship & career development in this level of courses	35	31	1	0	1	**4.46**

Overall, how will you rate this course?	Excellent	Good	Fair	Poor	**Average**
	50	17	1	0	**3.72**

on whether or not the student has "significantly improved my programming skills in this course." It is likely due to the inexperience of the author in designing the survey form. The term "significant" is not measurable and hence it is up for interpretation. It is apparent that many students did not think they have improved their skills significantly. If the term "significantly" were not used, the result would likely have been compatible with other items.

LESSONS LEARNED

Although the overall evaluation of the course is quite positive, it is challenging to teach a class with drastically different preparation levels. In the first several lectures, the tutorial style was used to teach students how to build simple applications with various features. Programming code was given with a step-by-step guide to build each application and only a set of basic tasks was provided. Soon it was observed that the group of students who were well prepared could quickly finish the tasks and had nothing to do the remaining time. To accommodate this group of students, a list of challenging tasks (as the set of advanced tasks) were prepared for each session. Only guidelines were given on the challenging tasks without the actual programming code. The students were required to search the Web and to figure out by themselves on how to accomplish the challenging tasks.

Perhaps even more challenging is to inspire students that are not well prepared in the class to try their best to learn and improve their programming skills. Five teams could not finish their projects despite the fact that their proposals looked excellent, and despite the fact that every team submitted two progress reports claiming good progress prior to the final project presentation. The next time this course is offered, the project proposal due will be moved early by one to two weeks and students will be required to use a public source code control repository Web site, such as github.com, to enable the instructor to track their actual progress. This way, students could no longer report good progress while in fact not. The use of source code control also minimizes the chance for students to cheat without being detected.

In the progress reports students turned in for their projects, an interesting trend was observed regarding self-grading of their own performance. International students consistently gave themselves significantly inflated grades while domestic students' self-grading was much more conservative in light of the quality of their final projects. This reflects culture differences. Due to this reason, the project grades were entirely based on the instructor's judgment without considering the grades students assessed for themselves.

Finally, the teaching of cutting edge technologies courses is demanding in both hardware resources and the instructor's time and skills.

Demand on the hardware resources. For the Kinect Application Development course, over 30 Kinect sensors were purchased with the support of the Center for Teaching Excellence (recently renamed as Center for Faculty Development and Leadership) and the Department of Electrical and Computer Engineering at Cleveland State University. For the iPhone Application Development course the author taught in 2010, an even larger investment was made by Cleveland State University to purchase Apple computers and iPod Touch devices. Making things more challenging is that such devices (e.g., Kinect sensors and iPod Touch) become obsolete fairly quickly. Even though the Kinect sensors purchased for the teaching the course in spring 2014 are still supported by Microsoft, a new version of Kinect sensor is already on the market and new funding would need to be secured to teach the programming of the new Kinect sensor in the future.

Demand on the instructor's time and skills. The instructor of this type of courses would need to learn the APIs very quickly and accumulate substantial application development experiences prior to the offering of a course. Compounding this challenge is the ultra-fast updates of the APIs (typically at least one major update per year). For each new release, significant changes might be introduced. For example, the APIs for the new Kinect sensor are completely different from those for the previous generation Kinect. This implies that most of the lecture notes and lab exercises prepared previously would need to be revamped completely, and the instructor would have to retrain himself or herself.

CONCLUSION

This paper shares the author's experience in the design and implementation of a special topics course on Kinect application development, as well as the lessons learned in teaching the course. Overall, the teaching of the course was successful based on the quality of the students' final projects (direct assessment) and the survey results (indirect measurement). In addition to the unique content on cutting edge technology and computer vision fundamentals, the course was taught with a strong active learning component with lab sessions and team-based projects. The course offered students an opportunity to practice real-world software engineering as well as solving multidisciplinary problems. It was observed that students in the class showed significantly stronger enthusiasm in learning compared to traditional lecture-based courses. The far majority of students in the class strongly agreed (51 of out 68) that this course has made the degree program more current and relevant.

ACKNOWLEDGMENT

The author would like to sincerely thank the Center for Faculty Excellence (formerly Center for Teaching Excellence), Cleveland State University (CSU) for sponsoring my travel to the 2015 IEEE Integrated STEM Education Conference to present this paper via a Teaching Enhancement Award. This work was sponsored by several previous Teaching Enhancement Awards from CSU Center for Faculty Excellence in 2008, 2009, 2011, and 2013, respectively.

REFERENCES

[1] "Kinect for Windows." http://www.microsoft.com/en-us/kinectforwindows/. Accessed November 6, 2014.

[2] "Welcome to CS193P iPhone Application Development." http://web.stanford.edu/class/cs193p/cgi-bin/drupal/system/files/lectures/01_Introduction.pdf. Accessed: November 6, 2014.

[3] "EEC492/693/793 iPhone Application Development." http://academic.csuohio.edu/zhao_w/teaching/Old/EEC492-F10/eec492.htm. Accessed: November 6, 2014.

[4] "EEC490 Senior Design." http://academic.csuohio.edu/zhao_w/teaching/Old/EEC490-S12/eec490.htm. Accessed: November 6, 2014.

[5] "Unity." http://unity3d.com/. Accessed: November 6, 2014.

[6] "Reflection Health." http://reflexionhealth.com/. Accessed: November 6, 2014.

[7] "Jintronix." http://www.jintronix.com/. Accessed: November 6, 2014.

[8] "InfoStrat ReMotion360." http://www.infostrat.com/solutions/Remotion360/. Accessed: November 6, 2014.

[9] "Kinect Application Development." http://academic.csuohio.edu/zhao_w/teaching/Old/EEC492-S14/eec492.htm. Accessed: November 6, 2014.

[10] Zhao, W., Espy, D., Reinthal, M. A. and Feng, H. December 2014. "A Feasibility Study of Using a Single Kinect Sensor for Rehabilitation Exercises Monitoring: A Rule Based Approach," *Proceedings of the IEEE Symposium on Computational Intelligence in Healthcare and e-Health*, Orlando, Florida, USA.

[11] Zhao, W., Lun, R., Espy, D and Reinthal, M. A. December 2014. "Rule Based Realtime Motion Assessment for Rehabilitation Exercises," *Proceedings of the IEEE Symposium on Computational Intelligence in Healthcare and e-Health*, Orlando, Florida, USA.

[12] Zhao, W., Feng, H., Lun, R., Espy, D. and Reinthal, M. A. June 2014. "A Kinect-Based Rehabilitation Exercise Monitoring and Guidance System," *Proceedings of the 5th IEEE International Conference on Software Engineering and Service Science*, Beijing, China, pp. 762-765.

[13] Bonwell, C.; Eison, J. 1991. Active Learning: Creating Excitement in the Classroom. *AEHE-ERIC Higher Education Report* No. 1. Washington, D.C.: Jossey-Bass. ISBN 1-878380-08-7.

[14] Bergmann, Jonathan and Sams, Aaron. 2012. *Flip Your Classroom: Reach Every Student in Every Class Every Day*. International Society for Technology in Education.

[15] "Individual Evaluation Rubric." https://sharepoint.ecn.purdue.edu/epics/teams/Public%20Documents/EPICS_Individual_Evaluation_Rubric.docx. Accessed: November 6, 2014.

[16] Coyle, E. J., Jamieson, L. H., & Oakes, W. C. 2005. "EPICS: Engineering projects in community service." *International Journal of Engineering Education,* 21(1), pp. 139 – 150.