

Using Modern Interaction Devices for HCI and Interaction Design Courses

Benjamín Hernández¹ and Adriana Alvarado²

¹Computer Sciences Department, Barcelona Supercomputing Center, Spain

²Universitat Pompeu Fabra, Barcelona, Spain

Abstract

Recent dissemination of proprietary and third-party PC drivers and SDKs of advanced console controllers and modern interaction devices has enabled new forms of 3D, tactile and multimodal user interfaces. In this paper, we present an educational methodology which allows students to use modern interaction devices (touch screen, depth sensors, gyroscopes) and programming environments in their projects. We re-design Human Computer Interaction (TI – 2004) and Introduction to Interaction Design (TC – 1015) undergraduate courses offered at the Tecnológico de Monterrey, Mexico City Campus, based on the project based learning (PBL) technique and interaction design process. The students were building their own knowledge through the development of a semester project enabling them to demonstrate, taught, and discussed with each other what they had learned. As a result, students learned up-to-date technologies and applied successfully concepts such as body gesture tracking and recognition, natural user interface design and multimodal interaction into their projects.

Categories and Subject Descriptors (according to ACM CCS): K.3.2 [Computer Graphics]: Computers and Education—Computer and Information Science Education

1. Introduction

One of the critical components of commercial software is the interface. It has been suffered important changes from text-based command line systems to highly-interactive graphical user interfaces (GUIs) to natural user interfaces (NUIs). Final users are part of a computer literate audience. They are now familiar with different interface styles found in a vast variety of consumer level devices, and they are less tolerant of difficult to use systems.

As a result, companies include interaction design teams and professionals as part of their product development groups. Understanding HCI will lead software designers to produce usable products for everyone. On the other hand, HCI and interaction design educators must offer strategies to provide fundamental knowledge of HCI practice. In addition, they should provide up-to-date learning tools that enable new graduates to take advantage of modern interaction devices to re-deploy legacy software or to develop innovative new applications.

Courses on HCI have become an integral part of com-

puter sciences curricula since the early-2000's. Since then, educators has perceived that HCI is not only about the implementation of user interfaces (UIs), web or graphic design [DTL*02, BBMR03]. HCI training requires that the student perceives HCI as a fundamental part of the software development lifecycle. They must be familiarized with task analysis, design, usability engineering and testing. We are extending these views by incorporating new approaches to HCI and interaction design in order to familiarize students with alternative UIs such as tactile, 3D and gestural UIs, modern interaction devices and their relationships to HCI concepts. In addition, students are exposed to practical experience through the project based learning (PBL) technique.

2. Course Context and Structure

Human Computer Interaction and Introduction to Interaction Design are courses offered at different campuses of Tecnológico de Monterrey system. Both are taught in the second semester of B.S. Computer Science and Technology and in the fourth semester of B.A. Animation and Digital Arts programs. Since 2012, Introduction to Interaction Design re-

placed Human Computer Interaction course, however, the program basis is maintained. Introduction to Interaction Design course has the pre-requisite of mathematics and basic programming and data structures courses.

The objectives of these courses are students will be able to apply the user-centered design process to develop interactive applications, considering knowledge of the user and his/her context; design low- and high-level prototypes; design digital media; evaluate the prototypes using empirical and analytical methods and identify collaboration skills in the development of a technology project. Similar courses and different views to achieve similar objectives can be found in [Rom13] while using interactive technologies to engage computer sciences students at the beginning of their studies are described in [APH*12].

In order to accomplish these objectives, we have opted to incorporate the Bill Verplank's approach to interaction design process [Mog06] and project based learning technique (PBL) [GS92]. Verplank suggests a four-step process. First, the designers are motivated by an error or inspired by an idea and decide what the ideal goal for the design should be. Next they find a metaphor that connects the motivation to the end goal and develop scenarios to help them create meaning. Then they work out step-by-step what tasks are and find a conceptual model that ties them all together and clarifies the modes. Finally, they decide what kind of display is needed, what the controls are, and how to arrange them [Mog06].

In addition to this, the course was organized as a PBL course, i.e. the students developed a hands-on project, having in mind projects are generators of complementary knowledge and are focused on questions or problems that "drive" students to encounter the central concepts and principles of HCI; the level of difficulty of the projects should be enough thus it can be student-driven to some significant degree and finally the central activities of the project must involve the construction of knowledge on the part of students [Tho00].

The course structure involves teacher lectures, laboratory classes and student presentations related to the advancement and discussion of the projects. Teacher lectures include theoretical concepts summarized next:

1. Introduction to human computer interaction.
 - Basic Concepts.
 - Interaction Styles.
2. The users and their context.
 - Need analysis, user preferences and limitations.
 - User profile design.
3. The interaction Design Process.
 - User centered design.
 - Design of interactive systems.
4. Prototypes and visual design.
 - Graphic Design principles.
 - Development strategies based on prototypes.

5. Evaluation and Usability testing.
 - Usability basic concepts.
 - Evaluation strategies and usability.
6. Practical cases.

A series of demos, videos and practical cases of past and recent advancements of HCI research and technology complement the original course contents. This material is usually retrieved from venues such as Siggraph's Emerging Technologies, Siggraph's Art Gallery, Ars Electronica, MIT's Media Lab demonstrations among others. In addition to the goal of making more attractive and dynamic the teacher lectures, the objective of presenting this material is to provide additional context to students, persuade and motivate them to built interactive systems beyond traditional GUIs for information systems or web pages and demonstrate examples of how metaphors can enhance or mess up interactions.

Laboratory classes cover different programming environments such as Processing [Pro13], XNA Game Studio [Mic13] and OpenFrameworks [ope13]. In this sense, through hands-on sessions students learn the generic stages of an interactive system and how these stages can be implemented in these environments. In addition, students learn the technical specifications of different interaction devices from mouse and keyboard to Microsoft Kinect, Wii mote and sensors available on mobile devices (touchscreen and gyroscopes) and how they can be used in such programming environments.

3. Project Description

Human Computer Interaction and Introduction to Interaction Design courses are offered in the second semester of B.S. Computer Science and Technology and in the fourth semester of B.A. Animation and Digital Arts programs. Each semester the course is integrated by scholars from both programs allowing the organization of interdisciplinary teams, which is a prerequisite for each project.

Every semester, we managed two groups formed by twenty to twenty five students typically integrated by 60% of male students and 40% of female students. Each project should be developed by three or four students in order to be able to record individual academic progress. However, smaller or individual projects were allowed but not recommended.

Student projects were organized in five stages. Each stage must be delivered during the semester. The course duration is fifteen weeks long, then project part 0 is reported at the second week, part 1 during the third week, part 2 is delivered at the fifth week. Students have eight weeks to work on part 3, project implementation, and another two weeks to perform system evaluations. Each project stage is described next:

Part 0 - Topic Definition

This part of the project is a warm-up activity in order to

students get to know each other. It is expected that at the end of this stage interdisciplinary teams will be formed. Then each team must define a primary role and two secondary roles for each participant in order to maintain balanced the abilities of the group. The roles are assigned according to the project needs; however, we supervise how the roles were chosen. Role assignment should stimulate the technical and soft skills development of each individual. Some examples of the roles are programming, graphical design, interaction design and management.

Deliverable consists in a definition and description of the problem, necessity or idea that their project will satisfy and a list of the team members and a description of their corresponding roles and activities. It is important to mention that, such list may change according to the knowledge and abilities the students are developing during the course.

Part 1 - Understanding the problem

In this stage the team should answer different design questions in order to get familiar with their project needs; they are all "How do you...?" questions [Mog06]:

- *How do you affect the world?*. It is related to way people act in order to provide affordances. Then interaction with the system may start.
- *How do you get feedback?*. After interaction has started, it is important to design how the system will respond to the user.
- *How do you know?*. Students must start thinking of how the user actions will map the state of the system, in order to show the user an overview of how everything works.

After these questions have been answered, the team will describe the psychological and cognitive characteristics of users that will interact with the system and a task analysis consisting on the task the user will perform.

Part 2 - Design Alternatives

This part of the project consist on designing a low fidelity prototype consisting of renders or sketches of how the user interface may look and different storyboards of how the user will interact with the system. Then, this material is evaluated empirically by other teams in order to determine if the system's task and interactions are understandable and detect possible GUI's inconsistencies.

Part 3 - Implementation

At this point of the project, students are aware of the implementation and design needs of their projects. Each team has selected the interaction device and interaction style the system will require. In addition, each team has extended the understanding of each programming environment covered in the laboratory classes in order to select the appropriate one to develop their project.

This stage consists in the implementation of a high fidelity prototype. At the end of this stage, teams have completed

60 to 70 percent of the project. According to our personal experience, this percentage of implementation is acceptable in order to perform usability tests.

Part 4 - Evaluation

In this stage, the teams will design an evaluation plan and perform usability test according to this plan. In general terms, the plan consist on a description and justification of the evaluation techniques to be performed, a list of the tasks to be evaluated and the group of users involved in the tests.

The evaluation plan also include the results of the evaluation and the implications of such results, in order to gather user feedback, detect interaction and design problems, and propose future improvements.

Part 5 - Final Presentation

Final presentation consists of two phases, the first one is a presentation in front of class. In a second phase, the presentation venue is the Annual Computer Science Student Congress (<http://movil.ccm.itesm.mx/Congreso/>) organized at Tecnológico de Monterrey, Mexico City campus. In this venue the two best projects are presented.

4. Student Projects

In this section, we describe some of the most significant projects. The topics covered by each project were art installations, platform video games, educational applications and remote virtual environments. These projects used different technologies from mobile devices to depth sensors. In particular, *Deep Sea Tour* was designed using C++, OpenGL-GLSL and Android SDK. *Paysage, Landscape, Paisaje* was designed using processing and OpenNI while *Womboard* was implemented using C# and Microsoft Kinect SDK. The *Ketchup Factory's* team used game salad to implement an iPhone game and *Space Freaks* project was developed using XNA Game Studio and Microsoft Kinect SDK.

4.1. Deep Sea Tour

In this project, a remote rendering and interaction pipeline using a client-server architecture (Fig. 1a) and an application to test the architecture were designed. On the server side, a desktop PC managing an OpenGL engine delivers virtual environments via streaming. On the client side, the stream receiver, is a mobile device, used for streaming decompression and user input sensing. User input captured at the client sensors -such as motion sensors and touch screen- are sent back to the server, which in turn, use it to update the camera position/orientation.

The application consists on an underwater virtual tour where the user travels inside a submarine. The user can interact with the virtual environment in two different modes. In exploration mode, the user takes the role of a passenger, i.e. the user can change the camera's orientation by changing

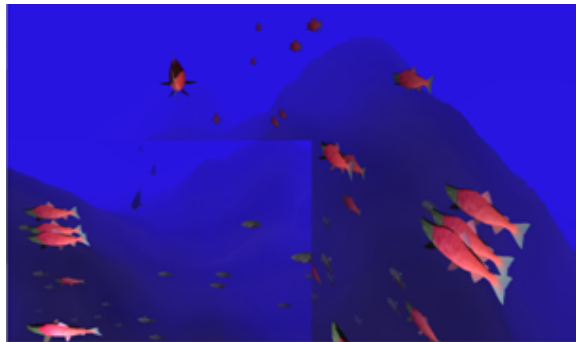
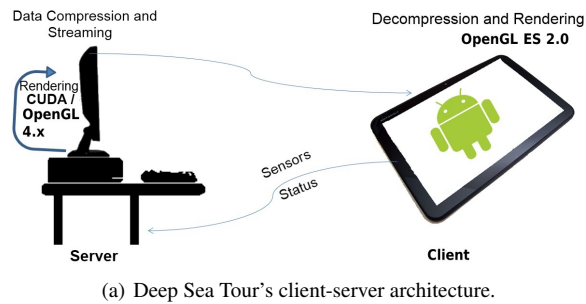


Figure 1: *Deep Sea Tour.*

tablet's orientation to watch the environment from different points of views. In this mode, the submarine's trajectory is controlled on the server side (Fig. 1b).

The second interaction mode is navigation mode. In navigation mode, the user takes the role of the submarine's captain by driving the submarine in any direction at a constant velocity using the tablet as a steering wheel. This mode is activated by positioning the tablet horizontally and touching the screen (Fig. 1c).

In exploration mode, orientation and gravity sensors were used. Data recollected from these sensors are used as a virtual axis to create the client's model view transformation matrix. In navigation mode, the gravity sensor and a tap on the

touchscreen allows us to recognize when the tablet is in a horizontal position, and the user wants to start this interaction mode.

4.2. Paysage, Landscape, Paisaje

Paysage, Landscape, Paisaje is an interactive piece where the features of the Natural User Interfaces are applied in the field of art. This project allows the spectator to perform an art work with an environment provided by the artist. Spectators are allowed to build ephemeral landscapes depending on their own motions and gestures in front of a screen, where their own performance unfolds a variety of drawings and particle animations (Fig. 2a).

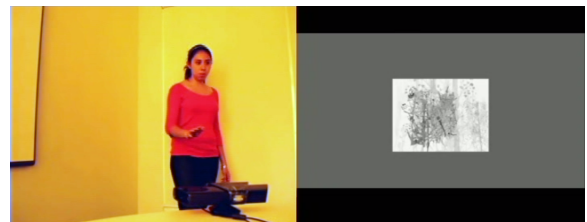


Figure 2: *Paysage, Landscape, Paisaje.*

The piece shows a blank canvas, when the user approaches to the piece, a ink drawing appears in the screen. Ink drawings represent organic forms such as tree structures, foliage, land and weather elements. The position of the ink drawing on the canvas depended on the user's body position detected by a Kinect sensor while its size depend of how far or how near is the spectator from the canvas (Fig. 2b).

Once the landscape was depicted, a second interaction stage is activated. It consists in interacting with weather elements such as rain or snow. Different parameters that control such elements (wind direction and speed) are retrieved from current weather conditions of the city where the piece is being used [Wea13].

By making interactive art works, we are looking for building a bridge between technology and the artistic practice. However in this particular case we are appealing to the art's playful sense. Therefore, the piece is giving the tools to the user to play, to build and to experiment with its own body.

4.3. WomBoard

WomBoard is a smart blackboard to digitize the information written in traditional blackboards; once this information has been digitized, instructors may save sketches, drawings, equations, graphs, or any other relevant information they draw in a blackboard. Then, instructors can reuse this data in next lectures or students can use it for self-study.

WomBoard consists of a projector that display a basic menu (Fig. 3a) on a traditional blackboard. A Kinect sensor is under the projector, and it is used to control the state of the system and to take pictures from the blackboard. WomBoard has two states, locked and unlocked; these modes are switched by leaving for 5 seconds any hand on the safe zone of the WomBoard GUI (bottom left corner).

Locked mode is used when the instructor is writing or drawing on the blackboard, in this state, the Kinect sensor only track the hands if they are on the safe zone. Unlocked state is used when the instructor is interacting with the WomBoard GUI, in this state the Kinect sensor tracks one hand for image dragging and GUI-control selection (Fig. 3b) or both to take a picture (Fig. 3c).

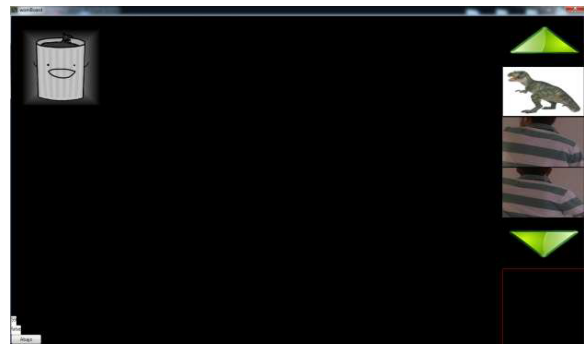
When WomBoard is in unlocked state, the user can perform the following tasks:

- Load images: recover a saved image to be used again.
- Save images: save an image to be used in the future.
- Image rotation: rotate an image.
- Copy images: duplicate an image.
- Move images: change position of an image.
- Physical recognition: recognize the user or the object to be captured.
- Delete images: delete images.
- Lock / Unlock: lock or unlock the interface.
- Take a picture: define the position with the hands an area which delimits the picture to be taken.
- Image selection: Choose the image wanted.

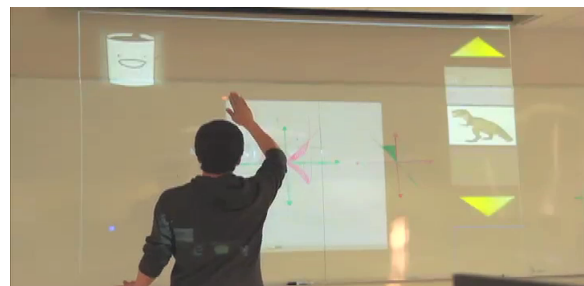
4.4. Space Freaks

Space Freaks is a game which seeks to sharp the physical skills of the player. The game has two different levels. The first one is Space Invaders-like game where the player controls a space ship with the movement of his own body. The ship will avoid falling asteroids when the player moves his body to the left or to the right directions, (Fig. 4). Body motion is tracked by a Kinect sensor.

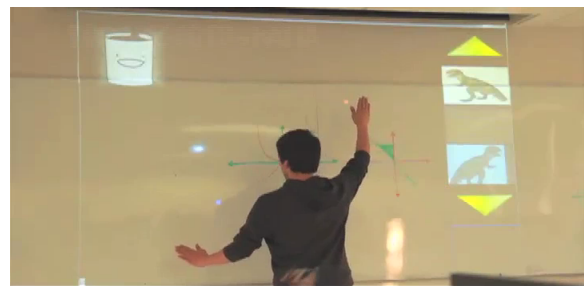
The second level is a side-scrolling runner game, the user



(a) WomBoard's GUI.



(b) With a hand, the user can drag the images.



(c) Taking a snapshot requires the use of both hands.

Figure 3: WomBoard

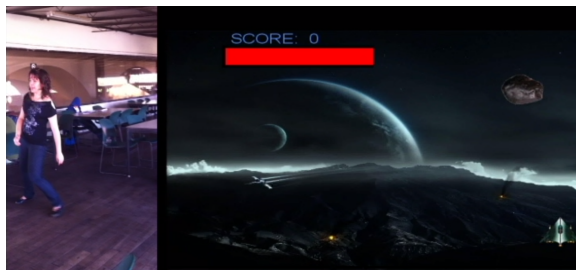
controls the main character by jumping or crunching in order to avoid road obstacles.

In both levels, power ups appear. They make invincible the player for a short amount of time. Power ups are activated by raising both arms and putting the hands together.

Both levels demand the player's attention in order to move his body in an agile to avoid collisions and complete the levels.

4.5. Ketchup Factory

Ketchup Factory is a platform game designed for iPhone or iPod, where the main character Tom Ate has to pass many different obstacles to escape from the tomato crushing ma-



(a) Level 1, Space Invaders like game.



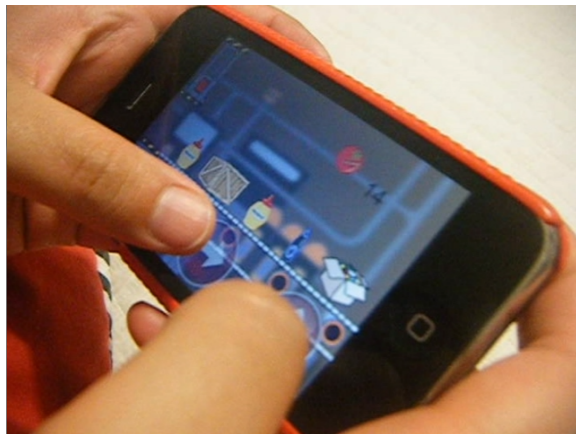
(b) Level 2, side-scrolling runner game.

Figure 4: *Space Freaks*.

chine. The aim of the game is to avoid the objects that come towards Tom Ate while keeping him on the band.

As time passes, the game goes faster challenging the user's skills. The user performs three movements by touching the screen: walking to the left or to the right and jump (Fig. 5).

After several changes in the user interface, the game was finished with good features. The controls were described as easy to use by the players, the ambient music and the rhythm of the game was perceived as entertaining and enjoyable.

**Figure 5:** *Ketchup Factory*.

5. Course Assessment

At the end of the semester students are asked to evaluate the courses they have taken during such period. The evaluation measures the quality of the course using a set of indicators such as theory-practice ratio, advising sessions outside the class, fair evaluation, intellectual challenge of the course, overall professor performance, among others. An score from 1 to 5 is given to each indicator, where 1 is the best score and 5 is the worst, then an average is calculated. We have received scores from 1.19 to 1.74 in different semesters and groups which suggest students are satisfied the way the course is taught.

6. Conclusions

After two years of offering these undergraduate courses, first as Human Computer Interaction (2010 - 2011) and then as Introduction to Interaction Design (2012), we found that students feel attracted to this kind of course organization. At the beginning of the semester, we show students some videos of previous best projects which also help to motivate them.

Previous instructors focused projects on web design and information systems. However, as it has been showed, this methodology fit well in any kind of design project. Nevertheless, constant supervision is required in order to avoid teams to lose their goals, in particular when they stuck in technical problems. In this sense instructors should possess additional technical skills.

Another interesting finding is related to the team creation. In some cases, it is difficult for students to work with others that they do not know, this tend to make difficult to form up interdisciplinary groups. Sometimes teams work well together, sometimes not. For this reason, it is mandatory to keep track of the student individual progress.

It is also important to involve the entire class in the different stages of evaluation. Evaluation of the low fidelity prototype helps them to understand the basis of each project. On the other hand, high fidelity prototype evaluation helps them to understand the technical problems, limitations, advantages of the employed technology, interaction problems and glitches. At the end of the semester, some teams are able to make last minute changes to the interaction design or UI after performing evaluations, as a result, subsequent testing make evident more precise errors such as consistency errors.

Acknowledgements

These classes could not have happened without the dedicated contributions of the students: Alvaro Herrasti, Alan Gabriel Dávalos, Manuel Enrique Zapata, Cynthia Pichardo (Deep Sea Tour). Adriana Alvarado (Paysage, Landscape, Paisaje). Manuel Hernández, Omar Eduardo Torres, Luis Ricardo Padilla, Mauricio Hernández (Womboard). Jessica Jiménez,

Marco Castro, David Alexis Balderas (Space Freaks), Rodrigo Barron, Paul Mary, Emilio Oropeza (Ketchup Factory).

We would also like to thank Nvidia for its kind donation of two Tegra 3 tablet prototypes supporting the development of Deep Sea Tour project.

References

- [APH*12] ANDERSON E. F., PETERS C. E., HALLORAN J., EVERY P., SHUTTLEWORTH J., LIAROKAPIS F., LANE R., RICHARDS M.: In at the deep end: An activity-led introduction to first year creative computing. *Computer Graphics Forum* 31, 6 (2012), 1852–1866. 2
- [BBMR03] BARNES J., BRYANT R., MCCracken D. D., REISER S.: Teaching human-computer interaction: Reports from the trenches. *SIGCSE Bull.* 35, 1 (Jan. 2003), 125–126. 1
- [DTL*02] DOUGLAS S., TREMAINE M., LEVENTHAL L., WILLS C. E., MANARIS B.: Incorporating human-computer interaction into the undergraduate computer science curriculum. *SIGCSE Bull.* 34, 1 (Feb. 2002), 211–212. 1
- [GS92] GALLAGHER S., STEPIEN W.: The effects of problem-based learning on problem solving. *Gifted Child Quarterly* 36, 4 (1992), 195–200. 2
- [Mic13] MICROSOFT: Xna game studio, Dec. 2013. URL: <http://www.microsoft.com/en-us/download/details.aspx?id=23714>. 2
- [Mog06] MOGGRIDGE B.: *Designing Interactions*. The MIT Press, 2006. 2, 3
- [ope13] OPENFRAMEWORKS: an open source c++ toolkit for creative coding., Dec. 2013. URL: <http://www.openframeworks.cc/>. 2
- [Pro13] PROCESSING: Processing development environment, Dec. 2013. URL: <http://www.processing.org>. 2
- [Rom13] ROMERO M.: Project-Based Learning of Advanced Computer Graphics and Interaction. *EG 2013 - Education Papers* (2013), 1–6. 2
- [Tho00] THOMAS W. J.: A review of research on project-based learning. In *Autodesk Foundation* (2000). 2
- [Wea13] WEATHER: library for processing., Dec. 2013. URL: <http://www.onformative.com/lab/google-weather-library-for-processing/>. 4