

Mensch-Computer Interaktion mit der Maus

Julian Knocke
 University of Hamburg (Student)
 Nord Schleswiger Strasse 46
 22049 Hamburg
 +49 40 696 449 10

5knocke@informatik.uni-hamburg.de

ABSTRACT

An nahezu jedem Heimcomputer, jeder Workstation und fast jedem Server sind Maus und Tastatur angeschlossen. Dabei wird noch immer ein WIMP (Window Icon Menu Pointer) verwendet, doch seit vielen Jahren gibt es bereits Möglichkeiten, die Arbeit am Computer zu vereinfachen, zu unterstützen oder deren Effizienz zu steigern. Ich diskutiere dabei einige, teilweise auch schon umgesetzte Ansätze, welche hierfür dienlich sind. Im Fokus bleibt dabei noch immer der bei weitem noch nicht ausgereizte Ansatz, den Computer ausschließlich mit Maus und Tastatur zu bedienen.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces – graphical user interfaces (GUI), interaction styles, theory and methods, user-centered design.

D.2.2 [Software Engineering]: Design Tools and Techniques – user interfaces

I.6.3 [Computer Graphics]: Methodology and Techniques – interaction techniques.

General Terms

Design, Human Factors

Keywords

Maus, Tastatur, Pie-Menü, WIMP, Mousegestures, Multiblending



1.EINFÜHRUNG

In den vergangenen zwanzig Jahren hat sich im Bereich der Rechenstärke der Computer einiges getan. Wie in Tabelle 1 zu sehen ist, hat sich allerdings nicht nur die reine Rechengeschwindigkeit der CPU, sondern auch der Monitor, als visuelle Benutzerschnittstelle, stark weiterentwickelt. Die Bildschirmdiagonale sowie die Auflösung haben sich erhöht, die Pixel wurden kleiner. Man kann also rein theoretisch deutlich mehr darstellen. Wenn man sich nun aber die Benutzerschnittstellen anschaut, so hat sich trotz all des technischen Fortschritts, eigentlich nichts getan. Der WIMP (Window Icon Menu Pointer), als Synonym für eine grafische Oberfläche, ist gleich noch den Ersten, es werden auch noch immer mit Maus und Tastatur Eingaben gemacht. Ich werde jedoch ausschließlich auf weitere Möglichkeiten, welche die Maus dem Benutzer bietet, eingehen und einen Überblick über Vor- und Nachteile von unter anderem Pie-Menüs und Alpha Blending geben.

Die Benutzeroberflächen, welche von den meisten Nutzern verwendet werden, sind die Standard Windows Oberfläche, Aqua (die Standard Macintosh OSX Oberfläche) und die Linux Oberflächen KDE und Gnome. Vom grundlegenden Prinzip her unterscheiden sie sich nur marginal von einem der ersten kommerziellen WIMP, Apples Macintosh Desktop von 1984. Man arbeitet mit Icons, Anwendungen mit linearer Menüführung, Fenster, welche sich überlappen und einer

begrenzten Anzahl an Interaktionsmöglichkeiten wie beispielsweise „widgets“ (Menüs, Buttons, Dialogboxen, Scrollbalken), „drag&drop“ und „copy-paste“ [1].

Tabelle 1: Vergleich zwischen Computern von '84 und '03 [1]

	 Original Macintosh	 iMac 20"	Vergleich
Datum	Januar 1984	November 2003	+ 20 Jahre
CPU	68000 Motorola 8 MHz 0,7 MIPS	G5 1,26 GHz 2250 MIPS	x 156 x 3.124
RAM	128 KB	256 MB	x 2.000
Speicher	400KB Disketten	80GB Festplatte	x 200.000
Geräte	Maus Tastatur	Maus Tastatur	dasselbe dasselbe
GUI	desktop WIMP	desktop WIMP	dasselbe

Anhand drei Anforderungsgebieten kann man festmachen, dass der WIMP seine Grenzen erreicht hat: Das exponentielle Wachstum an Informationen mit denen sich jeder Benutzer auseinandersetzen muss; die Verteilung der Informationen über diverse Computer und andere technische Geräte wie PDAs oder Mobiltelefone; die wachsende Anzahl an Computernutzern mit verschiedenen Fähigkeiten, Erwartungen und Bedürfnissen [1].

2.NICHTLINEARE MENÜS

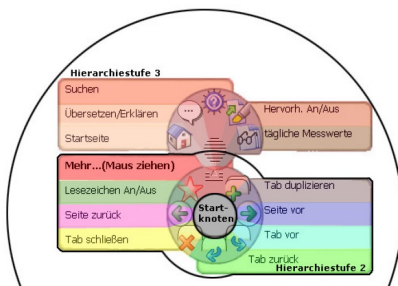
Lineare Menüs findet man zurzeit praktisch auf jedem Desktop. Sie existieren aber bereits seit den 60er Jahren des letzten Jahrhunderts. In der Zwischenzeit hat sich viel getan, es wurde viel auf diesem Gebiet geforscht.

Michael Geis und Marco Lehmann stellen eine Vielzahl möglicher Verbesserungen vor [2]. Um einige zu nennen wären da der „Cone Tree“ (1991), die „Perspective Wall“ (1991), der „Hyperbolic Tree“ (2000) sowie „Pie Menüs“. Im folgenden werde ich die Pie Menüs, sowie deren Weiterentwicklung, die „Mousegestures“, ausführen.

2.1 Pie-Menüs

Der Begriff Pie-Menü bedeutet aus dem Englischen übersetzt soviel wie Kuchen Menü. Bildlich kann man sich dieses Menü auch so vorstellen: ein Menü wird in diverse Kuchenstücke zweidimensional aufgeteilt. Dabei können allerdings vereinzelte Menüpunkte wieder zu einem eigenen Pie-Menü aufgehen. Dieses wird dann, wie bei den linearen Menüs, in mehrere Hierarchiestufen aufgeteilt (s. Abbildung 1). Der große Vorteil, den diese Menüs bieten ist die Möglichkeit, intuitive Menüführung durch direkte Handbewegungen zu realisieren. Wie bereits in dem Beispiel gezeigt, liegt „Seite vor“ auf der rechten Seite. Weiterblättern funktioniert dann wie folgt: Tastendruck und Maus nach rechts bewegen. Diese Bewegung kann man durchaus mit einem Buch vergleichen, bei dem man, um weiterzublättern, ebenfalls nach rechts greift, um die Seite umzublättern. Entsprechend ist „Seite zurück“ auf der linken Seite. Durch diese zweidimensionale Anordnung ist es also möglich, entgegengesetzte Operationen auch in entgegengesetzten Richtungen zu platzieren. Um ein weiteres Beispiel zu nennen, kann man anhand der Abbildung sehen, dass „Tab duplizieren“ rechts oben und „Tab schließen“ links unten ist.

Abbildung 1: Pie-Menü aus dem Mozilla Firefox [2]



Als Nachteil des „Pie-Menü“ kann man die fehlende Skalierbarkeit in Bezug auf komplexere bzw. umfangreichere Menüs nennen. Ein Knoten bietet effizient acht Möglichkeiten für Operationen oder Untermenüs. Darüber hinaus wird es schwer, die einzelnen Menüpunkte zielsicher zu treffen. Des Weiteren nimmt die Übersichtlichkeit mit wachsenden Menüpunkten rapide ab.

Für X-Basierte Systeme, wie z.B. Linux, gibt es den Windowmanager *piwm*. Dieser verfolgt das Ziel, alle linearen Menüs durch Pie-Menüs zu ersetzen.

2.2 Mousegestures

Wie bereits erwähnt sind die Mousegestures eine Weiterentwicklung der Pie-Menüs. Mousegestures zeichnen sich dadurch aus, dass man sie durch eine reine Bewegung der Maus in bestimmte Richtungen ausführen kann. Dadurch können bestimmte Operationen, wie das vorhin erwähnte Blättern von Seiten, noch intuitiver durchgeführt werden.

Dieses ist allerdings zugleich der entscheidende Nachteil. Beim „Herumspielen“ mit der Maus können nämlich die Bewegungen fälschlicherweise vom Computer als Mousegesture interpretiert werden und somit ungewollte Befehle auslösen. Des Weiteren fehlt bei den Mousegestures auch die optische Rückmeldung. Ein Pie-Menü ist genau beschriftet, wofür welche Richtung steht, Mousegestures muss man dagegen lernen. Sind sie aber erlernt, so kann eine Mousegesture die Effizienz beim Arbeiten enorm steigern.

Ein weiteres Anwendungsgebiet für Mousegestures sind PDAs oder im allgemeinen kleine „Computer“ mit Touchscreen. Hier wird mit den Fingern oder einem speziellen Stift direkt auf dem Display eine Figur gezeichnet, welche der Computer dann versucht zu interpretieren. Da man so auch Buchstaben

erkennen kann, kann sämtliche Art von Eingabeperipherie weggelassen werden.

3. FENSTERARRANGIERUNG

Die Arbeit auf einem Computerdesktop wird durch die Größe und Auflösung des Bildschirms begrenzt. Dadurch kommt man oftmals nicht daran vorbei, Fenster übereinander zu legen und so Teile zu verdecken. Abhilfe kann ein weiterer Bildschirm schaffen, doch nicht jeder hat den Platz, das Geld oder überhaupt Lust, sich mit zwei Bildschirmen beschäftigen zu müssen. Daher müssen auch Wege gefunden werden mit begrenzter darstellbarer Fläche aufgrund von Auflösung und Bildschirmdiagonale zu arbeiten.

3.1 Durchsicht von Fenstern

Gerade durch die enorme Steigerung der Rechenleistung sind uns da gute Möglichkeiten an die Hand gelegt worden: Transparenz.

Transparenz wird auch bereits auf diversen Desktops eingesetzt. Diese wird aber bisher ausschließlich durch das so genannte Alpha Blending verwirklicht. Dieses basiert auf einem konstanten Alphawert. Es errechnet die Summe aus den verschiedenen Layern, was dazu führen kann, dass sich Pixelwerte ergänzen oder auslöschen. Dadurch kann es passieren, dass für den Benutzer wichtige Bereiche unleserlich werden. Ein weiterer Nachteil ist die Unschärfbildung. Es ist möglich, dass sich zwei Anwendungen derart überlagern, dass der Nutzer nicht mehr unterscheiden kann, zu welcher Anwendung die Funktionen gehören oder ob die Funktionen in einem semantischen Zusammenhang stehen. Daher ist Alpha Blending zwar eine schöne Spielerei, stört jedoch wahrscheinlich eher, als dass es hilft.

Gerade für dieses Problem gibt es eine Methode von Patrick Baudisch (Microsoft Research) und Carl Gutwin (University of Saskatchewan) [2], welche sich mit dem Problem beschäftigt.

3.1.1 Multiblending

Beim Multiblending werden möglichst viele relevante Informationen aus sich überlagernden Layern geholt. Es werden dazu die Informationen eines Fensters in unwichtigere Informationen, welche dann einen stärkeren Alphawert haben, und wichtigere, welche dann besser zu sehen sind, unterteilt. Dabei wird zwischen Farben und Text unterschieden. Es wird eine Funktion genutzt, welche die Häufigkeit von Daten analysiert und hervorhebt. Dieses ist besonders in Ecken oder anderen Gebieten mit hohem Kontrast der Fall. Der menschliche Visualisierungsapparat ist darauf ausgelegt, genau diese Kanten wahrzunehmen.

Abbildung 2: Multiblending [3]



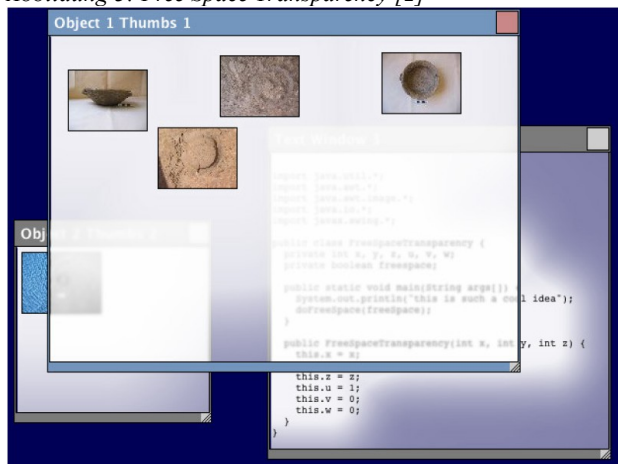
In Abbildung 2 wird ein Fenster mit Alpha Blending mit einem Fenster, auf das die Multiblending Funktion angewendet wurde, gegenüber gestellt. Daran kann man erkennen, wo die Probleme des Alpha Blending liegen. Der vordere Layer ist

kaum von dem Hintergrundbild zu unterscheiden. Im Gegensatz dazu kann man das mit Multiblending veränderte Fenster sehr gut von dem darunter liegenden Bild unterscheiden. Der Nachteil dieser Methode ist, dass man auf diese Weise Farben nur sehr schlecht darstellen kann. Im richtigen Anwendungsgebiet, wie beispielsweise Zahlenstatistiken oder Werkzeugleisten, ist es aber ein sehr mächtiges Werkzeug, welches auf dem Desktop für deutlich mehr Ordnung sorgt.

3.1.2 Free Space Transparency

Alternativ gibt es die Free Space Transparency. Sie verwendet einen Algorithmus zum Klassifizieren von unwichtigen Regionen. Hierbei wird sowohl von dem verdeckenden, als auch dem verdeckten Layer eine Wichtung der sich darauf befindenden Typen (wie Texte, Bilder oder Icons) und Farben gemacht. Es wurden hier diverse Ansätze von Wichtungen verfolgt. Bei einem wird lediglich der reine farbige bzw. texturierte Hintergrund eines Fensters als unwichtig angenommen. Ein anderer Ansatz lässt den Nutzer dies entscheiden. Einerseits geht das mit direkter Auswahl unwichtiger Bereiche, andererseits kann man über die Augenbewegung feststellen, welche Bereiche wichtig sind. Ein weiterer Ansatz stellt fest, wie wichtig bestimmte Bereiche sind. Dieses erfolgt im Allgemeinen über eine Gauß'sche Verteilung mit z.B. dem Radius 3 bei textähnlichen Objekten und 5 bei Grafiken oder Icons [4].

Abbildung 3: Free Space Transparency [2]



In Abbildung 3 kann man dann sehen, dass sich an der Lesbarkeit der Inhalte nichts verändert, jedoch unbenutzte Bereiche verschieden stark durchsichtig sind.

4.AUSBLICK

Es bleibt abzuwarten, welche Visualisierungstechniken sich ihren Weg in die WIMP zukünftiger Betriebssysteme bahnen. Dass dies ein bereits längst überfälliges Unterfangen ist, zeigt die Fülle an Informationen und Möglichkeiten, welche es zu bewältigen gilt. Microsoft hat bereits für Windows Vista angekündigt, einige Verbesserungen, was die Nutzbarkeit des Systems angeht zu implementieren aber auch bei Apple oder in der „X-Fraktion“ stehen die Räder nicht still.

5.ABSCHLUSS

Das Eingabegerät Maus hat ihr Limit noch lange nicht erreicht. Man braucht nicht einmal weitere Monitore oder andere Ausgabeperipherie, um die nötigen Informationen an den Nutzer zu bringen. Ich habe einige effektive Möglichkeiten dargestellt, mit denen sich die Nutzbarkeit der heutigen Desktops verbessern würde. Da aber gerade in diesem Bereich über so viele Jahre nichts getan wurde, stehen wir an einem Anfang die effektgesteuerte Unterstützung des Nutzers voranzutreiben. Als weiteres treibendes Glied in der Kette erweist sich die Hardwareentwicklung, welche mit immer schnelleren Prozessoren und Grafikchips auftrumpft. Was in Zukunft daher noch realisiert werden kann, ist nicht abzusehen. Abschließend kann man also zusammenfassen: Maus und Tastatur haben noch lange nicht ausgedient.

6.REFERENCES

- [1] Michel Beaudouin-Lafon (2004). Designing interaction, not interfaces. In *Proceedings of the Working Conference on Advanced Visual interfaces (AVI '04)* (pp. 15–22). ACM Press, New York, NY.
- [2] Michael Geis, Marco Lehmann (2005). WIMP Extended. In *Visualisierungs- und Interaktionstechniken für das Büro der Zukunft* (pp. 1–12). TU Dresden, Dresden.
- [3] Patrick Baudisch, Carl Gutwin (2004). Multiblending: displaying overlapped windows simultaneously without the drawbacks of alpha blending. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '04)* (pp. 367–374). ACM Press, New York, NY.
- [4] Edward W. Ishak, Steven K. Feiner (2004). Interacting with hidden content using content-aware free-space transparency. In *Proceedings of the 17th annual ACM symposium on User interface software and technology (UIST '04)* (pp. 189–192). ACM Press, New York, NY.
- [5] *ACM SIGCHI Curricula for Human-Computer Interaction* by Hewett, Baecker, Card, Carey, Garsen, Mantei, Perlman, Strong and Verplank (11.11.2005, 18:00). <http://www.acm.org/sigchi/cdg/cdg2.html>