A Collaborative Undergraduate Course for Pen-based Computing using Tablet PCs

Daniel G. Aliaga, Gustavo Rodriguez-Rivera, Dongyan Xu

aliaga@cs.purdue.edu, grr@cs.purdue.edu, dxu@cs.purdue.edu

Department of Computer Science at Purdue University

Abstract

Pen-based technology provides new means for interfacing with digital computers. At Purdue University, we have been developing courses that allows computer science undergraduates to explore non-traditional human-computer interfaces, to implement complex working systems, and to give them hands-on experience in developing team projects. After the first two years, we have successfully implemented a multitude of pen-based applications, across a wide-range of areas. In this paper, we present a summary of the course organization, projects, evaluations, and future directions.

1. Problem Statement and Context

We are interested in developing novel interfaces to digital computing technology. Currently keyboard and computer mice are the dominant input devices. Pen-based technology provides a much more intuitive and natural interface to computer applications. To foment the development of novel applications and to create the appropriate mindset, we wish to expose undergraduate students to such non-traditional human-computer interaction methods early in their education. Such intuitive and interesting applications will both attract students to Computer Science and expose the public to the benefits of modern non-traditional human-computer interfaces.

A major challenge for enabling such courses during undergraduate studies includes providing foundations, in terms of software, hardware, and teaching paradigms. Such foundations enable the rapid prototyping of novel applications, permitting students to quickly focus on the application at hand rather than the low-level details. Moreover, a course must be stimulating and challenging so that it entices participation and encourages applications spanning a wide variety of areas.

2. Solution Employed

Over the past two years, we have been developing a course offered each semester at Purdue University Computer Science Department for developing novel pen-based applications. During each semester, students are individually given a Tablet PC and, optionally, some additional hardware. Under the guidance of the instructor, a general topic is chosen for the semester and the students self-organize into several 2 to 3 person teams. The fall semester focuses on networking and system applications (Rodriguez-Rivera) while the spring semester focuses on more graphically-based applications (Aliaga). The students hold public demonstrations at the end of the semester to showcase their projects. Diversity amongst the applications is strongly encouraged and the organization supports significant freedom.

Each semester lecture topics are based on proposed projects. During the beginning of the semester, lectures include a general overview of a pre-chosen general topic for the semester and students produce a project design document. The design document includes individual and team milestones and deadlines. Subsequently, lectures cover supportive material relevant to the semester projects. During additional weekly meetings each student must present their individual milestones and the team must present group milestones. Students also provide written evaluations of the milestone presentations of their peers.

The course promotes the use of good software engineering tools (e.g., source control, regular meetings, design descriptions, etc.). This enables students to work tightly together and to experience, first hand, the advantages of good team coordination. These skills are very important to work in similar, real-world efforts. We encourage students to follow rules and practices of Extreme Programming [3] such as small releases, iteration planning, pair programming, unit testing, refactoring and continuous integration. Additionally, we motivate the students to use software patterns as much as possible in their software designs [4]. Some of the early lectures of the course are used to introduce both Extreme Programming and Software Patterns so the students can take advantage of them from the very beginning.

In parallel, we have been exploring the development of applications in *portable mixed-reality* (*PMR*) using pen-based Tablet PCs. This is a course development project funded by Microsoft Research (Aliaga and Xu). PMR is a fast-growing computing concept that combines traditional areas of computer graphics, visualization, multimedia computing, and mobile computing. As opposed to traditional computer graphics, which generates only synthetic environments, PMR captures *real-world* imagery and combines it with *virtual* imagery. The resultant *mixed-reality* imagery will then be able to augment a user's real-world perception as well as his/her interaction with other users or with the environment. Our teaching platform includes a PMR middleware toolkit as well as sample student projects built on top of it. The middleware toolkit is being progressively refined and enhanced during the semesters.

Our courses have resulted, so far, in the following three categories of applications: digital-paper applications, camera-based applications, and location-aware applications. The latter two groups assume future Tablet PCs will soon have cameras and/or location-awareness.

2.1 Digital Paper Applications

Puppet

This project enables users to quickly and easily draw and animate digital characters (Figure 1). Traditional character design is time consuming, difficult, and often requires sophisticated keyboard and mouse interfaces (e.g., 3DMax, Maya, Poser, etc). Puppet provides a simple intuitive interface using a digital pen to rapidly sketch, paint, and position characters [1]. Students developed toolkits for *freehand 3D character design* and *inverse kinematics*. Characters are represented by an underlying skeleton (e.g., bone) augmented with interactively sketched meshes (e.g., muscles and skin). Key-frame animation using inverse kinematics positions the skeleton to enable creating animations in minutes.

Sketch

Often, we wish to quickly sketch architectural structures. Such ability gives users freedom to quickly explore architectural configurations and styles. This project focuses on developing simple interfaces and gesture-based controls for creating architectural structures such as houses, buildings, etc

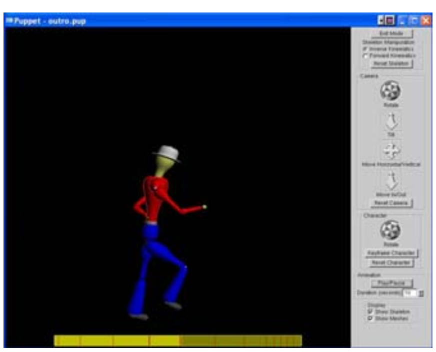

***Figure 1. Puppet**. A digital-paper application to interactively sketch and animate characters using a pen-based interface.*

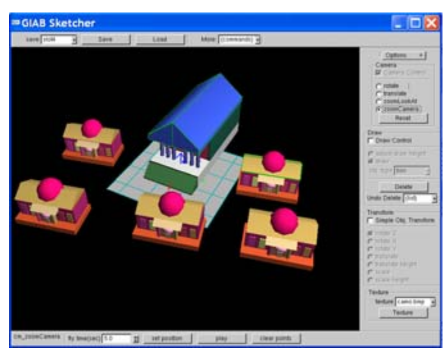

***Figure 2. Sketch**. This project enables users to quickly sketch architectural structures for rapid prototyping and visualization.*

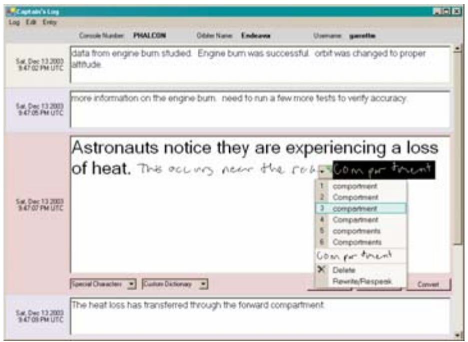

Figure 3. Captain's Log. Program allows flight controllers to keep an interactive log.

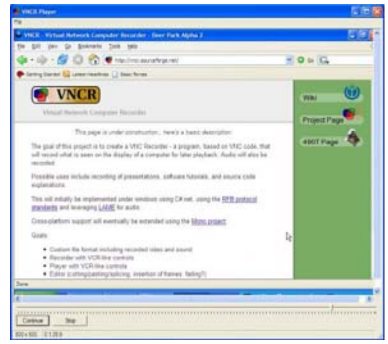

Figure 4. VNCR. This program records computer presentations for later playback.

[2], rather than on creating exact and detailed architectural models. The project developed toolkits using *gestures* for quickly creating 3D objects and using *constraints* for 3D editing on a 2D screen (e.g., movement constrained to a ground plane, explicit vertical displacement, etc). Utilities such as select, copy, paste, and animation paths are also supported using the pen-based interface.

Captain's Log

Flight Controllers at NASA currently record events on their console using a paper form. Because the handwritten log is filed or scanned at a much later date, this system does not allow instantaneous searching of records. The purpose of Captain's Log is to replace the current system with a digital pen-based interface consistent with the previous form and save the input to a database, which can be viewed and searched by an administrator. The advantage of Captain's Log is that filled-out information is available *immediately*. On the other hand, the former pen-and-paper system requires manually reading through all filed records for significant data. Captain's Log can accomplish the same task with a simple search of the database. This project was done in conjunction professor Barrett S. Caldwell.

VNCR

VNCR (VNC recorder) is based on the open-source program VNC (Virtual Network Computing) that allows displaying the screen of a PC remotely. The goal of this project is to create a VNC Recorder that will record what is seen on the display of a computer for later playback together with an audio track. Possible uses include recording of lectures and presentations, recording of teleconferences, software tutorials, and source code explanations. Once completed there will be a recorder and player program as well as a browser plug in. This program arises from the need for recording class lectures for future playing independent of the software used during the presentation. The classes recorded may include handwriting input from a Tablet PC.

2.2 Camera-Based Applications

CheckMate

The objective of this project is to design an autonomous and portable mixed-reality system that enables a Tablet PC to play chess against a human using a standard chessboard. A digital camera captures live views of a chessboard using custom camera calibration. The system automatically detects when the user moves physical chessboard pieces and then responds with a counterattack. The students of this project learned techniques in *computer vision*, *artificial intelligence*, and *computer graphics*. In particular, students developed camera calibration techniques, created an example chess-playing back-end complaint with common standards for chess engines, and implemented an interactive graphics application that displays both real and virtual imagery registered in place.

Figure 5. CheckMate*. This application uses a Tablet PC as a portable chess player.*

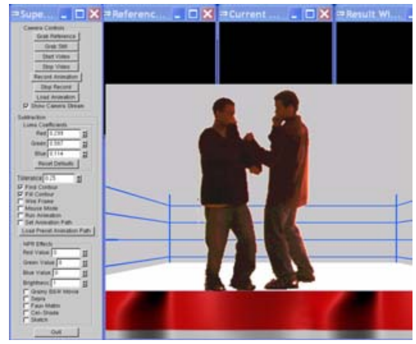

Figure 6. SuperImposer*. This project creates a stage for virtual and real-world imagery.*

SuperImposer

This project creates a live and portable mixed-reality stage. Inspired by the Playstation2 EyeToy, the same stage contains synthetic and real-world imagery for several entertainment applications. Using a digital camera and a static background scene, this program segments a moving actor object from the background. A virtual scene contains either live or pre-recorded sequences of the moving actor. A pen-based interface controls the positioning of the actor within the virtual scene. Students created this project using real-time *contour extraction*, *mixed-reality rendering*, and *non-photorealistic rendering* applied to the captured imagery.

2.3 Location-Aware Applications

ActiveMap

The goal of the ActiveMap project is to create a location-aware map program for Tablet PCs. The ActiveMap program may act as a tour-guide that displays information pertinent to the user's current location. ActiveMap shows where the user is located on a map, and even where other tablet users are located. The location information is obtained by sensing active RFID tags placed in the different rooms of a building using an RFID reader connected to the Tablet PC. The id number of the closest RFID is translated to a location and shown in a map on the tablet PC. Additional features include voice commands, a room finder, a virtual whiteboard that allows writing notes visible to other visitors of the same location, and a treasure hunt game.

Figure 7. ActiveMap. This project adds location-awareness and provides an interactive map for pen-based interfaces.

2.4 Results and Feedback

Students have demonstrated continued interest in these courses as is demonstrated by the exciting and wide-range of applications they have developed. Furthermore, we have received very strong written feedback about our pen-based computing courses. The course reviews give very high marks to both content and style of the course.

Projects of our pen-based courses have also persisted into later semesters. For instance, a revised version of the Puppet application was used during the middle-school Computer Science Summer Camp held at Purdue University. Purdue University's annual summer camp is a very large and yearly event attracting thousands of youngsters every summer. The ActiveMap application has been under continual development for two fall semesters with students each semester adding significant features to the end-application. The students are looking into using Bluetooth IDs instead of the more expensive active RFID's to make the application more cost effective.

To ensure continuity of the projects and to invite the participation of the open-source community, we have started placing the projects in SourceForge [5]. SourceForge is a web site that hosts resources for developers of open-source software. Through SourceForge, the students are able to put their projects in a source control system, browse their change history, use bug and feature track software, and also use a "wiki" to build documentation interactively.

Furthermore, specific toolkits have been developed and serve as starting points for subsequent semester projects. In particular, the teaching platform for rapidly prototyping portable mixed-reality (PMR) applications was developed during this past summer. The camera-based applications were quickly re-implemented using this framework. This work was presented as a Tablet PC poster during Microsoft Faculty Summit 2005.

3. Future Work

Going forward, we are very excited about future prospects with our pen-based technology courses. In particular, we are very interested in integrating our efforts with the EPICS (Engineering Projects in Community Service). EPICS is a unique program in which teams of undergraduates are designing, building, and deploying real systems to solve engineering-based problems for local community service and education organizations. EPICS was founded at Purdue University in Fall 1995 and has been extremely successful. We look forward to educating and training undergraduates to develop and deploy custom pen-based applications for real-world community service applications.

For Spring 2006, we are targeting the development and deployment of "learning assistant" tools. Students will develop custom pen-based applications to insist at high-school and college-level classroom learning. Unlike previous semesters, we look forward to giving students practical experiences in deploying and evaluating their own tools. These tools can also be used to help (1) measure students' grasp of material and learning progress via regular check-points and feedback meetings, (2) estimate the learning curve of computer science concepts, (3) evaluate the scalability of such Tablet PC applications to a larger user base, and (4) estimate the effectiveness of the instruction of key topics.

4. Acknowledgments and Additional Resources

We would like to thank the several funding agencies that have made this course possible. In particular, we are grateful to Intel Corporation for the seed grant providing us with the initial Tablet PC hardware. Furthermore, we would like to thank the Microsoft Research Tablet PC Curriculum Group for their subsequent funding.

References

[1] T. Igarashi, S. Matsuoka, H. Tanaka, "Teddy: A Sketching Interface for 3D Freeform Design", Proceedings of ACM SIGGRAPH, pp.409-416, 1999.

[2] R. Zeleznik, K. Herndon, J. Hughes, "Sketch: An Interface for Sketching 3D Scenes", Proceedings of SIGGRAPH 1996.

[3] Kent Beck, "Extreme Programming Explained", Addison-Wesley Professional; 1st edition (October 5, 1999)

[4] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional; 1st edition (January 15, 1995)

[5] Sourceforge.net, The world's largest development and download repository of Open Source code and applications, http://sourceforge.net.