Zoomable User Interfaces
CMSC 838B
Syllabus
Spring 1998
Prof. Ben Bederson

Course:

Zoomable User Interfaces
CMSC 838B
3180 A.V. Williams Building (HCIL-2)
Wednesdays 1-4pm
http://www.cs.umd.edu/~bederson/classes/cmsc838b
mailto:cmsc838b-0101@coursemail.umd.edu
news://news.umd.edu/csd.cmsc838b

Instructor:

Ben Bederson
3171 A.V. Williams Building
Office Hours: Wed, 4-6pm
bederson@cs.umd.edu

Class Texts:

None

Recommended Texts:

Designing the User Interface, 3rd Edition Shneiderman [Addison-Wesley], 1998.
Readings in Information Visualization Card, Mackinlay, and Shneiderman, eds. [Morgan Kaufmann], 1999.

Zoomable User Interfaces are an experiment in creating alternatives to the standard Windows, Icon, Menus, Pointer (WIMP) GUIs that we have all come to know and “love” with modern computers. The goal of this course is to study ZUIs and related alternatives to GUIs, with the hope of understanding some of the complexities of creating new interfaces that are more usable than traditional interfaces. One of the complexities is that “usable” is a broad term that is hard to measure. It can include speed of operation, ease of use, ease of learning, accuracy of operation, satisfaction of use, etc. To make things even harder, each of these characteristics of usability can be different for different tasks.

This is a *research* course. As such, there is no standard curriculum or textbook. We will learn about ZUIs by reading and discussing research papers, looking at other researcher’s software, building our own software, evaluating our software, and writing about it. There will be very little lecture in this course, especially as we start working on our projects.

This is a *projects* course. There are no exams. The primary mechanism we will learn with is by doing. Early in the semester, you will choose a project and write a short proposal explaining what you want to work on. You may choose to work by yourself, or work in a team of 2 or 3 people. The complexity of the project should be proportional to the number of people working on it. The project may involve programming, evaluation, or a combination of both. We will discuss the projects in more detail early in the semester.

My research group has been building a generic substrate in Java called *JPad* which you will use for all of your projects. As such, a requirement for this course is that you can program in Java. If you do not know Java now, you should be fluent in C++ or another object-oriented programming language, and expect to spend a significant amount of time on your own learning Java in the first month of class. We will spend class time discussing JPad, and advanced features of Java.

This course addresses a specific area of the field of Human-Computer Interaction (HCI). If you have not taken CMSC434 or the equivalent, you should read *Designing the User Interface* on your own early in the semester to catch up on some basic background.

Homework

The homework for this course will consist primarily of the following three things:

- Read papers, writing summaries of the papers on the web, and presenting the papers in class. Each paper that we read in class will be facilitated by one class member. The leader must read the paper with particular care, and must prepare a 500-1000 word analysis of the paper. This analysis should include a summary of the contribution of the paper, pointers to particularly relevant other work or web sites, and your comments. Your comments should include what you think are the particularly important contributions, and troubling areas of the paper. Finally, include a list of questions that you had while reading the paper. You must email me your analysis *no later than the Sunday night before class*. I will add your analysis to the class web page first thing Monday morning so the entire class has two days to read your analysis.

  To lead the class discussion, you should start with a short (5-10 minute) presentation about the paper. Summarize the main points of the paper, and since everyone will have already read the paper, the summary should not be long. Present any interesting points for discussion (issues, connections with other papers we've read), open questions, and critiques. Note that you are leading the discussion, although Prof. Bederson may comment along the way.

  After the class meets, the leader must revise the analysis based on the class discussion, adding any questions that the class had about the paper. Then, the leader must email the author(s) of the paper with their comments/questions and a link to the class web page and your analysis, and report back to the class with the author's response (if there is one). You must 'cc' me in your email to the authors, and if you're concerned about contacting them, I will be happy to preview the message before you send it.

  All students are required to read the paper and the facilitator's comments *before* class, and participate in the discussion.

- Do your independent or group project, presenting progress reports and a final demo in class. Each group will have the opportunity to get the entire classes feedback twice during the semester during the scheduled *critiques*. You must prepare for the critique. You should expect to give a short (5-10) minute presentation discussing what you are working on, and you *must* show your project, whatever state it is in. We will then have a class critique of your project, discussing what looks good, and what you can do to improve your work.

- Write a conference-quality paper about your project. Whether you actually submit your paper to a conference or not depends on how good your paper ends up being. But your goal is to convince me that your paper is good enough to submit to a research conference. The two primary conferences you should aim for are the conference on Human-Computer Interaction (CHI) at http://www.acm.org/sigchi/chi99 or the conference on User Interface and Software Technology (UIST) at http://www.acm.org/uist . The UIST deadline is May 7, 1999 so you'll have to be ahead of schedule to submit to it this year. The CHI conference deadline is September 1999.

Software

You can use any machines and any Java development environment you choose. Sun's JDK 1.2 is currently available for free for Windows 95/98/NT and Solaris. It will be available for Mac, Linux, and other operating systems at some point, but Sun has not promised any dates. Your code must run under Windows NT with Sun's JDK 1.2 for your presentations. Since there are some differences between platforms with Java, you should be sure to test your code under Windows before your presentations. See the course home page for links to sites about Java.

Lab Access

You can use the lab we meet in (3180 AVW, HCIL-2) to work on your class projects. Note that you can not use this lab for work unrelated to this course. Also note that some people work in this lab and will have higher priority access to the machines than you. As such, you can not be sure you will always have access to the same machine. Therefore, you must keep your personal files in the shared folder for this class (see below). Finally, this lab is unavailable certain times. We will keep a calendar with reserved hours on the door to the lab. Unfortunately, this lab has a lot of use, and we are unable to accommodate people that are not working for projects in this class.

All of your personal files on the lab computers must be stored in the "CMSC 838B" folder on the desktop. This folder is shared by all computers in the lab. Therefore, you can use any computer in the lab and have access to your project files. Please be aware that if your files do not reside in that folder, they will be deleted. In the class folder, you will be expected to create a folder with your name on it, where you can place all of your files.

If you would like a copy of your files on a personal Zip disk, you can do so from at least 4 computers in the lab. In addition, all lab computers are backed up by our support staff for emergency purposes. At the end of the semester, if you would like to keep any of your projects, you will need to make a copy of your files. The last day of the semester, all course files will be removed in preparation for next semester.

Writing

This course involves a lot of writing. Your writing will be graded on both the content and the quality of your writing. Good writing is incredibly important and is typically understressed in Computer Science courses. Think of this course as a wonderful opportunity to practice your writing as well as your technical skills. And don't plan on getting an "A" if you don't put effort into your writing.

Grading

All assignments are due at the **beginning** of the class on the day that they are due. The paper part must be turned in at class, and the electronic part must be submitted by the time of the beginning of class. Late assignments will be strictly penalized. Exceptional circumstances will be considered **only** if discussed with me in advance. All late assignments will have points deducted as follows:

-20%	Up to 24 hours late
-50%	Up to 48 hours late
-100%	More than 48 hours late

This course is going to be very interactive, and a significant part of our study will occur in class. Thus, you are required to attend class. I will include your attendance in your class participation grade.

All writing assignments are graded based on two equally weighted criteria: content and writing quality. Your final grade will be computed using the following contributions:

10%	Participation (in-class and on-line)
20%	Paper summaries and presentations
35%	Final Project
35%	Project paper

It is well understood that working in interdisciplinary teams is a difficult process, but a necessary learning experience. The computer systems developed commercially today are primarily the result of team collaboration. Therefore, I encourage you to consider this development experience. In doing so, if there is ever an issue, or difficulty that you would like to talk about concerning your team collaboration, please feel free to talk to me at any time. Every effort will be made for your team collaboration experience to be an enjoyable one!

Because commercial teams are evaluated primarily by the product the group delivers, your grade will primarily be a group project grade. However, I will ask each of you to evaluate your group experience, and describe what each member contributed to the project. I expect to give each group member the same group project grade, but I reserve the right to give different grades to group members.

Zoomable User Interfaces
CMSC 838B
Dates
Fall 1998
Prof. Ben Bederson

(1) **Feb 3**	Intro Pad++ demo Schedule reading assignments How to write an analysis	**(8)** **Mar 31**	Reading: [32] [19] Critique 3
(2) **Feb 10**	Reading: [11] [27] [8] JPad overview	**(9)** **Apr 7**	Reading: [13] [4] Critique 4
(3) **Feb 17**	Reading: [16] [23] [18] Java2D and Swing	**(10)** **Apr 14**	Reading: [5] [3] Critique 5
(4) **Feb 24**	Reading: [29] [21] [20] JPad more detail **Project Proposal due**	**(11)** **Apr 21**	Reading: [2] [12] Critique 6
(5) **Mar 3**	Reading: [26, Chap 1] [28] [17] Schedule critique assignments	**(12)** **Apr 28**	Reading: [6] [14] Critique 7
(6) **Mar 10**	Reading: [25, Chap 5] [22] [10] Critique 1	**(13)** **May 5**	Reading: [33] [9] Critique 8
(7) **Mar 17**	Reading: [24] [15] [7] [1] Critique 2	**(14)** **May 12**	**Project Presentations** **Paper due**
Mar 24	**Spring Break** (Makeup class: [30] [31])		

READINGS

1. Ahlberg, C., & Shneiderman, B. (1994). Visual Information Seeking: Tight Coupling of Dynamic Query Filters With Starfield Displays. *In Proceedings of Human Factors in Computing Systems (CHI 94)* ACM Press, pp. 313-317.

2. Baecker, R., & Small, I. (1990). Animation at the Interface. B. Laurel *The Art of Human-Computer Interface Design* (pp. 251-267). Addison-Wesley.

3. Bartram, L., Henigman, F., & Dill, J. (1995). Intelligent Zoom As Metaphor and Navigation Tool in a Multi-Screen Interface for Network Control Systems. *In Proceedings of IEEE International Conference on Systems, Man and Cybernetics* IEEE, pp. 3122-3127.

4. Bartram, L., Ho, A., Dill, J., & Henigman, F. (1995). Continuous Zoom: a Constrained Fisheye Technique for Viewing and Navigating Large Information Spaces. *In Proceedings of User Interface and Software Technology (UIST 95)* ACM Press, pp. 207-215.

5. Beard, D. V., & Walker, J. Q. I. (1990). Navigational Techniques to Improve the Display of Large Two-Dimensional Spaces. *Behaviour & Information Technology, 9*(6), 451-466.

6. Bederson, B. B., & Boltman, A. (1998). *Does Animation Help Users Build Mental Maps of Spatial Information?* Tech Report CS-TR-3964, Computer Science Department, University of Maryland, College Park, MD.

7. Bederson, B. B., Hollan, J. D., Druin, A., Stewart, J., Rogers, D., & Proft, D. (1996). Local Tools: An Alternative to Tool Palettes. *In Proceedings of User Interface and Software Technology (UIST 96)* ACM Press, pp. 169-170.

8. Bederson, B. B., Hollan, J. D., Perlin, K., Meyer, J., Bacon, D., & Furnas, G. W. (1996). Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing, 7,* 3-31.

9. Bederson, B. B., & Meyer, J. (1998). Implementing a Zooming User Interface: Experience Building Pad++. *Software: Practice and Experience,* (in press).

10. Card, S. K., Robertson, G. G., & Mackinlay, J. D. (1991). The Information Visualizer, an Information Workspace. *In Proceedings of Human Factors in Computing Systems (CHI 91)* ACM Press, pp. 181-188.

11. Card, S. K., Mackinlay, J. D., & Shneiderman, B. (1999). Information Visualization. S. K. Card, J. D. Mackinlay, & B. Shneiderman (eds.), *Readings in Information Visualization: Using Vision to Think* (pp. 1-34). California: Morgan Kaufmann.

12. Dill, J., Bartram, L., Ho, A., & Henigman, F. (1994). A Continuously Variable Zoom for Navigating Large Hierarchical Networks. *In Proceedings of IEEE International Conference on Systems, Man and Cybernetics* IEEE, pp. 386-390.

13. Donelson, W. C. (1978). Spatial Management of Information. *In Proceedings of Computer Graphics (SIGGRAPH 78)* ACM Press, pp. 203-209.

14. Donskoy, M., & Kaptelinin, V. (1997). Window Navigation With and Without Animation: A Comparison of Scroll Bars, Zoom, and Fisheye View. *In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 97)* ACM Press, pp. 279-280.

15. Druin, A., Stewart, J., Proft, D., Bederson, B. B., & Hollan, J. D. (1997). KidPad: A Design Collaboration Between Children, Technologists, and Educators. *In Proceedings of Human Factors in Computing Systems (CHI 97)* ACM Press, pp. 463-470.

16. Furnas, G. W. (1986). Generalized Fisheye Views. *In Proceedings of Human Factors in Computing Systems (CHI 86)* ACM Press, pp. 16-23.

17. Furnas, G. W. (1997). Effective View Navigation. *In Proceedings of Human Factors in Computing Systems (CHI 97)* ACM Press, pp. 367-374.

18. Furnas, G. W., & Bederson, B. B. (1995). Space-Scale Diagrams: Understanding Multiscale Interfaces. *In Proceedings of Human Factors in Computing Systems (CHI 95)* ACM Press, pp. 234-241.

19. Furnas, G. W., & Zacks, J. (1994). Multitrees: Enriching and Reusing Hierarchical Structure. *In Proceedings of Human Factors in Computing Systems (CHI 94)* ACM Press, pp. 330-336.

20. Furnas, G. W., & Zhang, X. (1998). MuSE: A Multiscale Editor. *In Proceedings of User Interface and Software Technology (UIST 98)* ACM Press, pp. 107-116.

21. Hightower, R. R., Ring, L., Helfman, J., Bederson, B. B., & Hollan, J. D. (1998). Graphical Multiscale Web Histories: A Study of PadPrints. *In Proceedings of ACM Conference on Hypertext (Hypertext 98)* ACM Press, pp. 58-65.

22. Jul, S., & Furnas, G. W. (1998). Critical Zones in Desert Fog: Aids to Multiscale Navigation. *In Proceedings of User Interface and Software Technology (UIST 98)* ACM Press, pp. 97-106.

23. Lamping, J., Rao, R., & Pirolli, P. (1995). A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. *In Proceedings of Human Factors in Computing Systems (CHI 95)* ACM Press, pp. 401-408.

24. Marcus, A. (1997). Graphical User Interfaces. M. G. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (2nd ed., pp. 423-440). North-Holland.

25. Nielsen, J. (1993). *Usability Engineering*. AP Professional.

26. Norman, D. A. (1988). *The Psychology of Everyday Things*. BasicBooks.

27. Perlin, K., & Fox, D. (1993). Pad: An Alternative Approach to the Computer Interface. *In Proceedings of Computer Graphics (SIGGRAPH 93)* New York, NY: ACM Press, pp. 57-64.

28. Plaisant, C., Carr, D., & Shneiderman, B. (1995). Image-Browser Taxonomy and Guidelines for Designers. *IEEE Software, 12*(2), 21-32 Ben S. said this was his best paper for discussing the overview + detail approach.

29. Plaisant, C., Mushlin, R., Snyder, A., Li, J., Heller, D., & Shneiderman, B. (1998). *LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records.* Tech Report CS-TR-3943, Computer Science Department, University of Maryland, College Park, MD.

30. Robertson, G. G., Mackinlay, J. D., & Card, S. K. (1991). Cone Trees: Animated 3D Visualizations of Hierarchical Information. *In Proceedings of Human Factors in Computing Systems (CHI 91)* ACM Press, pp. 189-194.

31. Ryall, K., Marks, J., & Shieber, S. (1998). An Interactive Constraint-Based System for Drawing Graphs. *In Proceedings of Human Factors in Computing Systems (CHI 98)* ACM Press, pp. 97-104.

32. Schaffer, D., Zuo, Z., Bartram, L., Dill, J., Dubs, S., Greenberg, S., & Roseman, M. (1993). Comparing Fisheye and Full-Zoom Techniques for Navigation of Hierarchically Clustered Networks. *In Proceedings of Graphics Interface (GI 97)* Canadian Inforation Processing Society, pp. 87-96.

33. Stone, M. C., Fishkin, K., & Bier, E. A. (1994). The Movable Filter As a User Interface Tool. *In Proceedings of Human Factors in Computing Systems (CHI 94)* ACM Press, pp. 306-312.