INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

INTERACTIVE FUNDAMENTALS FOR GRAPHIC DESIGN

Viviana CORDOVA
University of Maryland Baltimore County, United States of America

ABSTRACT

Graphic Design education is entering a new era where the influence of various fields, including interactivity, cognitive psychology, computer science, and more has broadened the research topics of Graphic Design in academia. Therefore, updating the current Graphic Design curriculum is urgently needed. My research raises current critical questions of the Graphic Design programs in the USA. The work of this paper has been done at the University of Maryland, Baltimore County in the USA. I analyzed and evaluated the teaching effectiveness of one of the basic courses, which students take in the Graphic Design Program at my University. The outcome of this course is the basis of the structure for what is to come in relation to Interactive Graphic Design. According to the results, this class provides positive outcomes for future graphic designers in the professional field. Graphic designers will be able to graduate with a competitive portfolio that involves the knowledge of the latest technology available in the field. I propose that this structure in the classroom would strengthen Graphic Design Programs and prepare their students for successful careers in the new interdisciplinary Graphic Design field.

Keywords: graphic design, data visualization, interactive design, graphics, design environments, interface design, web design, typography

Contact:
Prof. Carla Viviana Cordova Chacon
University of Maryland Baltimore County
Visual Arts
Owings Mills
21117
United States of America
cordova.viviana@gmail.com

1 INTRODUCTION

The influence of computers and creation of graphic design interfaces have led to rapid changes in the methodology, theory, and practice of a variety of interfaces, such as GUI, mobile, and virtual reality (Rogers, Sharp, and Preece, 2012). Computer scientists and software engineers were the first to create interfaces; however, now more than ever, graphic designers are actively designing interfaces thanks to the creation of the World Wide Web. Programming using a simple syntax, such as HTML and CSS, has become understandable to graphic designers, which allows for the creation of various interfaces for websites using basic graphic design principles. Nevertheless, as the devices and technologies that humans use to communicate evolve, new principles are required for graphic design. These new fundamentals need to expand and add flexibility to the tools traditionally used in the field. In addition, interdisciplinary learning within graphic design is becoming broader as new fields begin to influence the domain.

In this paper, I focus on interactive fundamentals, which have substantially influenced the graphic design field. In the past, graphic design had strict principles of form and function, including rhythm, balance, hierarchy, grid, and color (Lupton and Phillips, 2008), which were applied mostly to print. Today, as technology has progressed rapidly, graphic designers' software and hardware tools are changing and are becoming even more plentiful than before. Now, graphic designers are able to design not only a printed poster, but also an interactive graphic experience. As users target specific solutions, there is a need to find the tools necessary to create solutions that are more effective. Whereas graphic designers were once accustomed to using specific software due to limited availability, the democratization of software (Manovich, 2008), which has been a design and technological movement, has allowed designers to push themselves to be innovative visually by using new tools derived from different visual interface design systems.

A multi-semester course was evaluated in order to analyze whether the methodology applied was useful. Over time, the evaluation results became more positive, because the teaching methodology applied was viewed as being geared towards graphic designers by providing more visual semantics and definitions instead of going directly to the programming syntax components. In addition, the tutorials provided facilitate students' learning because it forces them to use their creativity to manipulate the existing code, rather than create it from scratch. Engineering Design Methods was thus a unique resource because it helped create the interface architecture and provide a preliminary conceptual design for the final prototype.

As a Professor in a Design Department, I have numerous questions related to curricula. Is there a need to teach cognitive psychology, computer science, math, entrepreneurship, and qualitative and quantitative research at the undergraduate level to graphic design students? There is a clear need for knowledge on these various environments in order to collaborate in the professional world. Owing to the current broad range of areas in the field of graphic design, these fundamentals serve to bolster the wide domain of graphic design, thereby strengthening the current field as well as the skills of future graphic designers. Graphic designers will therefore have a broader perspective of the structure behind social and technological systems. Moreover, once they are able to collaborate with scholars in other fields, they will possess the knowledge to push the boundaries through creativity.

Should this new breed of designers be called interactive graphic designers? Many papers and theories have been written on the culture of information design. However, no author has yet considered the need for designers to learn the fundamentals of lower-level programming languages, such as Objective C, C, and C++. With this in mind, graphic designers need to understand the syntax and recognize how mobile or desktop applications are created. If they cannot understand the basic structure, then how can they design? Although not every designer needs to learn these principles, those that do are encouraged to create innovative code-driven applications using the latest technologies.

At an event hosted by PARC Xerox titled "Innovation Design, Is Design Innovation?" (PARC, 2012), John Maeda, President of the Rhode Island School of Design, and Don Norman, co-founder of the Nielsen Norman Group, discussed science, technology, engineering, and mathematics (STEM). The former argued that STEM should become STEAM by adding the "A of Art", whereas the latter argued that STEM should become ART instead. This intense argument underscores the point that the current graphic design curriculum needs further development to include several new classes. Indeed, in a perfect world, graphic design would be taught to students of all fields that deal with a lot of information, at least at a basic level, in order for these fields to appreciate graphic design. Also, engineering principles will be applied to other fields. In this case programming systems is applied to the graphic design field.

2 RELATED WORK

Many students in the US are introduced to programming in advanced classes within graphic design programs. Analyzing the need for students to learn programming basics should help educators redesign courses and introduce various programming languages, or at least one programming language, during the first year of classes. Programming has become a basic tool for graphic designers— just like a pencil in a drawing class. To that end, many books have been written and programming languages created for creative people, such as Reas and Fry (2007). Moreover, HTML, CSS, and JavaScript have become available to everyone because of the need to be a part of the World Wide Web. Books such as *Programming the World Wide Web* by RW Sebesta, *Cascading Style Sheets* by Lie Bos, and *Designing Interfaces* by Jennifer Tidwell are extremely easy from which designers can learn. Free online tutorials are also available, such as at W3Schools.com, Jquery.com, and Processing.org. Indeed, learning through online tutorials has become extremely accessible. In the past, there have been a number of approaches to methodology, including a determination of which technologies should be used as graphic design tools. It has also been suggested that research should be based on learning from graduate students (Myers et al., 2008). In this paper, Dreamweaver software was found to be one of the most often used program, which is an application that involves programming. Therefore, programming for novice graphic designers should be essential and taught using design methodology as well as the most recent programming languages available.

Many schools across the US constantly update their graphic design curricula. According to the Aquent and AIGA surveys of 54,698 graphic designers, 65% perform web design, which does not include those working in mobile or other interactive graphic environments. This group of graphic designers is also involved in multidisciplinary practices (i.e., print, web, mobile, and motion). Moreover, the compensation premiums for mobile interface designers that work as consultants rather than as permanent members of staff are 36% higher. Therefore, current design curricula need to be updated because numerous design firms and employers have shared concerns about students not possessing the required knowledge on new graphic design tools. According to the survey above, 29% of designers receive additional educational and professional development support from employers (Figure 1).

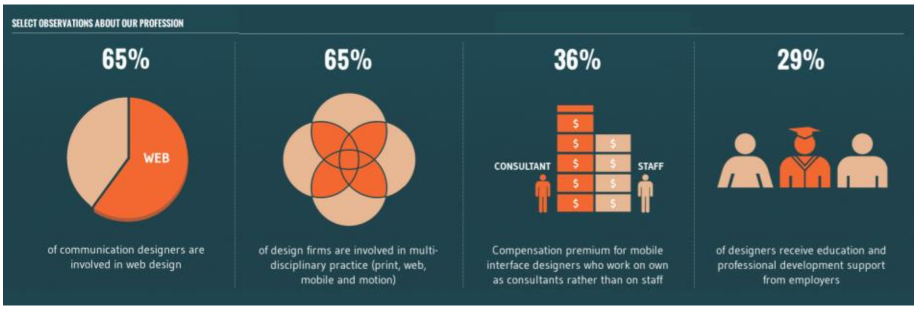


Figure 1. Graphic Design Study in the US.

Rick Grefé, the AIGA Executive Director, emphasized the following: “The positions reflecting increasing compensation appear to be those that revolve around defining or managing the integration of design into business strategy in these areas: strategists, usability experience and operations management; or those roles that deal with web, motion and interactive design” (Aquent and AIGA, 2012).

3 METHODS

Classroom evaluations and observations were conducted through qualitative research by asking the participants to complete questionnaires. I have applied new ways of teaching in some of my courses and found them to be successful. The US higher education system uses a 1–100 grading scale, where A represents 100% and F represents below 60%. The class presented in this paper, ART 336: Design and Technology II: Screen, teaches web design. In this class, the fundamentals of graphic design are applied to programming languages, such as HTML, CSS, and JavaScript. By learning the syntax of these programming languages, students can learn advanced languages in other courses as long as the terminology is geared towards graphic designers (Figure 2).

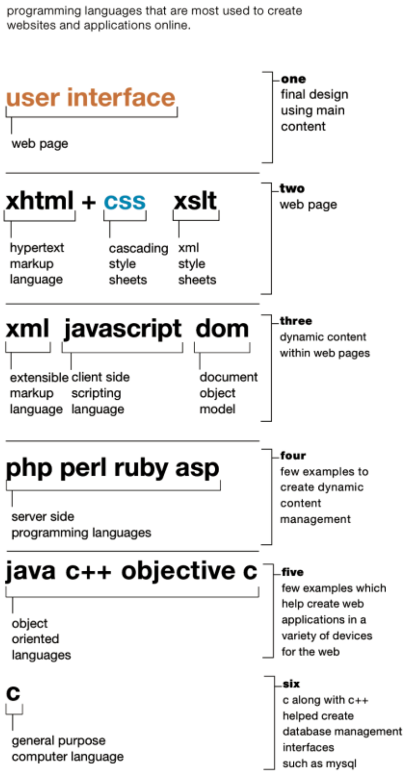

programming languages that are most used to create websites and applications online.
user interface
web page
one
final design using main content
xhtml + css xslt
hypertext markup language
cascading style sheets
xml style sheets
two
web page
xml javascript dom
extensible markup language
client side scripting language
document object model
three
dynamic content within web pages
php perl ruby asp
server side programming languages
four
few examples to create dynamic content management
java c++ objective c
object oriented languages
five
few examples which help create web applications in a variety of devices for the web
c
general purpose computer language
six
c along with c++ helped create database management interfaces such as mysql

Figure 2. Basic programming languages.

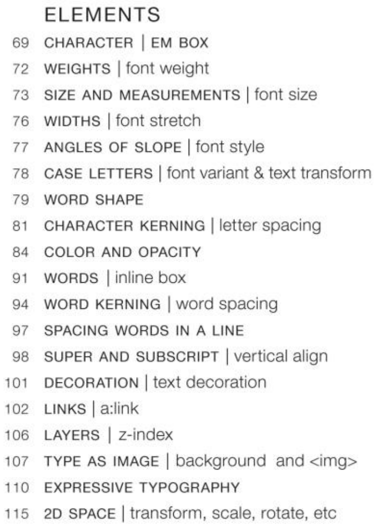
3.1 Number of Students

The graphic design studio courses at the University of Maryland, Baltimore County (UMBC) do not exceed 14 students per classroom. As mentioned above, I tested a new approach to a basic interactive graphic design class called ART 336 Design and Technology: Screen, in which students are taught the basics of web design. I have been teaching this class for over five years and recognize the need to teach graphic design fundamentals concomitantly with syntax programming languages. Therefore, I wrote a tutorial entitled *Web Typography: a Handbook for Graphic Designers,* which was funded by a fellowship of $5000 awarded by UMBC. This tutorial explains graphic design principles and designs an interactive work through HTML and CSS coding. Further, it provides free online design tutorials for students (see the examples in Figures 2, 3, and 4). I taught this class in 2009, at which point I had provided only 50% of the content due to the need for updates. By 2011, I had provided 100% of the tutorial at that time for the course and since then it has been updated until 2012 with a total of 231 pages.

3.2 New Principles for the Course Curriculum

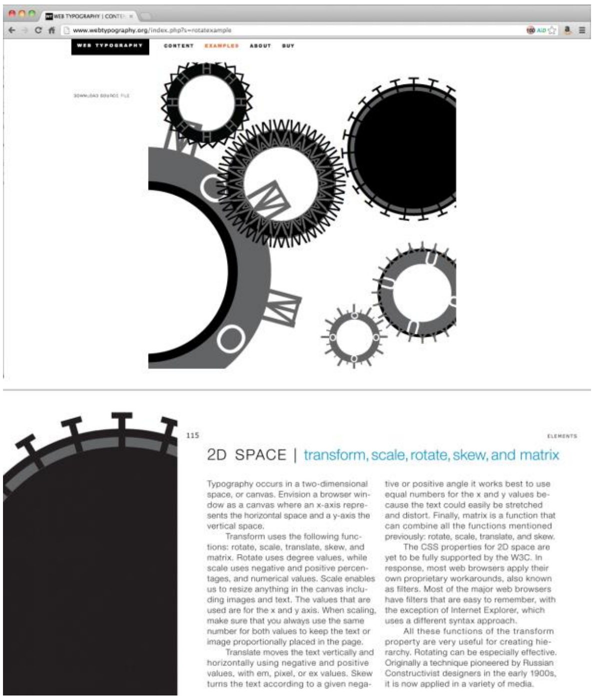
My experience of teaching this course has made me realize that new programming languages must be taught to students using graphic design terminology. Students often brainstormed ideas in the classroom, which resulted in more questions on how to solve complex design problems than on how to understand the syntax of the code. Students wanted to build systems, such as diagrams and complex websites. By using my tutorial, they were able to understand the coding by using a simple language that clearly explained graphic design terminology (Figure 3).

I am working on creating additional tutorials using graphic design terminology for students to be able understand more complex programming languages. These tutorials should allow interactive graphic designers to rapidly evolve and keep pace with other fields. In this regard, the graphic design field is embarking on an exciting future where the tools used and created by computer scientists and engineers are now shared with graphic designers (see Figures 4 and 5).



ELEMENTS

Figure 3. A section of the graphic design terminology used in the tutorial.



2D SPACE | transform, scale, rotate, skew, and matrix

Typography occurs in a two-dimensional space, or canvas. Envision a browser window as a canvas where an x-axis represents the horizontal space and a y-axis the vertical space.

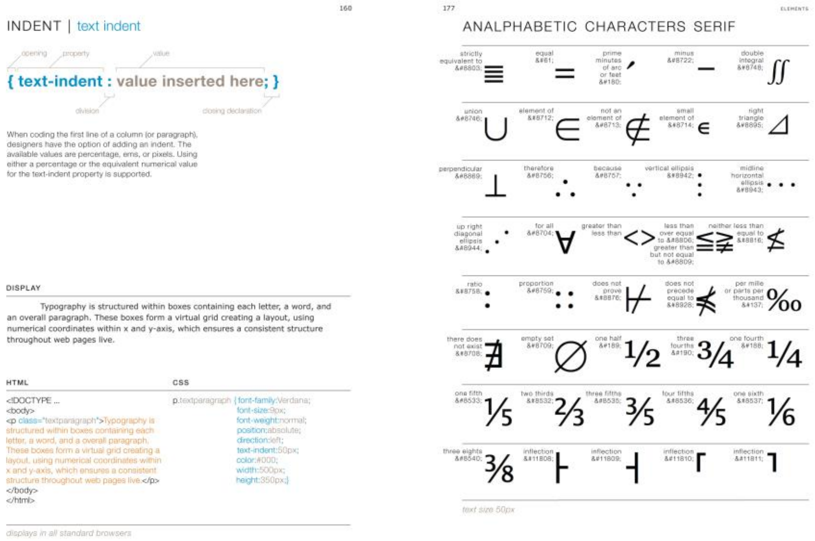
Transform uses the following functions: rotate, scale, translate, skew, and matrix. Rotate uses degree values, while scale uses negative and positive percentages, and numerical values. Scale enables us to resize anything in the canvas including images and text. The values that are used are for the x and y axis. When scaling, make sure that you always use the same number for both values to keep the text or image proportionally placed in the page.

Translate moves the text vertically and horizontally using negative and positive values, with em, pixel, or ex values. Skew turns the text according to a given negative or positive angle it works best to use equal numbers for the x and y values because the text could easily be stretched and distort. Finally, matrix is a function that can combine all the functions mentioned previously: rotate, scale, translate, and skew.

The CSS properties for 2D space are yet to be fully supported by the W3C. In response, most web browsers apply their own proprietary workarounds, also known as filters. Most of the major web browsers have filters that are easy to remember, with the exception of Internet Explorer, which uses a different syntax approach.

All these functions of the transform property are very useful for creating hierarchy. Rotating can be especially effective. Originally a technique pioneered by Russian Constructivist designers in the early 1900s, it is now applied in a variety of media.

Figure 4. Print and web version of the tutorial.



160

INDENT | text indent

When coding the first line of a column (or paragraph), designers have the option of adding an indent. The available values are percentage, ems, or pixels. Using either a percentage or the equivalent numerical value for the text-indent property is supported.

DISPLAY

Typography is structured within boxes containing each letter, a word, and an overall paragraph. These boxes form a virtual grid creating a layout, using numerical coordinates within x and y-axis, which ensures a consistent structure throughout web pages live.

HTML

```
<!DOCTYPE ...
<body>
<p class="textparagraph">Typography is
structured within boxes containing each
letter, a word, and a overall paragraph.
These boxes form a virtual grid creating a
layout, using numerical coordinates within
x and y-axis, which ensures a consistent
structure throughout web pages live.</p>
</body>
</html>
```

CSS

```
p.textparagraph {font-family:Verdana;
                 font-size:9px;
                 font-weight:normal;
                 position:absolute;
                 direction:left;
                 text-indent:50px;
                 color:#000;
                 width:500px;
                 height:350px;}
```

displays in all standard browsers

177 ELEMENTS

ANALPHABETIC CHARACTERS SERIF

strictly equivalent to &#8803; ≣	equal &#61; =	prime minutes of arc or feet &#180; ′	minus &#8722; −	double integral &#8748; ∬
union &#8746; ∪	element of &#8712; ∈	not an element of &#8713; ∉	small element of &#8714; ∊	right triangle &#8895; ⊿
perpendicular &#8869; ⊥	therefore &#8756; ∴	because &#8757; ∵	vertical ellipsis &#8942; ⋮	midline horizontal ellipsis &#8943; ⋯
up right diagonal ellipsis &#8944; ⋰	for all &#8704; ∀	greater than less than <>	less than over equal to &#8806; greater than but not equal to &#8809; ≦≩	neither less than equal to &#8816; ≰
ratio &#8758; ∶	proportion &#8759; ∷	does not prove &#8876; ⊬	does not precede equal to &#8928; ⋠	per mille or parts per thousand &#137; ‰
there does not exist &#8708; ∄	empty set &#8709; ∅	one half &#189; ½	three fourths &#190; ¾	one fourth &#188; ¼
one fifth &#8533; ⅕	two thirds &#8532; ⅔	three fifths &#8535; ⅗	four fifths &#8536; ⅘	one sixth &#8537; ⅙
three eights &#8540; ⅜	inflection &#11808; ⸠	inflection &#11809; ⸡	inflection &#11810; ⸢	inflection &#11811; ⸣

text size 50px

Figure 5. Additional examples from the tutorial.

3.3 Case Studies of UMBC

This section lists some of the projects that students worked on during the course, including a diagram containing an organic navigation where students were required to create a page that was easy to navigate. The projects used HTML5, CSS (1,2,3), and Jquery.

3.3.a Vy Vu, Interactive Diagram of Ramen Noodles Worldwide

Vy Vu was a student in the junior level at UMBC when she enrolled in ART 336: Design and Technology II: Screen. At that time, she was extremely afraid of programming code. However, by the end of my class, she was comfortable enough to be able to create what she had designed. No longer limited by a lack of programming experience, she created a diagram showing ramen noodle companies around the world. As shown in Figure 6, her design incorporates a circle in the center of the page that represents an abstract version of the world, with each different sized circle representing a country. Once the user selects a circle, a line immediately appears that connects the company with the type of noodles. In addition, a black bar at the bottom provides the location and name of the country. Typography on this page was provided by web fonts of fonts.com.

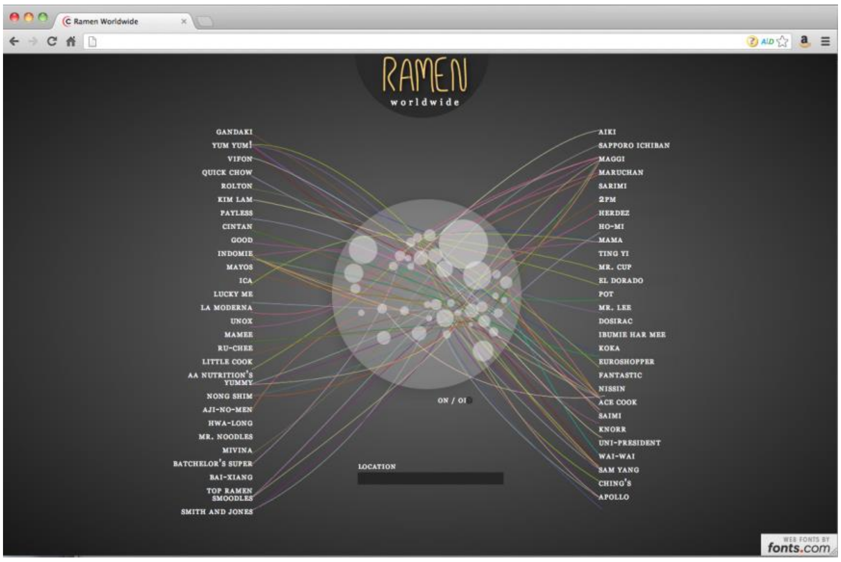


Figure 6. More examples.

3.3.b Justin Bendis, Interactive Diagram of Baltimore Ravens

Justin Bendis was also a Junior at UMBC when he enrolled in ART 336 Design and Technology II: Screen. At the time, he knew nothing about interactivity. However, in his last project, he produced a typographically created diagram of the Baltimore Ravens' player stats for the 2011 season. When the user places the cursor over the name of any player, a blue circle blinks to provide player-specific information. Typography was provided by web fonts of fonts.com (see Figure 7).

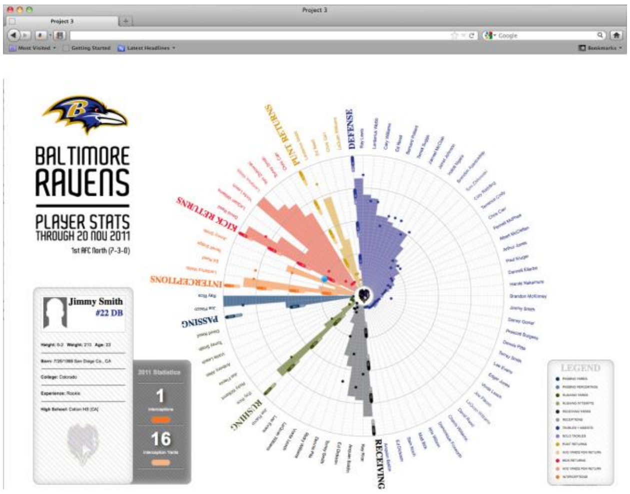


Figure 7. Justin Bendis, Baltimore Ravens.

3.3.c April Sipe, Autobiographical Study of her Monetary System, Interactive Diagram

April Sipe was a Junior at UMBC when she enrolled in ART 336 Design and Technology II: Screen. She was a graphic design student who also taught dance classes off-campus. The diagram in Figure 8 presents her money management system. The user places the cursor over an icon, and the hierarchy of arrows is then limited to the selected icon. The arrow then points to the month to indicate when the money was spent. This practical diagram also used web fonts of fonts.com.

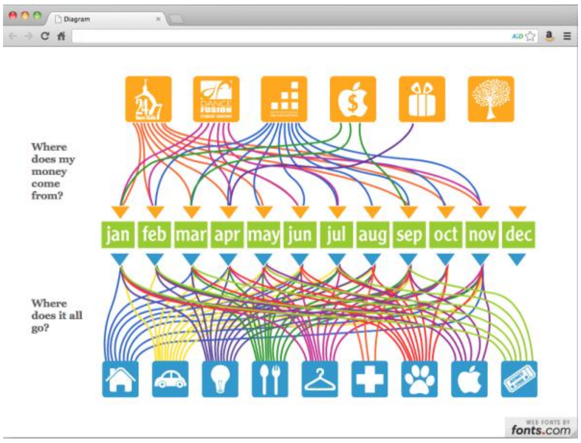


Figure 8. April Sipe, Monetary Management System.

4 PROJECT OUTCOMES

As the graphic design curriculum evolves, there is demand for more interactive projects that involve innovative, rather than traditional technologies. This need applies not only to web design classes, but also to foundation graphic design courses. According to the data retrieved from the courses I taught in 2009 and 2011 (see Tables 1–2 and Figures 9-10), the class provided positive feedback when the tutorial was used. Therefore, the same structure should be applied to this type of programming so that graphic design students can easily familiarize themselves with the content and build their own interfaces and applications.

Table 1. Characteristics of participants.

Class	Section	Enrolled	# Evaluations	# Meetings per week	Total time	Lecture time per week	Studio time/week	Critique time weekly
ART 336 Fall 2011	1	14	14	2	4	1hr:30mins	1hr:30mins	1 hr
ART 336 Fall 2011	2	14	14	2	4	1hr:30mins	1hr:30mins	1 hr
ART 336 Spring 2009	1	14	12	2	4	1hr:30mins	1hr:30mins	1 hr
ART 336 Spring 2009	2	14	12	2	4	1hr:30mins	1hr:30mins	1 hr

In terms of the course design, the course lasts for four hours per week across two days (two hours per day) due to the technical-based lectures provided. Half of this time is dedicated to lectures and the other half is studio time. Critique makes up the remaining hour. I taught these courses two years apart, but the time split was kept the same because it proved to be successful the first time it was used (Table 1).

Table 2. Student Representation.

Programming Tutorials in Class	Transfers	UMBC Students since Freshmen yr	Innovative, Interactive, Effective	Weak
ART 336 Fall 2011, Enrollment = 28	13	15	22	6
ART 336 Spring 2009, Enrollment = 28	12	16	13	5

Students in these courses represent a combination of transfers (in their incoming Junior years) and Sophomores (Table 2). According to their evaluations, 79% provided positive feedback (Innovative, Interactive, and Effective) in Fall 2011 compared to 72% in the Spring 2009 semester (Figure 9 and 10). Students that believe the class was weak were most likely not interested in the subject and preferred not to use programming as a tool (Table 2). Overall, there has been a higher positive feedback on the tutorials given to students from 2009 to 2011. The tutorial was 50% applied in 2009 and 100% by 2011, and since then it has been updated until 2012 with a total of 231 pages.

Figure 9. Enrollment and Evaluations.

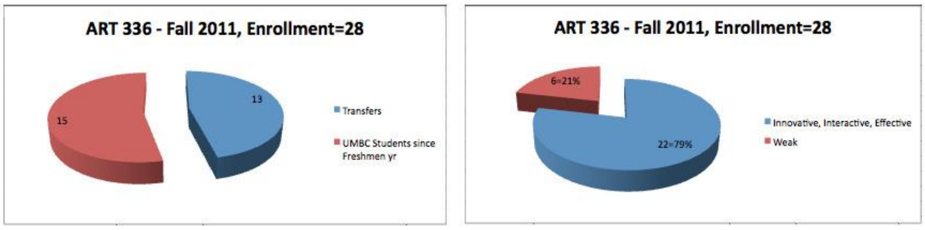
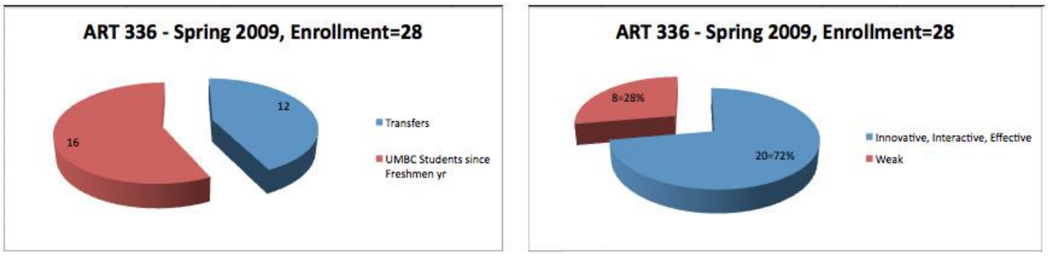


Figure 10. Enrollment and Evaluations.



5 CONCLUSIONS

All of the students from this class graduated in the Fall 2012 semester or will be graduating in Spring 2013. The three students described above have since entered the interactive graphic design field. Other students have taken freelancing or graphic design roles where they are creating interactive interfaces in various environments, and another is planning to go back to school to obtain a Master's degree in graphic design.

The design education community needs to be aware of the urgent need expressed by graduating students to understand all currently available technologies. Without addressing this need, students may settle for a job with a lower salary because of their lack of capabilities and because schools are ineffectively preparing their students. As the field of interactive graphic design continues to grow, students are being forced to learn the basics at a minimum. Furthermore, the competitive job market leaves no other option but innovation. Our responsibility as educators is to push the limits of our curricula. I look forward to the future knowing that the advancement of graphic design is occurring concomitantly with developments in other fields. Many courses need to be reevaluated in graphic design programs, and I believe this paper generates more questions than answers. Nevertheless, it offers a good starting point for solving the fundamental problems in the field of interactive graphic design.

The tools created for graphic designers are not sufficient, and the more interactive our environments become, the greater is the need for graphic designers to learn about interactivity. Therefore, programming is part of the curricula and demands a new methodology geared towards graphic designers. Engineering design education has a broad perspective of learning new technologies and innovations. However, although graphic design is part of ART and has its strength in creativity, it lacks a suitable methodology to teach students the interactive rules that comprise the important programming basics. The group of UMBC students analyzed herein and the evaluation carried out by AIGA both reveal the interest in learning programming as an engineering tool, which confirms that using an engineering design methodology in the classroom is becoming more relevant to the graphic design field.

ACKNOWLEDGMENTS

I would like to thank John Jeffries, Dean of the College of Arts, Humanities, and Social Sciences, Vincent Grabill, Chair of Visual Arts, all the students that participated in the study, and the support of UMBC.

REFERENCES

Aquent and AIGA. 2012. *AIGA | Aquent Survey of Design Salaries 2012.* Available: http://designsalaries.aiga.org/#descriptions. Last accessed 01.08.13.

Kidwell, P. and Ceruzzi, P., 1994. *Landmarks in digital computing.* 1st ed. US: Smithsonian Institution. pp. 107-108.

Lawson, B. 1980., *How designers think*. Westfield, NJ: Eastview Editions.

Lie W. and Bos B., 2005. *Cascading style sheets designing for the web*. 3rd ed. US: Pearson.

Lieberman, H., 2001. *Your wish is my command programming by examples*. 1st ed. US: Morgan Kaufmann Press. pp. 7-19.

Lupton, E. and Phillips, J., 2008. *Graphic Design the New Basics*. 1st ed. US. Princeton Architectural Press.

Maeda J., 2004 *Creative code*. US: Thames & Hudson.

Maeda, J., 2006. *The laws of simplicity*. 1st ed. US: MIT Press. pp. 15-21.

Manovich, L., 2008. *Software takes command.* 1st ed. San Diego: University of California. p. 62.

Murray, J., 2012. *Inventing the medium*. 1st ed. US: MIT Press. pp. 71-73.

Norman, D., 2011. *Living with complexity*. 1st ed. US: MIT Press. pp. 254-263.

Myers, B., Young Park, S., Nakano, Y., Mueller, G., 2008. How Designers Design and Program Interactive Behaviors. IEEE Symposium on Visual Languages and Human-Centric Computing, 2008.

Pahl, G. and Beitz, W., In Marjanovic, D., Storga, M., Pavkovic, N. and Bojcetic, N., eds. *International Design Conference - DESIGN 2010*, Dubrovnik, University of Zagreb/The Design Society.

Palo Alto Research Center Incorporated (PARC). 2012. *Event - Innovation: Design – Is design innovation?* Available: http://www.parc.com/event/1774/innovation.html. Last accessed 01.07.13.
Reas C. and Fry B., 2007. *Processing: A programming handbook for visual designers and artists*. US: MIT Press.
Rogers, Y., Sharp, H. and Preece, J., 2012. *Interaction design.* 3rd ed. Italy: John Wiley & Sons. p. 158.
Romsizowski, A. J., 1981. *Designing the user interface*. Reading, MA: Addison-Wesley.
Sebesta R., 2011. *Programming the World Wide Web*. 6th ed. US: Addison- Wesley.
Tidwell, J., 2005. *Designing interfaces*. 1st ed. US: O'Reilly Media Inc.