

Joel Ross Piper - p129196 - 3D Menu System for Computer Games - Report

by Joel Ross Piper

FILE	15244_JOEL_ROSS_PIPER_JOEL_ROSS_PIPER_-_P129196_-_3D_MENU_SYSTEM_FOR_COMPUTER_GAMES_-_REPORT_78196_1217642272.DOCX (4.68M)		
TIME SUBMITTED	20-APR-2016 01:35PM	WORD COUNT	5227
SUBMISSION ID	56122342	CHARACTER COUNT	29081



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

3D Menu System For Computer Games

by
Joel Ross Piper

Supervisor: Dr. Glenn L. Jenkins
Second Marker: Gordon Dickers

Project submitted as part of the requirements for the
award of BSc (Hons) Computer Games Development

April 2016

DECLARATION

I, ... declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed :

Date :

FORM OF CONSENT

I hereby consent that my Project, submitted in candidature for the degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed :

Date :

Abstract

The purpose of this project is to create a system/library that can be integrated in to a program, most likely during development, and used to create menus for said program that will be shown as 2D objects all of which would be rendered in a 3D environment, navigation from one menu to the next would be done through small movement animation that take place within the 3D environment. The primary method used for this project is a literature review of related research and literature that may provide insight into things like usability, user interface design, and other project that may have previously gone over subject areas of use to this project. This project has provided some interesting looks into usability and user interface in the field of computer games development and how important they are to the users' experience.

Table of Contents

Abstract.....	3
1.0 – Introduction.....	5
2.0 - 3D Menu System for Computer Games – Literature Review	6
2.1 - Literature Review Introduction	6
2.2 –Usability.....	6
2.2.1 – Usability Engineering	6
2.2.2 – Cognitive Walkthrough.....	7
2.2.3 – Usability in games.....	8
2.3 - Libraries & frameworks for menus/UIs.....	10
2.3.1 - 2D Menus	10
2.3.2 - Qt.....	10
2.3.3 - Existing Frameworks and Libraries.....	11
2.4 –Existing Work in similar areas.....	12
2.4.1 –Examples of existing 3D menus and interactions in games.....	12
2.4.2 – 3D User Interfaces	13
2.5 - Conclusion.....	14
3.0 – Research and Development Methodology	15
4.0 – Design & Implementation.....	16
4.1 - 3D Menu Library – User Requirements	16
4.1.1 - End User Requirements.....	16
4.1.2 - API Requirements.....	17
5.0 – Evaluation	19
6.0 – Conclusions and Recommendations	19
7.0 – Reflection.....	20
8.0 - References	21
9.0 – Bibliography	23

1.0 – Introduction

The way users interface with computers and the software on them has developed significantly over the decades since their creation, from punch cards and vacuum tubes in the beginning to the graphical user interfaces (from here referred to as GUIs) of today is a tremendous leap forward and user interfaces continue to evolve and change to this day. Despite the modern implementation of GUIs as a fairly standard interface method for software the actual appearance and interaction with the software can vary wildly between programs causing potential users some trouble when attempting to grasp the use of these programs. In no other area is this more prevalent than computer games. Due to the accessibility of computer game development and the vast number of people interested in the field thousands, or even hundreds of thousands, of computer games exist today, many with varying degrees of quality, and the majority of which need menus or interface systems of some kind. However not all programmers are well versed in graphics design, this is the problem that this project is meant to fix; by creating an easy to use, visually appealing menu creating library that can be integrated into theoretically any program we remove the need to expend resources on interface development.

The objective of this project is the development of a menu library as described above, one that can be integrated into the majority of programs that can or will be developed. The menus created by this library would be 2D rectangular objects, rendered into a 3D environment, upon which data or options specified by the developer would be displayed, should the user activate an option that is supposed to take them to a different menu a kind of animation will take place where the ‘camera’ that the user views the 3D environment through will move along a pre-generated path to the desired menu ending with the menu being shown front and centre on the screen for the user to interact with. This being a 3D environment depending on where the menus are placed within the environment the user should be able to see other menus in the ‘background’ while interacting with a menu.

The following chapters will talk about the development of the project and research that relates to the project or that may be of use when developing a user interface.

2.0 - 3D Menu System for Computer Games – Literature Review

2.1 - Literature Review Introduction

The purpose of this project is to develop a library/API/framework that can be used by programmers to create a menu, or menus, for their software. This paper will examine research related to the project goal, beginning with usability, the work of Jakob Neilson and others in his field and usability with regard to video games, then we will look at existing libraries and frameworks for information on how they work and if they can be of any use to our research, finally we will review research in similar areas to our own for the sake of comparison; how or what have they done and can it be of use to us.

2.2 –Usability

Since this project is one that involves the design of a user interface as an essential part of the requirements then it was determined that research into usability would provide useful insight and techniques that will assist the development of the project. Usability is defined by Oxford Dictionaries as ease of use or user friendliness[1]. For the sake of developing or building a product that is to be used by another person it is necessary to test the usability of said product. The purpose of usability testing is to determine any problems as user may find in the use of the product pre-emptively to said problems occurring and fix them. Using usability testing is also meant to improve the method by which those that are developing the product develop future products thereby foregoing the need for usability testing on said future products. Many methods have been found and detailed that can be used to evaluate the usability of a product during its development.

2.2.1 – Usability Engineering

There have been many works of literature and research papers on usability and its uses. One of the earliest pioneers in usability engineering, Jakob Nielson[2] whose book on the matter is still referenced today, has written extensively on the subject and been a part of much of the literature in many ways, either as an editor or his work having been referenced within said literature.

Nielsons' book, Usability Engineering, is widely regarded as a formative work that established a foundation from which usability in software development grew into what it is today. The book is quite thorough in its explanation of the process one must go through during development to properly evaluate the usability of your product. One of the points touched upon in the book is the balance between usability in the terms of novice users and usability in the terms of experienced users[2]the usability requirements of a novice user may relate to the ease of learning the various functions or

interactive elements within the product whereas the usability requirements of an experienced user may relate to the speed with which they can activate certain functions within the product or how well information is relayed to them, the focus of an experienced user is in efficiency of use. Another aspect of usability engineering mentioned within is the lifecycle of usability[2]Nielson talks about this in relation to when usability engineering takes place describing it as “... *not a one-shot affair where the user interface is fixed up before the release of the product.*” Instead usability engineering is said to be something done throughout the life of a product, constantly refining and tweaking the product to meet usability requirements and improve it. Another important part of usability engineering is knowing your user, if you have no idea who your user base is then any or all of the usability evaluations you’ve made in relation to a theoretical user requirement may be completely off base.

2.2.2 – Cognitive Walkthrough

Nielson also edited a book called Usability Inspection Methods[3] which details methods for the evaluation of the usability of a product. One such method of particular interest, due to its relation to the usability of software, within the book is the Cognitive Walkthrough method detailed by Wharton et al[4] in Chapter 5 of the book. The Cognitive Walkthrough method, in their own words: “... *is a usability inspection method that focuses on evaluating a design for ease of learning, particularly by exploration.*” This is the case because it has been observed that many users prefer to learn software by exploration. The Cognitive walkthrough method is done by choosing one or more specific user tasks and evaluating them; going through the tasks and at each point where the user would need to provide input asking yourself what the user would be doing. If the design of the interface is well done then the proposed action the user undertakes should be the correct one for the task they intend to do. However if the design is poor and the action the user will undertake is not obvious or correct then those reviewing the usability of the product must ask themselves why this is the case and what they can do to change it.

In the development process the Cognitive Walkthrough method comes almost any time after the requirements analysis and definition as long as there is some kind user interface design that the usability testing can be applied to. This means that the method can be performed at many different points within the development cycle of the software, it may even be performed multiple times if necessary, for example if the design of the UI changes over time. When it comes to the project this method of evaluating the usability of the product may be more suited than other methods because of the way it works in relation to the future users.

2.2.3 – Usability in games

Video games being another kind of software also benefit from usability engineering and inspection methods, in fact because of the incredibly wide audience they reach usability is likely a vital part of the games design process. In the games industry there are a number of different monikers for video games depending on the financial strength and place of the developer in the industry, the most common of these being AAA (Triple A) and Indie. Triple A usually refers to a game made by a developer with large budgets, large teams of staff, and/or the strong backing of a 'big' publisher, for example the video game series Call of Duty is a Triple A series. Triple A video games are commonly recognised by their high production quality and, though there are some exceptions, the usability of these games can often be said to be top notch. Indie games on the other hand are categorised by their relatively small budgets and the small amount of people who work on them, often times less than 10 people will have worked on an indie game. Indie games also commonly have poor production quality, though again this cannot be said of all such games. Though the exact definition of these categories cannot be pinned down the point is that people can often tell the difference primarily through sight and this can be attributed to the design philosophy differences between big corporation and small teams, meaning that many indie developers do not realise the importance of usability or do not even test it.

There have in fact been works that discuss the importance of usability in video games including a study of usability in MMORPGs (Massively Multiplayer Online Role Playing Games) [5] and research funded by the Microsoft Game Studio on User-Centred Design (UCD) [6] published in the Handbook for Human-Computer Interactive Systems.

MMORPGs are a massive part of the games market these days and the nature of their development and lifecycle means that usability can mean the life or death of the game; a successful MMORPG can have upwards of millions of players at a time with thousands of new players joining at any point, retaining these users as the lifeblood of the game, and possibly even the developer, would be next to impossible if the usability of the games interface is abysmal. The research done by Cornett [5] involved a number of persons taking part in an usability study of a number of video games, primarily MMORPGs, for the sake of evaluating any usability issues presented to new players in a game and how they compared to the issues encountered in the other games they explored. As stated within the work [5] due to their nature MMORPGs are quite difficult to properly test for usability as they may change gradually over time or there may be a sudden update that drastically changes the user interface of the game. Although there is this risk of sudden or constant

change in the UI is can also be observed that the majority of these UIs follow a similar pattern in their design and the arrangement of their UI elements, as shown below:



Figure 1

As shown above MMORPG user interfaces contain 4 primary points of interest; the Minimaps (Located top right Figure 1), the use chat (Located bottom Left Figure 1), the user character information (Located top right Figure 1), and the 'Hotbars' containing the users abilities and techniques (Located bottom middle Figure 1). These elements can be found in the user interface of nearly every MMORPG on the market and could therefore be a standard UI element that is often unchanging and may be assessed for usability.

Although usability as a whole, including software usability, has produced innumerable works on assessment of, and designing with, usability, works focused on the usability of video games and their user interfaces are lacking in presence and in many cases they are demonstrably needed. There is however a research group created by Microsoft Game Studios focused on the examination and research of usability in video games whom have written a number of works on the subject, though the majority are not available to the public at large. One such work that is available to the general public is a chapter in the Handbook for Human-Computer Interactive Systems focusing on game design that is centred on the user and the usability of the game. This work speaks of the issues unique to the development of video games as compared to that of 'regular' software, one such issue unique to video games is the threat of a challenge to the player becoming a "...*frustrating usability*

problem”[6]. Such potential ‘pot holes’ in development mean that the usability of a video games user interface is not the only part that need be tested but instead the usability of the product as a whole should be assessed, this can also provide additional problems with the testing as not all users will have the same problems with the same aspects of the game meaning that even with usability testing a game may not be completely usable for all parties but can still be as usable as possible.

The research [6] goes into great detail as to the different parts and systems within a game that can require usability testing; how the player is rewarded for overcoming challenges, how functions are mapped to an input device, how the player is taught to use the games systems i.e. tutorials, how feedback is provided to the player in regards to their actions and current status within the game, and the rate at which the player encounters a challenge or detail within the game they haven’t seen before. The research goes on to speak of case studies in usability and game testing they have done themselves.

2.3 - Libraries & frameworks for menus/UIs

2.3.1 - 2D Menus

Most software and videos games in existence today use what would be referred to as 2D menus for the sake of facilitating user interaction. When we talk about 2D menus we are referring to a menu displayed on, and interacted with through, a singular graphical plane.

2.3.2 - Qt

Qt is a cross-platform framework most prominently used for the development of app software that can be run on a large variety of software and hardware platforms with little to no changes to the underlying code. Qt is open source software that is developed by the Qt Company as well as other individual developers and firms.

2.3.3 - Existing Frameworks and Libraries

Below is a table (taken from the internet) of existing frameworks and libraries that are used in the design of User interfaces:

Existing UI Frameworks/Libraries		
Name	Platforms	Integrated Graphics Backend
Qt	Windows/Linux/OS X/iOS/Android	OpenGL(Extra Module), Wayland
CEGUI	Windows/Mac/Linux	Direct3D, OpenGL, OGRE 3D Engine, Irrlicht Engine
FLTK (Fast, Light Toolkit)	X11/Windows/Mac	OpenGL
GTK+ (GIMP Toolkit)	X11/Windows/DirectFB/Quartz on Mac OS X	OpenGL, Wayland, HTML 5
Guilliani	Windows/ X11/ Mac/ DirectFB/ BareMetal/ Android/ iOS	OpenGL. OpenGL ES, Windows GDI, SDL, eGML, CAFE, D/AVE
IUP	Windows/ X11	Windows API, Motif/Lesstif, GTK+, CanvasDraw(LUA Scripts), Native C/C++ application
JUCE	Windows/ X11/ Mac/ iOS/ Android	CoreGraphics, OpenGL
MKS Toolkit	Windows from X11 code	X, ncurses, Motif, OpenGL

2.4 –Existing Work in similar areas

2.4.1 –Examples of existing 3D menus and interactions in games

2.4.1.1 – Call of Duty: Black Ops 3(Developed by Treyarch and Published by Activision)



[7]



[8]

2.4.1.2 – Bioshock Infinite (Developed by Irrational Games Published by 2K Games)



[9]

2.4.2 – 3D User Interfaces

3D user Interfaces are the one of the most uncommon interface systems used in modern, or any, software. When we speak of 3DUIs we may be referring a UI that is displayed within a 3D space, a UI whose structure is 3 Dimensional, or even a UI that is designed to look 3D, there is no real standard definition or idea of what a 3D UI is or how one should work. Although uncommon there is a fair amount work on 3D user interfaces and interaction around including two works we shall be examining by Doug Bowman and his colleagues [10][11]

Bowman's book [10] talks about the theory that goes into 3D user interfaces and how to evaluate them. The majority of the talk about 3D user interfaces in the book speaks of them in reference to their use within virtual environments like those used in virtual reality. Bowman also speaks of means through which one would evaluate a 3D user interface, describing how existing evaluation methods like usability testing could be used, and outlines a number of metrics with which to assess the 3D user interfaces quality.

2.5 - Conclusion

Research and papers related to user interfaces, their design, their development, and their functionality, are widespread and varied and the primary focus of many is the experience of the user and how programs may be impacted by how the user interprets the interface. There are also numerous papers on the proper development of user interfaces using usability studies and testing. Usability testing and the research involved in that area has become an integral part of user interface development and has been evolving for a few decades now. 3D user interfaces also appear to be the subject of a number of research projects that have been appearing in recent years.

Figure Captions

Figure 1. An example of the UI of a MMORPG, this one being from World of Warcraft.

3.0 – Research and Development Methodology

The research method used for this project is primarily literature review. The reason this method was used was to take advantage of the abundant research materials available in the world that talk about usability, user interfaces, computer games, and games design. The advantages to this method are that countless research papers and projects have been published that could provide background or insight into areas relevant to whichever project you intend to work on, in fact some areas have thousands of papers published that could be used. There are some disadvantages to this method however as if you find yourself researching a relatively new or scarcely talked about area then available research may be sparse and this could prove to hinder you progress, to work around this it is best to find related areas or research that may not necessarily be exactly like your own but could provide insight into areas that you lack.

The intended development methodology for this project is iterative and incremental development, scaled down for a single person. This is an agile methodology. This methodology was chosen for the purpose of allowing the development of the project to remain flexible as the design of the program may undergo many changes even after it is complete as it is a library that must be maintained in order to keep it compatible with a wide variety of potential programs. The software to be used in the development of this project is as follows; Microsoft Visual Studio will be used as a development environment, OpenGL will be used for graphical matters such as the rendering of the menus, and GLUT will be used temporarily as a go between to manipulate OpenGL, for things like creating a window, etc.

4.0 – Design & Implementation

Design

4.1 - 3D Menu Library – User Requirements

4.1.1 - End User Requirements

4.1.1.1 - The user shall be able to interface with menus created using a keyboard

Rational: Keyboards are the most prevalent and simplest of interaction devices used with computers, and should the library be used in a video game bound for consoles inputs using a controller can be mapped quite closely to a keyboards inputs e.g. up arrow on a keyboard – up on a joystick on a controller.

4.1.1.1.1 - The user may be able to interface with menus created using a mouse

Rational: The majority of modern software and video games on PC allow mouse interaction in menus

4.1.1.2 - The user shall be able to select options on the menus. This would be done with either a specific button on the keyboard (e.g. Enter/Return) and should mouse interaction be available with a left click of the mouse

Rational: Without interaction on this level most menus are practically useless

4.1.1.2.1 - The user should be able to alter variables or change options via the menus. This would be done after selecting the option (as in 1.2) using either the arrow keys on the keyboard or the left click/scroll wheel on the mouse

Rational: The user should be able to alter a variety of options provided to them by the programmer via the menus.

4.1.1.3 - The user shall be able to see the menus displayed on their screen.

Rational: The user should be able to see what they are interacting with.

4.1.2 - API Requirements

4.1.2.1 The API shall be able to create a menu. This menu should have a number of options on it linked to variables or functions in the main program to allow for the desired interaction

Rational: The basic requirements for this library are that it should allow programmers to create menus and link them in to their software with ease

4.1.2.1.1 The API should be able to create multiple menus. These menus should be linked through options specified by the programmer that are selectable on the menus

Rational: Many video games and software require more than one menu to properly convey options and functions to the user

4.1.2.2 The API shall be able to create interactive elements on the menu. These elements may take the form of sliders, spin boxes, or drop down lists of items specified by the programmer

Rational: A menu without interaction is useless and different kinds of interaction are useful for different options and functions.

4.1.2.2.1 The API should be able to create an interactive element that 'takes' the user to another menu upon selection.

Rational: If the programmer is creating multiple menus there needs to be a way for the user to navigate between them.

4.1.2.3 The API shall be able to display the menus on the screen. This shall be done via a graphics back end.

Rational: The programmer needs to be able to display the menus so that the user can interact with it.

4.1.2.3.1 The API shall be able to display buttons on the menus.

Rational: The programmer needs to be able to display buttons for the user to interact with.

4.1.2.3.2 The API shall be able to display interactive elements on the menus. These elements may take the form of sliders, spin boxes, or drop down lists of items.

Rational: The programmer needs to be able to display the elements for the user to interact with.

5.0 – Evaluation

Unfortunately the development of this project has not been complete. The literature review provided ample information on usability and the design of a user interface and insight into current ongoing research to do with 3D interfaces however as the project did not reach this stage this information was of little use. Very little, if any, objectives or aims were met during the development of this project, this is most likely the fault of the project developer, myself, lacking experience, a clear grasp of time management and a nose dive in motivation and effort, although there are some outside factors that may have held a small part in this, they are not wholly responsible and thus do not bear mentioning. If this task were to be taken up again a drastic change in work ethic and discipline would be required, the commencement of development should have taken place much earlier and the project developer, myself, should have taken a more active role in trying to make time for work and if necessary travel to the university much more frequently for the sake of working in an environment that is not as comfortable and distraction prone as the home.

6.0 – Conclusions and Recommendations

During the course of this projects development months have passed and many things have happened, the primary focus of research undertaken has been on usability and user interfaces and, although that may have been useful, there are other areas that haven't been explored that would have provided insight that would've been vital to the project in later stages, stuff to do with library development, graphical engines, and even animations and spline may have been of great help at a later stage in the projects development.

6.1 – Suggestions for further work

Further work on this project would be for the purpose of completing the project to its original specifications; the current code base, a simple maths library adapted from previous years' work, would have to be expanded greatly, design diagrams would have to be drawn up, further research into animation and compiling libraries would need to be done, and proper documentation would have to be produced.

7.0 – Reflection

I'm not sure if first person is allowed to be used and have been trying my best to avoid it but I see no way to properly complete this section without writing in first person. This project has been an unfamiliar experience for me and although going into it I was at least semi confident I could produce a rudimentary program for display I have unfortunately found myself unable. Although I did not find it enjoyable I still found some interest in the research I was finding and the subject areas they covered, unfortunately I find myself to be underwhelming when it comes to analysis and adapting research for my own needs. I also seem to find myself inexplicably unable to force myself into doing work at home without expending more effort than should be necessary, this combined with the fact that I am hesitant pay £4.20 to catch the bus into university to actually do my work has been a great detriment to my progress. I know I could have done much better and if given the chance maybe I could have but what's done is done and currently I plan to just finish my university course and move on to the job market.

8.0 - References

- [1] "usability: definition of usability in Oxford Dictionaries (American Writers Thesaurus)," *Oxford Dictionaries*. [Online]. Available: http://www.oxforddictionaries.com/definition/american_english-thesaurus/usability. [Accessed: 02-Dec-2015].
- [2] J. Nielsen, *Usability Engineering*, vol. 44, no. 3. 1993.
- [3] J. Nielsen, "Usability inspection methods," in *Conference companion on Human factors in computing systems - CHI '94*, 1994, vol. 25, no. 1, pp. 413–414.
- [4] C. Wharton, J. Rieman, C. Lewis, and P. Polson, "The cognitive walkthrough method: A practitioner's guide," *Usability inspection*. pp. 105–140, 1994.
- [5] S. Cornett, "The usability of massively multiplayer online roleplaying games," *Proc. 2004 Conf. Hum. factors Comput. Syst. - CHI '04*, vol. 6, no. 1, pp. 703–710, 2004.
- [6] R. J. Pagulayan, K. Keeker, T. Fuller, D. Wixon, R. L. Romero, and D. V. Gunn, "User-centered Design in Games," *Human-Computer Interact. Handb. Fundam. Evol. Technol. Emerg. Appl.*, pp. 795–824, 2012.
- [7] Y. İ. Ergül, "Call of Duty: Black Ops III - İnceleme - Call of Duty: Black Ops III - İncelemesi | IGN Turkey," *IGN*, 2015. [Online]. Available: <http://tr.ign.com/call-of-duty-black-ops-3-1/81242/review/call-of-duty-black-ops-iii-incelemesi>. [Accessed: 19-Nov-2015].
- [8] M. Bailey, "Does Call of Duty: Black Ops 3 change the game? | moviepilots.com," *Movie Pilots*, 2015. [Online]. Available: http://moviepilot.com/posts/3634572?lt_source=external>manual. [Accessed: 19-Nov-2015].
- [9] WSGF, "Bioshock Infinite | WSGF." [Online]. Available: <http://www.wsgf.org/dr/bioshock-infinite/en>. [Accessed: 19-Nov-2015].

- [10] D. A. Bowman, E. Kruijff, J. J. Laviola, and I. Poupyrev, *Evaluation of 3D User Interfaces*. 2004.
- [11] R. P. McMahan, D. Gorton, J. Gresock, W. McConnell, and D. A. Bowman, "Separating the effects of level of immersion and 3D interaction techniques," *Proc. ACM Symp. Virtual Real. Softw. Technol. - VRST '06*, p. 108, 2006.

9.0 – Bibliography

- J. Nielsen, "User Interface Direstions For The Web," vol. 42, no. 1, pp. 65–72, 1995.
- J. Nielsen, R. M. Bush, T. Dayton, N. E. Mond, M. J. Muller, and R. W. Root, "Teaching experienced developers to design graphical user interfaces," Proc. SIGCHI Conf. Hum. factors Comput. Syst. - CHI '92, pp. 557–564, 1992.
- J. J. Nunamaker and M. Chen, "Systems development in information systems research," J. Manag. Inf. Syst., vol. 7, no. 3, pp. 89–106, 1990.
- D. B. Chertoff, R. W. Byers, and J. J. LaViola, "An exploration of menu techniques using a 3D game input device," Proc. 4th Int. Conf. Found. Digit. Games, pp. 256–263, 2009.
- P. Sweetser and P. Wyeth, "GameFlow: a model for evaluating player enjoyment in games," Comput. Entertain., vol. 3, no. 3, p. 3, 2005.
- C. Hand, "A Survey of 3D Interaction Techniques," Comput. Graph. Forum, vol. 16, no. 5, pp. 269–281, 1997.
- A. Ezust and P. Ezust, An Introduction to Design Patterns in C++ with Qt 4. 2006.
- S. K. Cart, "User Perceptual Mechanisms in the Search of Computer Command Menus," in Conference on Human Factors in Computer Systems, 1981, p. 7.