120th ASEE Annual Conference & Exposition
FRANKLY, WE DO GIVE A D*MN
June 23-26, 2013
ATLANTA

Paper ID #7094

Engagement Overload: Using Augmented Reality to Promote Student Interest in Computing

Dr. Jeffrey Chastine, Southern Polytechnic State University

Dr. Chastine has both academic and industry experience in mobile and interactive system development. He served as Chief Software Architect at a Manhattan-based mobile media development company developing augmented reality systems. He has also designed and implemented numerous mixed-reality systems for a variety of platforms and clients, including the British pop phenomenon, Duran Duran, and is currently developing augmented reality games for mobile platforms. Academically, he is an active researcher with several ACM and IEEE publications in virtual and augmented reality and has recently published a book chapter in the Handbook for Augmented Reality (Springer). As a graduate student in the Graphics, Visualization, and Usability (GVU) Center at the Georgia Institute of Technology, he contributed to early research in the nascent field of self-harmonizing karaoke software. He currently serves as an Associate Professor in Computer Game Design and Development, teaching courses such as Computer Graphics (OpenGL), 3D Modeling and Animation, and Production Pipeline & Asset Management. He has served in a variety of capacities academically including Interim Department Head, Associate Dean of the College of Information and Mathematical Sciences as well as the Graduate Program Director of the Masters of Archival Studies at Clayton State University.

Engagement Overload: Using Augmented Reality to Promote Student Interest in Computing

Jeff Chastine
School of Computing and Software Engineering
Southern Polytechnic State University
Marietta, GA 30060

Abstract

Educators face many challenges when incorporating engaging elements into their computing courses. One approach is to leverage emerging paradigms that draw interest not only from their novelty but their relevance to daily life. Augmented Reality (AR) is a nascent field of computing in which virtual artifacts are embedded into the physical world in real time, providing users with new ways to visualize and interact within their environment. Further, it contains many elements that have been shown to be engaging in the classroom, such as mobile devices, 3D media, graphics and image processing. Students are often curious about how such technology is developed, yet in years past, there were significant barriers to creating even simple AR systems. Technological advancements are now emerging in mobile computing that lower these barriers, requiring a re-examination of the role of AR in computing education. Though not the first undergraduate course in AR, this work discusses a pilot course in augmented reality intended to foster interest in computing as well as strengthen development skills. Course design and implementation are discussed, as well as challenges that emerged throughout the semester and how they were overcome. The enrollment of the course was purposely restricted in size, yet contained a cross-cut of the disciplines, including majors from computer science and computer game development. This work discusses how the diverse skillsets and backgrounds of a varied student population were addressed and showcases two representative student projects. This work also discusses the results of two surveys on student perceptions of a course in AR under the hypothesis that, because AR encompasses several elements shown to be engaging in the classroom, students would perceive AR as highly engaging. The first survey was administered to students who participated in the course, while the second was given to the general computing population within SPSU. The goal of the survey is to help educators make informed course decisions using the data collected.

Introduction

As a new technology emerges, students are naturally curious about how it works and how it might be applied across a variety of contexts. Frequently, however, there are significant costs associated with working with such mediums, as it generally requires proprietary hardware and software. Further, developing even the most basic of applications can require advanced technical skill, making the field not only technologically inaccessible to students, but potentially negatively influencing their perceptions about computing. History has shown that over time both of these barriers are typically lowered through decreased hardware costs and higher levels of code abstraction, enabling students to experiment with the technology and be creative in how they apply it. Early adopters also have "bragging rights" among their peers, fostering a sense of accomplishment and empowerment.

Augmented Reality (AR) enables virtual objects to be embedded into the physical world in real time, and its applications are almost limitless. In the past, this technology has been confined to specialized laboratories, primarily because the equipment necessary to support these kinds of environments was prohibitively expensive. In addition, developing meaningful AR applications required a broad, yet deep, set of computing skills. As such, initial experimentation was reserved exclusively for advanced graduate research. Similar to other paradigms, these constraints are being removed, providing educators an opportunity to introduce AR at the undergraduate level.

AR environments encompass many educational approaches that have been used to foster interest in computing, such as mobile devices, 2D and 3D computer graphics, and computer gaming. AR requires unique interaction modalities, and because much about these environments remains unknown, invites ideas that are not constrained by pre-existing metaphors. When combined, these elements create an attractive and potentially highly engaging platform for computing education. In this work, AR serves as a platform for increasing student interest in computing. Recently, the pilot of a special topics course in augmented reality was offered at the undergraduate level. The enrollment of this pilot was purposely kept small – with only five students participating - but consisted of a group of computing students with heterogeneous backgrounds. To begin, we discuss the structure of the course, the challenges that emerged, and the development approaches that were employed. We conclude with the results of two anonymous surveys on engagement; the first was given to students who participated in the class, while the second was sent to the general computing population of the university.

Background

Milgram describes mixed reality environments as a spectrum, with the physical world at one extreme and purely virtual environments at the other [1]. In purely virtual environments, most (if not all) aspects of the user experience are synthesized. However, because virtual objects are embedded within the physical world, AR lies between these two extremes. Its strong ties to the physical world naturally shift its position to the left (see Figure 1).

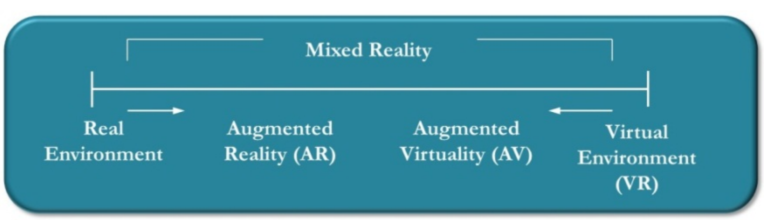

Figure 1 - Milgram's Spectrum

According to Azuma, there are three requirements for AR environments [2]. First, they must combine the physical and virtual worlds. Second, they must work in real-time (defined here at "interactive rates"); as such, pre-rendered graphics (e.g. film) do not qualify as AR. Third, virtual objects must be spatially registered into the physical world, meaning that the physical and virtual worlds must share a precise coordinate system. While the ability to render digital images and superimpose them in the view of the user has existed for some time, this third requirement presents particularly difficult challenges. It requires the system to understand the physical environment that surrounds participants, including the geometry, position and orientation of physical artifacts, as well as the viewpoint of the user. Tracking physical objects is a hinge pin for AR, yet it is often riddled with problems and, in practice, functions correctly only in restricted situations. At a minimum, once the viewpoint of the user is known, it is a straightforward task to leverage existing computer graphics techniques to render virtual objects.

Traditionally, AR system would include a head-mounted display (or HMD) to render an augmented view, similar to the helmets used in virtual reality. Using an HMD, the user views the physical world through transparent glass capable of reflecting virtual information, or alternatively, using the two small displays in helmet that display a video stream of the environment. While providing an immersive experience, they are prohibitively expensive for many universities. An alternative to HMDs is to use smart mobile devices that serve as a window through which to view the augmented world. By pointing the device at an area of interest, the video stream from the built-in camera can be sent to the display, providing the user with a mediated view of the world. This view provides no depth information, but can still serve as a foundation for an augmented experience. One approach to tracking physical objects (frequently the participant's viewpoint) is to use the video feed to identify known targets (called *fiducials*) in the scene. Recent algorithmic breakthroughs have enabled this kind of tracker to be successfully ported to today's mobile devices. Once the position and orientation of the fiducial is known, the system can deduce the location and orientation of the viewpoint of the camera.

Because most mobile devices are equipped with a camera, they can therefore satisfy all three of Azuma's requirements by tracking physical artifacts while providing an augmented view in real time. As a final note, while it is true that some tracking technology leverages accelerometers and gyroscopes, GPS is, in most cases, inappropriate for determining the position of the device. Even under ideal conditions, the error of GPS manifests as the incorrect registration of virtual objects within the physical environment.

Augmented reality has many applications. By acquiring volumetric data from CAT scans and ultrasounds, it has been used in medical scenarios to help visualize the internal state of patients, essentially providing "X-ray vision" for medical experts [3, 4]. It has been used for scientific visualization, enabling participants to interact with simulations of very large and microscopic scale [5, 6]. It has been used as an educational tool, such as demonstrating the seasons of the Earth or superimposing historical physical structures over modern terrain [7, 8]. Other examples include navigational aide and use in live performances [9, 10]. Perhaps the most famous example is seen during televised sports events, such as the yellow “first down line” in professional football.

Though enrollments are in the process of rebounding (since 2007), the computing disciplines witnessed staggering declines in the number of entering freshmen [11]. Consequently, it is predicted that the U.S. will have a workforce shortage in stem-related fields [12]. Some have suggested that the declines occurred because computing was perceived as boring and irrelevant [13]. As such, computing educators have been investigating approaches to engage students in new ways. For example, Guzdial proposed an introductory media-centric programming course for non-majors, in which students manipulated sound, images and other media [14]. Others have argued for game-based projects, and some have integrated gaming throughout their computing curriculum [15, 16]. Alternatively, some educators have introduced hardware into the classroom. For example, Mahmoud uses mobile devices extensively through many computing courses, enabling students to work with their own, personal devices [17]. Others have attempted using robotics in their curriculum, but have experienced varying results in student success [18]. Overall, it is apparent that many educators are concerned with the current state of computing education, and are investigating new approaches of teaching, reinforcing, and importantly, changing student perceptions about computer science.

Course Structure

SPSU is not the first to offer a course on augmented reality at the undergraduate level. Hollërer has offered a course on Mixed Reality Systems since 2003 [19]. Similarly, Feiner has offered AR courses since Spring 2007, if not earlier [20]. Barba et al. conducted a course on mobile augmented reality gaming, interviewing students about group dynamics and the design process [21]. The course discussed here is similar to the others in many ways, yet focused heavily on the development of AR applications, and consequently examined a broader range of current (tracking) technologies. Because of the heterogeneous background of the group, the technology was presented using two primary levels of abstraction, providing students with an option for which development path they used. However, the primary goal was to use AR gaming as a tool to engage students. Thus, producing mobile AR games was a means to an end.

Class was held once a week for 16 weeks (i.e. one semester), and was originally designed to include an extensive amount of programming to foster interest in development. The first three weeks of the course were lecture-based to ensure that students understood the fundamentals of AR development, and were later accompanied by required readings, including survey papers of the field as well as case studies. Throughout the semester, students were sent links to videos of recent commercial releases and asked to give their critique of whether or not AR was used “appropriately” (e.g. linking the physical and virtual worlds or adopting established interaction techniques). In addition to two small projects, the course required the delivery of a semester-long project. Because of the small class size, students were expected to work individually, or at most, to have one project partner.

A guiding constraint for the course was in keeping costs under control. Unlike larger research universities, SPSU does not have ready access to expensive trackers or to proprietary software that had not already been made available to the public. However, it was possible to leverage existing technology. First, though a few students chose to use their personal mobile devices, a limited number of Android and iOS tablets were made available for checkout (which were shared with another course). Second, students had access to a gaming lab that had many media-related software packages available, including the full academic licenses to Unity, enabling them to build and deploy to mobile devices. Game engines like Unity provide extensive support for common game-related elements, such as graphics rendering, physics, 3D model import, animation, and the behavioral scripting of game objects. Further, the AR plug-ins leverage existing metaphors that game developers are familiar with. However, for institutions truly on a minimal budget, Unity is not required. Therefore, at little to no cost to the students, tools were available to develop prototypes of mobile AR games.

The student body was composed of a mix of computer science and computer game design and development, and each student was interviewed at the beginning of the semester regarding his or her development experience. Several students had prior experience in developing games and were familiar with multiple programming environments. One student claimed to be a weak programmer. Another had no background in game development, though was comfortable with programming. Because of this, it was necessary to find a way to meet students at their development level while keeping the grading “fair”.

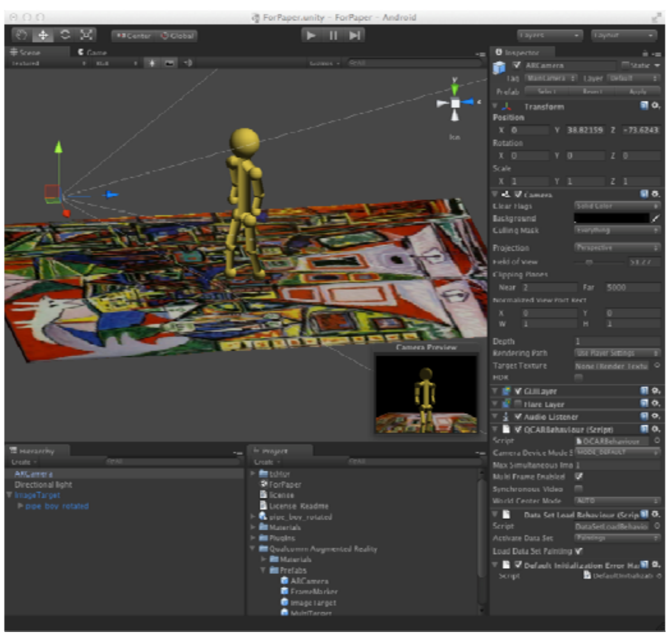

Figure 2 - The Vuforia Plugin for Unity

Early mobile AR game development restricted developers to only a handful of development options. However, the field is now mature enough to offer flexibility in the level of abstraction that developers work. This fact was leveraged to help address the disparity in student backgrounds. For example, a series of modern, visual tracking technologies were reviewed, starting with Qualcomm's Vuforia plugin for Unity [22]. A game-engine-first approach enabled students with weaker programming backgrounds explore concepts and develop AR games at a high level via scripting and a visual interface. An example of the Unity environment with Vuforia plugin can be seen in Figure 2, where the "ARCamera" is metaphorically similar to the default camera. Later, the Vuforia tracker was examined at the code level, tracing through the sample code that came with the SDK. Further, the SDK is available for both Android and iOS, the class further fragmented based on language and platform.

The Vuforia SDK tracks well but currently tracks only planar images that are known at compilation. Further, for the tracker to work correctly, at least part of the image must be visible at all times. This introduces limitations to user interaction and the spatiality of the game. To expose students to a broader range of capabilities (and to keep the course technologically agnostic), the course also focused on two other visual tracking technologies that overcome the planar limitation. The first alternative was the Dekko SDK [23]. Similar to Vuforia, this SDK download provides pre-built projects and samples of code; a plugin to Unity is also available. Dekko uses an alternative tracking technology that is capable of recognizing non-planar physical objects (that contain a sufficient amount of detail). Further, the system is trained at runtime by aiming the device at a surface (again, of high contrast) and then panning the device. This eliminates the need for pre-loading recognizable images, and also allows for the occlusion of virtual objects by physical ones. Finally, the course looked into 13th Lab's PointCloud SDK [24]. By studying this tracker, students were exposed to a system that dynamically builds a 3D map of the physical environment. As of this writing, 13th Lab plans on releasing a Unity plugin within the next few days. Though there were clearly similarities across all the tracking technology, by providing multiple examples of trackers, we were able to have meaningful educational discussions about the affordances of each of the trackers, comparing and contrasting between them as they were introduced.

Two Example Projects

In order to provide a better understanding of the kind of projects that were developed, two student examples are briefly presented. The projects both use the Vuforia SDK (likely due to the order that the trackers were introduced as well as online support), but take different approaches. To keep the assignment of grades "fair", there was a higher gaming expectation of the Unity project. The code-only project was encouraged to be purposely minimalistic, as no gaming support (not even collision detection) was provided. Much like the "Why AR?" question asked by Barba et al., both projects were encouraged to include physical elements that would draw out unique aspects of AR [21].

The Unity-based project was based on the Legend of Zelda series (see Figure 3). Because this student did not have ready access to an artist, the 3D models were found online. Using Maya, the student was able to provide simple animations for *Link* – the main character – that include idling, walking and jumping. By tapping on the screen, a virtual ray was cast onto a table to direct the character where to go in the physical world. The final version included a "Gossip Stone", a treasure chest and an artificially intelligent companion *Navi*.

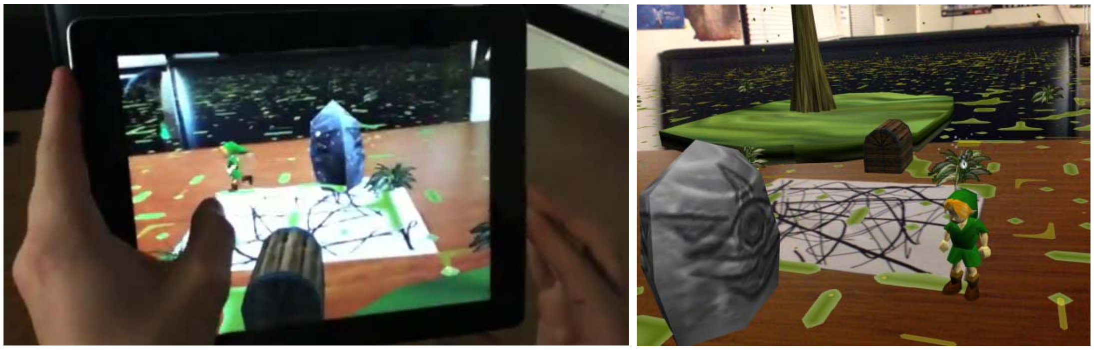

Figure 3 – A user view and close-up view of an AR project using Unity

The second project also used Vuforia, but was developed using code only (specifically, Objective-C in the XCode IDE; see Figure 4). Because of the increased complexity, the expectation of game was lowered. This student modified one of the samples that came with the SDK to create a dominos game, and was able to deploy the game to her personal device. While the game was not as robust as the previous example, a code-only approach required the student to work in an environment to which she was already accustomed.

Figure 4 – A code-only project

The Challenges

There were several logistic and technological challenges that surfaced early in the semester. First, though students had a general interest in working with iPads, not every student had access to a Mac at home, requiring them to either come into the lab or to work with an Android-based tablet. One group of students decided to develop for Android, causing a "split" in the overall discussion and requiring a significantly different development environment. The group was constantly fighting the technology, and the development of the group was hindered.

Deployment is often highly anticipated by students, but the process was plagued with problems. First, though the university has purchased an academic iOS developer license, the administrative rights to the account are coordinated by a lab manager. Thus, configuration required a tedious trial and error methodology to determine if students had been added to the account correctly, followed by the installation of certificates on their machines. The initial process became so troublesome that it was necessary to build on the instructor's machine and deploy using *TestFlight*, which enables iOS applications to be distributed on a very small scale. However, turnaround time depended on the availability of the instructor, causing additional delay. To overcome these problems (depending on the resources that are available at their institutions), educators may require students to purchase a personal developer's license. While this may seem unusual, the cost of the license is typically less than a college textbook (i.e. less than $100), is generally valid for one year, and enables students to use it beyond the period of the course.

Finally, it is important to set expectations at the beginning of the course. In the case of Augmented Reality (as well as gaming), students often envision large scale projects that are either out of scope or not technologically feasible. This can be overcome by showing videos of previous projects, or those that have been recently released commercially. By setting expectations early, students can appropriate scope their projects and will likely be less disappointed if their projects do not turn out as they planned.

Student Feedback

What did the students think? Two surveys were administered. The first survey (*Survey 1*) was given anonymously to students who participated in the course (after final grades had been assigned), inquiring about the level of engagement they experienced with the technology and topic of AR. Few students replied (3) to this survey, likely because of other obligations during the semester break. While these results are not statistically significant, the feedback still provides a glimpse of how students perceived the class. To strengthen these findings, a similar survey (*Survey 2*) was given to students majoring in computer science, software engineering or computer game design and development.

Results of Survey 1

When asked how engaging the course was compared to other computing courses, the respondents ranked this course as either more engaging or exceptionally engaging. Students were also asked which factors they felt were important, such as working with mobile devices, learning how AR applications are developed, the theory of AR, working with visual content, and developing "cutting edge" technology; they were also provided an "other" option for a free response. In general, students ranked all factors as being important. However, working with visual content ranked the highest, followed by working with mobile devices. The others received equal ranking. Students were also asked which development approach they would choose to develop AR applications (i.e. code only, a 3D engine only or either approach). While one student chose to use a 3D engine exclusively, the others responded that they would use either approach.

To capture the subjective feedback of students, they were also provided free response questions. Participants were asked to reflect on how the class could be more engaging. All responses included comments about the challenges of deploying to the device (e.g. license issues) – that there was a "technology barrier" – with one commenting that the projects should have been "set up" early. One response was that the class should meet more than once a week, and would have preferred to work on more projects. Another respondent commented that the tutorials were good, but preferred to have more of them. Overall, these responses suggest that students wanted more course structure and assignments, instead of a survey course of the technology behind AR.

The final free response question asked respondents to comment on how important it was to have a choice of development paths, as well as how prepared they were to take the course. Here are three (unedited) excerpts from the responses:

- "I am personally more in favor of visual development but being shown both choices was important so I would know the strengths/weaknesses and how to do each if needed"
- "I feel that I was almost prepared to take this course, but not quite. I honestly like both visual and traditional programming methods, and I feel that the challenge wasn't so much in one versus the other, but in the variety of methods we learned."
- "The option was awesome. Although people that cannot recognize early on when they have reached a project stopping road block when choosing the programming side are really hurt by their choice and stubbornness. I always feel prepared for any programming course because language, dev environment, and apis are super dynamic in the real world at the entry programming level. If I was unable to learn the new dev methods quickly I should go home and rethink my life as a programmer."

Results of Survey 2

To gain a better understanding of the student perceptions of AR, a second survey was given to students majoring in computer science, software engineering and computer game design and development, with a total population of 871 students. Of these candidates, 118 (~13.6%) participated in the survey, answering the first question; the remaining questions were completed by 109 respondents.

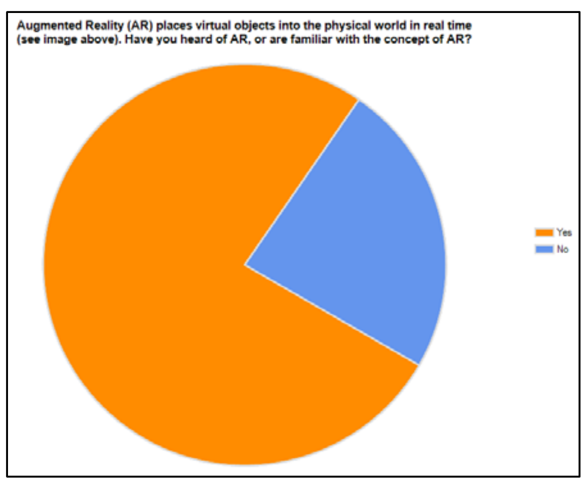

Figure 5 - Familiarity with AR

Given AR's relative novelty, the first question asked if the student was familiar with the concept of AR or had any exposure to it; 76.3% responded in the affirmative (see Figure 5). This does not come as a surprise, since many of the students were declared as computing majors.

When asked how likely they would be to take a course on augmented reality as an elective (with 1 being "Never" and 5 being "Highly Likely"), 70.6% said they would be either "Likely" or "Highly Likely" to take it. Those who were familiar with AR were more likely to take the course than those who were not (with averages of 3.942 and 3.478, respectively); the overall response was 3.844. Similarly, when asked whether or not they believed AR to be more interesting than general computing courses, 80.8% said the topic was either "More interesting" or "Much more interesting". Again, those who were familiar with the field responded more positively than those who were not (4.209 vs. 3.783, with an average of 4.119).

To better understand the importance of approaches used to engage students, participants were asked to provide a rating of several elements using a similar Likert scale. There was little to no statistical difference between those who were familiar with AR and those who were not. When asked about the importance of working with mobile devices, the average response, while still favorable, was surprisingly the lowest at 3.606. When asked about being exposed to the theoretical aspects of AR, as well as how AR applications are developed, the responses were 3.651 and 3.679, respectively. Slightly higher than this was the importance of working with visual content (e.g. 3D models), at an average of 3.771 (see Figure 6).

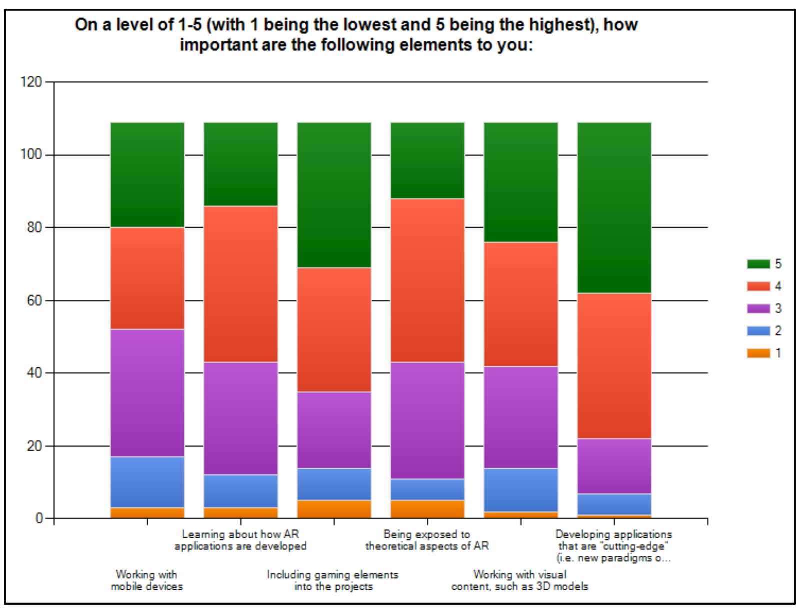

Figure 6 - Importance of course elements

The highest-rated elements were the most surprising. The results show that the inclusion of gaming elements was important to students, with an average of 3.872. This result is likely tied to the fact that there is significant population of students who major (~17%) and minor in game development, and parallels the efforts and results of others [15, 16]. Perhaps even more surprising is that students rated "Developing applications that are 'cutting edge' (i.e. new paradigms of computing)" as the most important factor, with an average rating of 4.156. There are numerous explanations for this rating, including the desire to possess an updated skillset, or an interest in developing something novel.

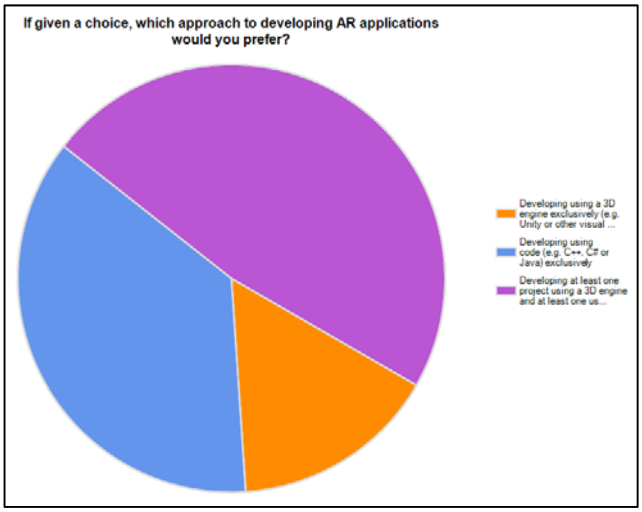

Figure 7 - Development options

The final set of questions pertained to teaching courses where students possessed diverse skill sets. Participants were asked to rate their programming skills, relative to other students, as well as their adeptness at working with 3D engines. They were also asked, if given a choice, their preferences of development – specifically 1) working exclusively in code, 2) working exclusively with a 3D engine+scripting, or 3) developing multiple applications using each approach. Almost half of the participants (47.7%) preferred the third option. This came as a surprise as well, because it is natural to believe that students will gravitate towards using their strongest skills. To better understand this, responses were categorized by skill and preference. As shown in Figure 8, there is a very strong argument that, regardless of a student's skillset, a significant number of students prefer the option of being exposed to both approaches of development.

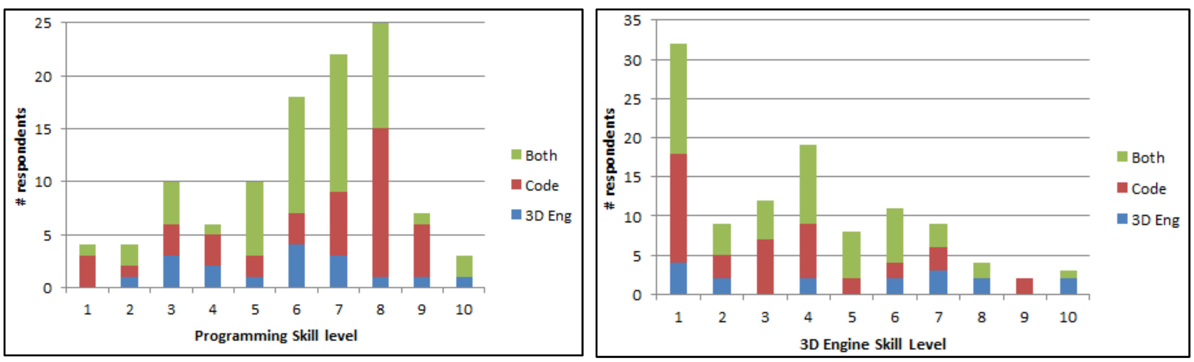

Figure 8 – Skillsets and preferences

Summary and Conclusion

In this work, we discussed a pilot course in mobile AR intended to strengthen student development skills and interest in computing. The field is now mature enough to offer students multiple paths of development to enable students with various backgrounds to develop simple AR games. Though there can be several logistical and technological challenges to such a course, initial student feedback suggested that the course was engaging and that having alternative development approaches was important. To strengthen these results, a second survey suggested that, if offered, many students would likely enroll in an AR course. Students ranked many of the elements found in AR as important to them, placing special emphasis on working with cutting-edge technology. Because of the small pool of students, it is recognized that the first survey provides only anecdotal evidence on student engagement. However, there is strong evidence from the second survey that students are becoming aware of AR, prefer developing applications using different methodologies, and perceive many elements encompassed by this field as important to them.

Acknowledgement

The author would like to thank his students for their patience throughout the semester. Special thanks go to Kevin Gross and Nicole Kosoris for their willingness to share images of their work.

References

[1] Milgram, P., Fumio K. A Taxonomy of Mixed and Reality Virtual Displays IEICE Transactions on Information Systems E77-D, 9 (September 1994), 1321-1329
[2] Azuma, R. T. A Survey of Augmented Reality. In Presence: Teleoperators and Virtual Environments 6, 4 (August 1997), 355-385.
[3] Mellor, J.P., "Enhanced Reality Visualization in a Surgical Environment, A.I. Technical Report No. 1544," 1995.
[4] Grimson, W. E. L., Ettinger, G. J., White, S. J. and Lozano-Perez, T. "An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality," *IEEE Transactions on Medical Imaging*, vol. 15, pp. 129-140, 1996.
[5] Schmalsteig, D., Fuhrmann, A., Hessina, G., Szalavari, Z., Encarnacao, L. M., Gervautz, M., Purgathofer, W., "The Studierstube Augmented Reality Project," *Presence*, vol. II, pp. 33-54, February, 2002 2002.
[6] Chastine, J.W., Brooks, J. C., Zhu, Y., Owen, G. S., Harrison, R. W., and I.T. Weber, "AMMP-Vis: A Collaborative Virtual Environment for Molecular Modeling," Proc. of *ACM Symposium on Virtual Reality Software and Technology (VRST)*, Monterey, CA, 2005.
[7] Billinghurst, M., Duenser, A., Augmented Reality in the Classroom. IEEE Computer, Volume 45, Issue 7, pp 56-63, July 2012. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6171143
[8] B. Shelton, N Hedley. Using Augmented Reality for Teaching Earth-Sun Relationships to Undergraduate Geography Students. The First IEEE International Augmented Reality Toolkit Workshop. Darmstadt, Germany. September, 2002.
[9] Reitmayr, G., Schmalstieg, D., "Collaborative Augmented Reality for Outdoor Navigation and Information Browsing," Proc. of *Symposium Location Based Services and TeleCartography*, 2004.
[10] Pair, J., Wilson, J., Chastine, J., Gandy, M., The Duran Duran Project: The Augmented Reality Toolkit in Live Performance. The First IEEE International Workshop for Augmented Reality Toolkit. Sept. 2002, Darmstadt, Germany.
[11] Harsha, P., Undergrad Computer Science Enrollments Rise for Fourth Straight Year – CRA Taulbee Report. http://cra.org/govaffairs/blog/2012/04/undergrad-computer-science-enrollments-rise-for-fourth-straight-year-cra-taulbee-report/. April 9th, 2012.
[12] McDougall, P., U.S. Tech Worker Shortage Looms, Study Warns, http://www.informationweek.com/global-cio/outsourcing/us-tech-worker-shortage-looms-study-warn/240000853. May, 2012
[13] Beaubouef, T., and Mason, J.,"Why the high attrition rate for computer science students: some thoughts and observations," SIGCSE, Jun. 2005, pp. 103-106.
[14] Guzdial, M., A Media Computation Course for Non-Majors. In the proceedings of ITiCSE '03, Thessaloniki, Greece, 2003.
[15] Overmars, M., Teaching computer science through game design. IEEE Computer, Vol. 37, Issue 4. April 2004.
[16] Leuteneggar, S., Edgington, J., A games first approach to teaching introductory programming. Proceedings of

the 38th SIGCSE Symposium on Computer Science Education. pp 115-118. New York, N.Y., 2007
[17] Mahmoud, Q.H., Integrating mobile devices into the computer science curriculum. In the proceedings of Frontiers in Education Conference (FIE), Oct. 2008.
[18] Cliburn, D. C., Experiences with LEGO Mindstorms throughout the Undergraduate Computer Science Curriculum. 36th Annual Frontiers in Education Conference. Oct. 2006.
[19] Hollerer, T., CS 290I: Mixed and Augmented Realty. http://www.cs.ucsb.edu/~holl/CS290I/
[20] Feiner, S., COMS W4172: 3D User Interfaces and Augmented Reality. http://graphics.cs.columbia.edu/courses/csw4172/missive.html
[21] Barba, E., Xu, Y., MacIntyre, B., Tseng, T., Lessons from a Class on Handheld Augmented Reality Game Design. International Conference on the Foundation of Digital Games. Orlando, Florida. 2009
[22] Vuforia SDK. http://www.qualcomm.com/solutions/augmented-reality
[23] Dekko SDK. http://developer.dekko.co/
[24] PointCloud SDK. http://13thlab.com/