INCORPORATING DATA VISUALIZATION IN A COURSE ON COMPUTER GRAPHICS

Timothy Urness
Department of Mathematics and Computer Science
Drake University
Des Moines, IA 50311
515 271-2118
timothy.urness@drake.edu

ABSTRACT

Data visualization techniques and tools can facilitate the identification of meaningful correlations within data as well as exploring new models and theories. Visualization has become an integral component for computer science programs that support data analytics or "big data" curriculums. This paper describes an introductory computer graphics course that incorporates learning objectives outlined in the ACM-IEEE curriculum guidelines for a unit on data visualization. We identify challenges related to teaching an introduction to computer graphics course, particularly at small-to-medium sized universities, and propose a unit on data visualization within an introductory computer graphics course with example assignments and student work.

INTRODUCTION

A course in computer graphics can be taught in many different ways. One approach is to introduce a specific programming approach (e.g. OpenGL, DirectX, mobile devices, rendering engines, ray tracing) and explore graphics through this tool in depth. An alternative is to focus on the fundamental theories that are universal to all of computer graphics. This decision of depth vs. breadth isn't unique to computer graphics, but is exacerbated by the uniqueness of each tool, how quickly the tools change, and the amount of study required to master a particular rendering option. Furthermore, a professor has to balance how the course fits into the curriculum goals at the school: e.g. a professional-preparation program vs. a liberal-arts approach.

Visualization has become an important component in the analysis of data, identifying meaningful correlations and exploring new models and theories [2]. The techniques and tools that allow for accurate and fast visualizations are a consistent component of data analytics or "big data" programs, and have become a valuable, marketable application of computer graphics.

In this paper, we explore the challenges of teaching an introductory course in computer graphics and make recommendations of incorporating a unit on data visualization. We present a six-session unit covering data visualization principles, example assignments, and exemplars of student work.

CHALLENGES TEACHING COMPUTER GRAPHICS

A traditional computer graphics course typically includes topics such as basic rendering, lighting and shading calculations, applications of linear algebra (e.g. affine transformations), and texture mapping. For many years, introductory

computer graphics courses and textbooks have frequently focused on the programming of these effects using OpenGL, a multi-platform API for rending 2D and 3D graphics. The result of this approach is that the course can become programming-intensive, requiring an in-depth knowledge of the OpenGL API and rendering pipeline [7]. In the following section, we identify challenges that are considered when developing an introductory to computer graphics course.

Student Expectations

Computer graphics is often a very enticing course for students that have been already exposed to the glamorous manifestations of computer graphics such video games, special effects, and animated movies. Students enter the course with high expectations, and may not be prepared for the mathematical nature or programming requirements of the subject [10].

Small-to-Medium Sized Universities

The content related to computer graphics is vast. Many programs at larger universities will spread the content over several courses. For example, individual courses could focus on animation, modeling, rendering, ray tracing, etc. At small-to-medium sized universities, offering more than an introductory course is not feasible due to either a lack of faculty resources or expertise, and a lack of student enrollment. The curriculum at smaller schools is typically designed to emphasize a broad "liberal-arts" exposure to computer science and not necessarily a focused, deep concentration on a particular topic. Professors teaching at these schools are forced to strike a balance between foundational computer graphics theory and the technical details of a graphical programming language.

Hardware Dependencies

Instructors teaching computer graphics courses continually face a challenge to keep up with recent trends in graphics hardware [1]. The new OpenGL paradigm requires instructors to reexamine how to teach the course, as the introductory "Hello GL" program requires a deep understanding of graphics hardware in order for students to produce something visible. Furthermore, utilizing native OpenGL requires specific graphics hardware to utilize the API effectively. Supporting OpenGL across multiple operating systems with a variety of graphics hardware can be an overwhelming task. While options such as WebGL (www.khronos.org/webgl/) may provide a potential solution, getting all students enrolled in the course rendering on a consistent platform remains a potential obstacle to overcome.

ACM-IEEE 2013 Guidelines

In the ACM-IEEE 2013 computer science curriculum recommendations, only 3 classroom hours are recommended in the "Graphics and Visualization" knowledge area [5]. We propose that students taking a computer graphics course (particularly within a liberal arts curriculum) could potentially be better served by a redesigned course that incorporates elements of computer graphics alongside topics from visualization and software engineering, which could cover more of the

recommended topics in the curriculum recommendations as well as more generally applicable skills and tools.

A BREADTH-FOCUSED INTRODUCTORY COURSE WITH DATA VISUALIZATION AND SOFTWARE ENGINEERING

An alternative to a programming centric course focused on OpenGL is to emphasize conceptual and foundational elements, providing students with hardware specific code only as needed [7]. Sacrificing the cutting-edge for foundational elements, this course can better emphasize the underlying computer graphics principles, incorporate data visualization issues, and emphasize good software design. The introduction to computer graphics course outlined below includes principles of graphics, and also introduces elements of software engineering and an added a unit on visualization.

Software Engineering

Many computer science students will graduate and pursue a career dealing with some aspect of software. The ACM-IEEE guidelines support this notion and suggest 28 hours of *software engineering* classroom instruction over the course of the curriculum [5]. Incorporating software engineering concepts can be done in a very natural way in a computer graphics course, as basic elements of software engineering are necessary when learning graphics programming. Topics such as program correctness (unit tests), library components and APIs, debugging strategies, code repositories, documentation and programming style are very natural lessons for novice programmers in a graphics course.

Visualization

The presentation of data can be equally as important as the processing of collecting it. The field of visualization examines how to effectively represent data to enhance understanding. Accurate and fast data visualization has become an important component in the analysis of "big data." The visualization of data is an essential tool for identifying meaningful correlations and exploring new models and theories [2].

While creating a graph or a table from some data is trivial, few have been trained how to construct visualizations that best communicate the data in an accurate, succinct, and effective fashion [3,4]. Furthermore, there is an increasing need in industry to process and visualize large data sets in a reliable and efficient manner. Visualization skills and tools are becoming an important part of a software engineer's skill set.

The ACM-IEEE curriculum task force [5] has identified the following learning objectives in the "Graphics and Visualization" knowledge area.

- Propose a suitable visualization design for a particular combination of data characteristics and application tasks.
- Analyze the effectiveness of a given visualization for a particular task.
- Recognize a variety of applications of visualization including representations of scientific, medical, and mathematical data; flow visualization; and spatial analysis.

These learning objectives were the foundation of a six-session unit on data visualization that is summarized in table 1.

Session	Topic	Resources
1	Foundation for a science of data visualization	[8, 13]
2	Effective Visualization; Colormaps	[9, 11]
3	Perception: attentive vs. preattentive processing, color theory	[3, 13]
4	Effective use of tables, graphs, scatterplots	[4]
5	Things to avoid (e.g. pie charts, representing scalar data with width and height, rainbow colormap)	[4, 9, 11]
6	Applications: scientific, medical, mathematical, flow visualization, spatial analysis. Visualization Toolkit (vtk)	[6, 12]

Table 1: A proposed six-session unit in data visualization that can be implemented into a graphics course.

The study of data visualization requires students to experiment with the creation of charts, graphs, and diagrams. Using a free cross-platform visualization software package such as Tableau (www.tableausoftware.com/), D3 (http://d3js.org/), processing (https://processing.org/), gnuplot (www.gnuplot.info/), or an extension of the pythyon programming language mathplotlib (http://matplotlib.org/), students are able to create effective visualizations of large data sets. Our course utilized Tableau, and students were given a free version of Tableau Desktop for the semester (www.tableau.com/academic).

Example Assignments

Assignments during the visualization unit require students to think critically about the effectiveness of their representations they generate. The free, sample data sets provided at www.tableausoftware.com/public/community/sample-data-sets or www.data.gov/ provide students with "real" data to visualize.

Assignment Number	Description
1	Given a data set that consists of all of the Olympic medals won over the past decade (name, country, date, and medal type), generate a recommendation, supported by at least 3 graphs or tables, to the United States Olympic Committee on which Olympic events to increase financial support with the goal of increasing the medal count won by American Olympians.
2	Given a data set, create an effective table, graph, and scatterplot. Describe why you chose to represent the data with the visualization method. What conclusions can you draw from your visualization?

3	Find an example of a bad visualization of data. Next, redesign it in a way that will better reflect the data. Describe in detail why your representation improves the original visualization.

Table 2: Example data visualization assignments.

Student Work

Students rarely struggle to find examples of poor data visualization. Assignment 3 (table 2) asks students to identify an example of a poor visualization and redesign it to be more effective.

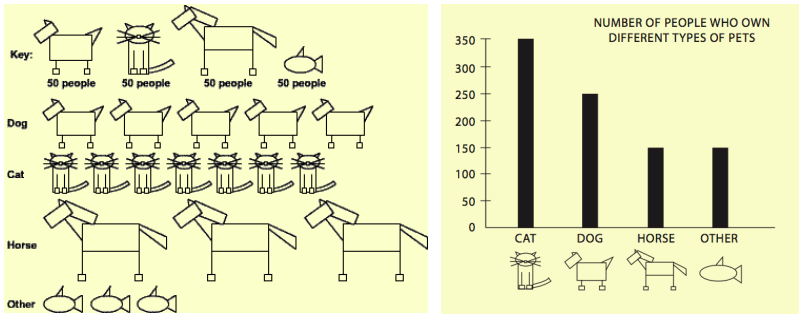

Figure 1 (left): Visualization in which the scales of the icons are not uniform. (right) A student submission for an alternative visualization of the data depicted that is more succinct and allows for a more accurate analysis.

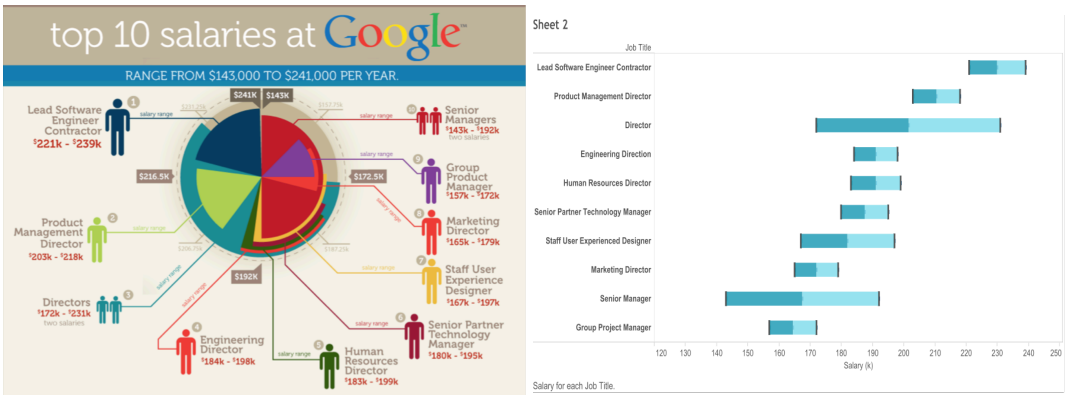

Figure 2 (left): A complex visualization representing salaries at Google. The goal of the visualization isn't clear and is overly complicated. (right) A student's simplified visualization of the data.

There are many ways that information can be distorted when visualized. An example of an inaccurate visualization of data is to use icons of different sizes or shapes to represent a (quantitative) value. Figure 1 (left) shows and example of this visualization error in practice, while figure 1 (right) displays a student's representation using a bar graph that more accurately represents the data. Visualizations can also fail because they are too complicated and do not represent the data in an effective manner. Figure 2 (left) illustrates this issue with a confusing chart. The student representation (right) facilitates the understanding of the data.

CONCLUSION

Professors at small-to-medium sized universities teaching an introduction to computer graphics course must balance the instruction of fundamental concepts with the details of a specific implementation programming language. As an alternative to a programming-centric course, we have proposed a course that covers broad topics, including a unit on data visualization. We feel that this course serves students in a liberal-arts program well and also aligns with the ACM-IEEE curriculum guidelines.

The data visualization unit was incorporated in a class taught in the fall of 2014 at Drake University. The section was very popular with the students with several remarking that it was the highlight of the course. After the course, several students have continued their study of computer graphics with an independent study on OpenGL and data visualization.

ACKNOWLEDGEMENTS

A special thanks to the students that took part of this class. Thank you to Marcel Finan (Arkansas Tech University) and Ben Hoffmann (www.jobvine.co.za) for allowing the use of the images displayed in the figures.

REFERENCES

[1] Angel, E., Shreiner, D., Teaching a shader-based introduction to computer graphics, *IEEE Computer Graphics and Applications,* 31 (2), 9-13, 2013.
[2] Bollier, D., Firestone, C. M., *The Promise and Peril of Big Data*, Washington, DC: Aspen Institute, 2010.
[3] Few, S. *Now You See It: Simple Visualization Techniques for Quantitative Analysis,* Oakland, CA: Analytics Press, 2009.
[4] Few, S. *Show Me the Numbers: Designing Tables and Graphs to Enlighten,* , Burlingame, CA: Analytics Press, 2012.
[5] Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*, 2013.
[6]Kitware Inc., *The VTK User's Guide,* Kitware, Inc. 2010.
[7]Lowther, J. L., Shene, C.-K., Rendering + modeling + animation + postprocessing = computer graphics, *Journal of Computing Sciences in Colleges,* (1), 20-28, 2000.
[8]Munzner, T., *Visualization Analysis and Design*, Boca Raton, FL.: Taylor and Francis, 2014.
[9]Rogowitz, B., Treinish, L.A.,. Data visualization: the end of the rainbow, *IEEE Spectrum,* 35 (12), 52-59, 1998
[10] Shesh, A., Toward a Singleton Undergraduate Computer Graphics Course in Small and Medium-sized Colleges. *ACM Transactions on Computing Education* 13(4), 17:1-17:21, 2013.
[11]Tufte, E., *Visual Explanations: Images and Quantities, Evidence and Narrative,* Cheshire, CT.: Graphics Press, 1997.
[12]Telea, A. C., *Data Visualization: Principles and Practice*, Wellesley, MA.: A K Peters, Ltd., 2008.
[13] Ware, C. 2012. *Information Visualization: Perception for Design*, Waltham, MA.: Morgan Kaufmann, 2012.