



**HAL**  
open science

## **TFlex: Um Simulador de Teclado com Múltiplos Modos de Varredura**

Marcio R. Juliato, Eduardo H. Tanaka, Christian Baudet, Miguel Galves,  
Thiago T. Coelho, Heloísa V. Rocha

► **To cite this version:**

Marcio R. Juliato, Eduardo H. Tanaka, Christian Baudet, Miguel Galves, Thiago T. Coelho, et al..  
TFlex: Um Simulador de Teclado com Múltiplos Modos de Varredura. VI Simpósio sobre Fatores  
Humanos em Sistemas Computacionais (IHC 2004), Dec 2004, Curitiba, Brazil. hal-01093638

**HAL Id: hal-01093638**

**<https://inria.hal.science/hal-01093638>**

Submitted on 10 Dec 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# TFlex: Um Simulador de Teclado com Múltiplos Modos de Varredura

Marcio Juliato

Eduardo H. Tanaka

Christian Baudet

Miguel Galves

Thiago T. Coelho

Heloísa Vieira da Rocha

Instituto de Computação (IC) – Universidade Estadual de Campinas (UNICAMP)  
Avenida Albert Einstein, 1251, Caixa Postal 6176, CEP13084-971 Campinas, SP – Brasil

{marcio.juliato, etanaka, christian.baudet, miguel.galves, thiago.coelho, heloisa}@ic.unicamp.br

## RESUMO

Atualmente, existem inúmeros periféricos de entrada para computadores. Contudo, nem todas as pessoas podem utilizar os dispositivos mais comuns como o teclado e o mouse. Este é o caso de algumas pessoas portadoras de deficiências físicas e motoras. Para este público, uma alternativa é um simulador de teclado. Todavia, os simuladores de teclado atuais apresentam restrições que nem sempre podem ser superadas, não garantindo um bom nível de autonomia e independência dos usuários. Portanto, propõe-se um novo modelo de simulador de teclado, denominado TFlex, sem as restrições apresentadas até então em softwares similares. Pelo modelo proposto, um usuário portador de necessidades especiais pode ter completa independência no uso do computador utilizando esse simulador de teclado. Além disso, tentativas para melhorar a eficiência de uso foram implementadas, como diferentes modos de varredura e layouts de teclado.

## ABSTRACT

Nowadays, there is a number of computer input peripherals, however, not every person can use the most common input devices such as keyboard and mouse. This is the case of some people with physical and motor disabilities. For these people a keyboard simulator is a good alternative, but the current keyboard simulators have restrictions, which not always can be overcome, not assuring a good user's independence level. Therefore, a new keyboard simulator model is proposed, called TFlex, without the restrictions presented in others similar software. With the proposed model, a user with special needs could have total independence in computer while using TFlex. Besides, approaches to improve user's efficiency were developed, with different scan modes and keyboard layouts.

## Palavras Chave

Teclado Virtual, Simulador de Teclado, Tecnologias Assistivas

## INTRODUÇÃO

A Organização Mundial de Saúde (OMS) estima que 10% da população mundial é portadora de deficiência. No Brasil, dados do censo demográfico de 2000 do IBGE [4] apontam para mais de 14% da população como de portadores de deficiência, sendo que há mais de 7 milhões de pessoas com algum tipo de deficiência motora e quase um milhão e meio de portadores de deficiência física.

Apesar desses números serem altos, ainda há poucas iniciativas para integrar as pessoas portadoras de deficiência na sociedade. Ruas e avenidas não permitem a locomoção de usuários de cadeiras de rodas pela falta de guias rebaixadas. Nas bibliotecas, é muito difícil encontrar livros em Braille ou falados para permitir a “leitura” dos cegos. Soma-se a este descaso o preconceito e a imagem negativa que a sociedade faz da pessoa portadora de deficiência, considerando-a um cidadão especial no sentido negativo (não “normal”), incapaz de trabalhar ou estudar, conduzindo a ações segregadoras ou paternalistas.

Da mesma forma que o mundo físico ainda é uma barreira para os portadores de deficiência, o mundo virtual também o é. As tecnologias de informação e comunicação (TICs) contribuem ainda mais para a exclusão. Contudo, assim como em um prédio é possível construir rampas e elevadores para os usuários de cadeiras de rodas, nos computadores existem adaptações ou recursos capazes de promover a acessibilidade, conhecidas como tecnologias assistivas. Hogetop e Santarosa [3] definem tais tecnologias como o conjunto de recursos que, de alguma maneira, contribuem para proporcionar maior independência e qualidade de vida para os portadores de deficiência, contribuindo também para a inclusão.

Dentre as diferentes tecnologias assistivas existentes, encontra-se o simulador de teclado. Frequentemente, os simuladores de teclado apresentam na tela do computador uma representação de um teclado físico, isto é, um teclado virtual.

Contudo, um simulador de teclado se diferencia de um teclado virtual. Um teclado virtual apenas apresenta as teclas de um teclado na tela do computador e é utilizado, por exemplo, em websites de bancos para aumentar a segurança na entrada de dados, permitindo a entrada de dados através de dispositivos de apontamento como um mouse. Já um simulador de teclado, além de permitir a

escrita via dispositivo de apontamento, possui um sistema de varredura entre as teclas mostradas na tela e permite a usuários que possuem restrições motoras graves o uso de praticamente todas as opções disponíveis em um teclado com o pressionamento de uma única tecla do teclado físico ou um *switch* ou comutador (que nada mais é do que um botão ou chave), bastando que a varredura se encontre na tecla desejada. A Figura 1 mostra alguns comutadores [5] que podem ser utilizados junto com simuladores de teclado e outras tecnologias assistivas de software.



Figura 1. Exemplo de comutadores.

Além da varredura, outros recursos que frequentemente estão incluídos em simuladores de teclado são sistemas de predição de palavras e “leitor” de palavras, que lê aquilo que acabou de ser escrito. Entretanto, a maioria dos simuladores de teclado existentes atualmente sofre restrições diversas, mesmo sabendo que sua implementação é considerada simples.

#### ESTADO DA ARTE

Existem inúmeros simuladores de teclado no mercado e outros dispositivos de hardware e software que podem substituir o teclado físico. A seguir, serão analisados alguns dos mais populares entre eles.

#### Teclado Virtual do Windows XP

Apesar do nome, o Teclado Virtual presente no Windows XP possui um sistema de varredura simplificado. Ele é apresentado na tela como um teclado padrão QWERTY com 101/102 teclas, contendo todas as teclas de um teclado (incluindo acentos, teclas de função F1-F12, Caps Lock, Shift, Alt, Ctrl e até mesmo a tecla do menu Iniciar do Windows). As vantagens deste teclado sobre os demais é que ele acompanha o sistema operacional, não sendo preciso a compra e/ou a instalação de outros softwares, e é totalmente integrado ao sistema operacional, funcionando com praticamente qualquer software para Windows.

Há como configurar o teclado para não mostrar o teclado numérico, reduzindo os estímulos visuais e a quantidade de itens a serem percorridos pela varredura, e também é possível acionar uma opção para emitir som de clique quando se pressiona uma das teclas. Mas, diferente de um teclado físico comum em que se pode enxergar que no número 1 está o ponto de exclamação (!) quando se pressiona o Shift, no teclado virtual este e outros símbolos só irão aparecer após o pressionamento do Shift. A varredura acontece primeiro entre as linhas do teclado. Selecionada uma linha, a varredura se dá então entre as

colunas/teclas desta linha. A Figura 2 mostra a tecla *j* sendo percorrida pela varredura.



Figura 2: Teclado Virtual do Windows XP.

Quanto à varredura do teclado, só é possível configurar o tempo de varredura entre 0,5 e 3 segundos e de que dispositivo (joystick, porta serial, ou uma tecla do teclado físico, como a barra de espaço) deve-se capturar o sinal que indica a seleção da opção sendo varrida. A própria Microsoft admite que este teclado é limitado e que, para uso freqüente, o usuário deveria procurar outros teclados com mais funcionalidades.

#### Teclado Amigo

O Teclado Amigo é um software nacional, desenvolvido por pesquisadores do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro (NCE-UFRJ), sendo distribuído gratuitamente pela Rede SACI [11] como “Kit SACI 2”. O Teclado Amigo é um conjunto de programas que possui diferentes teclados virtuais com varredura. Estes programas são uma calculadora, um editor de textos, e um teclado básico que permite apenas a escrita de pequenas frases.

A velocidade da varredura no teclado segue uma escala numérica de zero a quatro (quanto maior o número selecionado, maior a velocidade). Porém, ao abrir a janela de configuração de velocidade de varredura, não fica visível qual é a velocidade atualmente em uso. Na Figura 3 observa-se o Teclado Básico. A disposição das teclas é, no mínimo, inédita e única. Separou-se entre subgrupos contendo vogais, consoantes, números e símbolos diversos (pontuação, sinais matemáticos, parênteses e outros). Também há teclas com vogais já acentuadas. Ilustrando falta de consistência, há alguns botões com letras em maiúsculas e outros com letras em minúsculas e sem nenhuma razão para tal diferenciação.



Figura 3. Teclado básico do Teclado Amigo.

O Teclado Básico e o Editor de Textos do Teclado Amigo possuem abreviações nem sempre fáceis de serem inferidas (ZUPT, ABRV, FALA, por exemplo). Além de não funcionar com outros programas, outro ponto negativo de todo o conjunto de programas do Teclado Amigo é que, para indicar qual é a linha ou tecla que está sendo varrida, os programas movem o cursor do mouse,

praticamente impedindo o uso do mouse concomitantemente com estes programas.

### Click-N-Type

Outro software distribuído gratuitamente, o Click-N-Type da Lakefolks [6] é um teclado com varredura similar ao Teclado Virtual do Windows XP e funciona em qualquer programa deste sistema operacional. O Click-N-Type, apesar de ser em Inglês, possui um pacote que traduz a maioria das mensagens e textos nos menus para uma mistura entre o Português de Portugal e do Brasil. Também, há muitas opções para customização.

Além da tradicional possibilidade de configuração da velocidade da varredura (entre 0 e 100, sendo 100 o mais rápido), é possível configurar o layout do teclado e escolher quais teclas devem ser mostradas e a ordem das teclas (alfabética, QWERTY, ou qualquer outra). Junto com o software, acompanham layouts de teclados QWERTY de 101 e 102 teclas e um teclado de tamanho reduzido, indicado principalmente para uso com o sistema de varredura. Há para download outros layouts, inclusive o ABNT2 utilizado no Brasil.

O sistema de varredura é o que o difere dos softwares anteriores. No Click-N-Type, existe um retângulo de contorno vermelho (cor padrão, possível de ser alterada) que se movimenta por blocos de teclas do teclado. Quando se pressiona um botão do teclado ou o botão esquerdo do mouse ou ainda um comutador ligado ao computador, uma linha vermelha passa a se mover horizontalmente dentro do retângulo. Outro pressionamento faz uma outra linha vermelha se mover verticalmente. Mais um pressionamento faz com que a tecla do teclado virtual na intersecção das linhas vertical e horizontal seja “clificada”, como ilustrado na Figura 4.

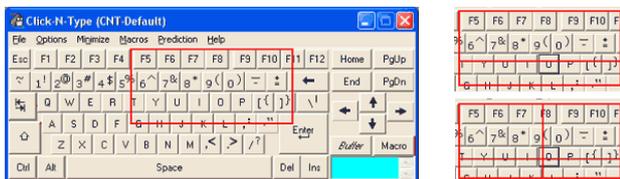


Figura 4. Telas do Click-N-Type.

Percebe-se que este tipo de varredura acaba produzindo mais pressionamentos de botões do que o do Teclado Virtual do Windows XP ou dos programas do Teclado Amigo. Outro ponto negativo é que o retângulo de varredura se move entre blocos de teclas e nem sempre as teclas ficam totalmente dentro do retângulo, exigindo que o usuário tenha uma certa coordenação motora para acertar a intersecção na tecla desejada. E, assim como o Teclado Amigo, enquanto a varredura estiver ativada no Click-N-Type, o cursor do mouse fica desabilitado. Apesar de não incluir em seu pacote básico, existe um meio de instalar um sistema de predição de palavras em Português.

### WiVik

Outro simulador de teclado com alguma similaridade com o Click-N-Type, o WiVik [14] tem como diferencial a possibilidade de “ler” letras, palavras ou frases na medida em que são escritas, dando um feedback sonoro ao usuário. Este software, assim como o Click-N-Type,

permite customizar totalmente o teclado virtual, inclusive colocando botões para, por exemplo, mover e redimensionar janelas, varrer opções de menus e atalhos como copiar e colar (Ctrl+C e Ctrl+V, respectivamente).

O WiVik é um software totalmente em Inglês que não oferece nenhuma forma de alteração do idioma. Também, o software pode não funcionar em todos os programas para Windows, como salienta o fabricante. Por fim, a licença para um único usuário custa US\$ 350,00.

### Simulador de Teclado – ST(NIEE)

Este simulador de teclado é um dos pioneiros no Brasil. A interface é baseada em modos “semi-gráficos” (gráficos gerados a partir de caracteres de texto), tipicamente comuns em programas baseados no sistema operacional MS-DOS. Trata-se, na verdade, de uma adaptação de um projeto da Universidade Técnica de Madri realizada por uma equipe do Núcleo de Informática na Educação Especial da Universidade Federal do Rio Grande do Sul [9].

Por se tratar de um software desenvolvido há muito tempo, logo na instalação o usuário é questionado sobre qual a porta serial será usada pelo acionador. Caso tenha sido feita uma escolha por engano ou mesmo se queira instalar um acionador em outra porta serial, será necessário reinstalar o software. A varredura implementada no Simulador de Teclado do NIEE é a de linhas por colunas, mas as letras são dispostas na tela seguindo um layout diferente do QWERTY. E, como não havia espaço na tela do MS-DOS para dispor todas as opções, existem sub-janelas para escolha entre vogais com acentos, números e símbolos matemáticos, teclas de função (F1-F12), e teclas para navegação no texto (home, end, page up, page down, etc.).

A pequena área reservada para ver a escrita do texto no Simulador de Teclado do NIEE só possui espaço para uma única linha, fazendo com que o usuário use muito das teclas de navegação pelo texto. A Figura 5 mostra uma tela deste simulador de teclado. Outro ponto negativo, por causa da idade do software, é que ele só consegue interagir com outros programas do MS-DOS. Mesmo assim, ao menos uma característica presente neste simulador de teclado é rara em outros mais recentes: a varredura por todas as opções de configuração, permitindo que o usuário possua independência e autonomia para configurar o software.

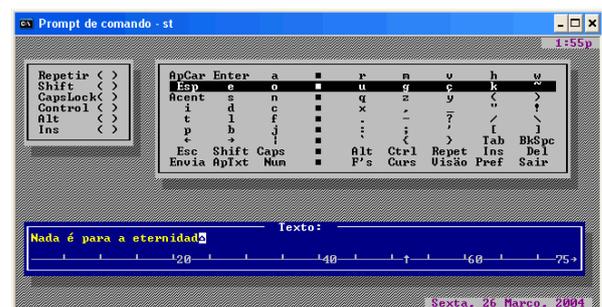


Figura 5. Simulador de Teclado do NIEE.

## Dasher

O Dasher [7] não pode ser considerado nem um teclado virtual nem um simulador de teclado. Contudo, a sua finalidade é a mesma: permitir a escrita no computador através de um meio alternativo ao teclado físico. Ele é uma interface para entrada de textos que utiliza um modelo de predição de palavras (em diversos idiomas, inclusive português) para tornar a escrita eficiente. Sua tela pode ser observada na Figura 6.

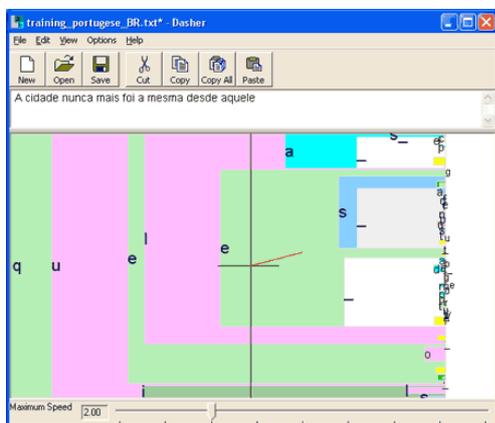


Figura 6. Tela do Dasher.

O software pode ser utilizado via mouse ou qualquer outro dispositivo que substitua o mouse, como um *eye tracker* (o autor do software afirma que a versão para *eye tracker* possibilita escrever tão rápido quanto por digitação normal, sendo também mais rápido do que teclados virtuais e simuladores de teclado). Outra vantagem do Dasher é que ele é um software livre e seu código-fonte pode ser modificado e distribuído.

A principal desvantagem do Dasher é que ele exige que o usuário tenha condições de mover um dispositivo de apontamento. Entretanto, para o público comum dos simuladores de teclado (pessoas com restrições motoras, principalmente nos membros superiores), o único tipo de dispositivo de apontamento que garantidamente atende a esta exigência seria um *eye tracker*, que é um dispositivo de rastreamento ocular de preço elevado.

## Via Voice e outros sistemas de reconhecimento de voz

É possível substituir o teclado e o mouse através de um sistema de reconhecimento de voz. A maioria destes softwares é de uso pessoal, já que vai se adaptando à voz do usuário. Depois de um período em que o software é treinado para reconhecer e entender a fala do usuário, praticamente todos os comandos do computador podem ser dados através do microfone. Há dois problemas no uso destes sistemas. Primeiro, a língua utilizada é predominantemente o Inglês. Segundo, nem todos são gratuitos, como é o caso do ViaVoice, da IBM.

## A PROPOSTA

Como visto, os simuladores de teclado e teclados virtuais existentes hoje são projetos que limitam o uso por pessoas portadoras de deficiência física ou motora. E mesmo outros softwares como o Dasher e o ViaVoice, que podem substituir um teclado convencional, também possuem restrições. Estas restrições podem ser devido à

falta de integração com o sistema operacional, impossibilitando o uso junto com outros programas, interferências com o mouse e o teclado físicos, ausência de varredura nas janelas de configuração, layouts inapropriados e custo elevado.

Entretanto, nota-se que o funcionamento de um simulador de teclado é bem simples e a sua implementação, a princípio, não pode ser considerada muito complexa, mas com enormes desafios de interface.

Assim, propõe-se o desenvolvimento de um simulador de teclado, o TFlex [12], para uso por pessoas portadoras de deficiência física ou motora, tendo o mesmo, recursos configuráveis que aumentam a facilidade e eficiência de uso.

Pelas condições culturais e sócio-econômicas do país, passa a ser interessante a distribuição do mesmo sem qualquer custo ao usuário, e em Português. Outro fator importante é que ele não seja difícil de instalar, configurar, aprender e usar.

A forma mais comum e tradicional de varredura em simuladores de teclado é a baseada em linhas e colunas, como se o teclado se comportasse como uma matriz. Neste tipo de abordagem, primeiro seleciona-se a linha em que a tecla desejada se situa, sendo em seguida realizada uma busca entre as colunas desta linha até que o caractere almejado seja encontrado.

Porém, para se chegar a um caractere desta matriz não necessariamente deve-se seguir essa abordagem tradicional, sendo possível adotar outro tipo de varredura. Uma aproximação razoável seria seguir um algoritmo de busca binária para encontrar, através de divisões sucessivas, a tecla desejada.

No projeto do TFlex, esses dois modos de varredura foram implementados, sendo os mesmos descritos nas seções a seguir. Além disso, foram estudados e implementados layouts alternativos ao QWERTY, como o alfabético, por exemplo, com o intuito de agilizar o processo de escrita e/ou facilitar a localização do caractere a ser digitado.

## LAYOUTS DO TECLADO

O layout QWERTY surgiu no final do século XIX, há 130 anos, junto com a invenção das primeiras máquinas de escrever. Desde então, é considerado o padrão universal para teclados não só de máquinas de escrever como também de computadores e outros dispositivos, embora outros layouts tenham sido propostos no século XX. O nome QWERTY vem das 6 primeiras letras dispostas na primeira linha do teclado [2].

Por ser considerado um padrão universal, utilizar o layout QWERTY no TFlex facilitaria o reconhecimento da disposição das teclas por usuários que já conhecem este layout. Mas o TFlex poderia utilizar um layout alfabético (com as teclas de letras dispostas em ordem alfabética), facilitando a aprendizagem e o uso por pessoas pouco acostumadas com um layout QWERTY. Ainda, poder-se-ia pensar em um layout, baseado na frequência de uso de caracteres na língua portuguesa. Ou seja, distribuir no layout do teclado os caracteres mais utilizados de tal

forma que não seria preciso esperar tanto para selecionar um caractere, de acordo com o tipo de varredura, seja ela tradicional ou binária.

Este último layout, sem dúvida, exigiria um tempo de aprendizagem maior para que os usuários localizassem os caracteres, uma vez que esse seria diferente do layout mais comum, o QWERTY, e também não disponibilizaria as teclas em uma ordem pré-definida, como no caso do alfabético.

O layout baseado em frequência de uso foi inicialmente pensado para a abordagem com busca binária, pois poder-se-ia distribuir os caracteres de forma a manter sempre no primeiro nó sendo varrido aqueles caracteres que são mais frequentes. Portanto, depois de aprendido, possivelmente seria o layout que levaria a uma eficiência maior (aquele em que se escreveria com maior agilidade).

### A BUSCA TRADICIONAL

A primeira abordagem concebida para desenvolver o TFlex foi a de varredura tradicional, em que o sistema de varredura é o mais simples possível: selecionando-se primeiro uma linha de teclas e, dentro desta linha, uma coluna (uma tecla).

Contudo, como se observa no caso do Teclado Virtual do Windows XP e do Simulador de Teclado do NIEE, o tamanho da tela é um fator que limita o número de linhas por colunas que podem ser mostradas pelo teclado virtual. Assim, é preciso ou desenhar teclas virtuais de tamanho reduzido o suficiente para que todas caibam na tela ou então trabalhar apenas com um conjunto de teclas em um dado momento.

Considerado estes aspectos, a varredura tradicional no TFlex foi implementada de forma a considerar 3 modos de operação: varredura em Grupos, varredura em Linhas e varredura em Colunas. O teclado varre primeiro os 4 grupos existentes: letras e símbolos, teclas de navegação (Home, End, etc.), números e teclas especiais (Alt, Ctrl, etc.). Quando o grupo desejado é encontrado, como mostrado na Figura 7, o usuário pressiona uma tecla (*Espaço*, no protótipo implementado) e a varredura começará a percorrer as linhas do grupo selecionado, como pode ser observado na Figura 8. Da mesma maneira, quando a linha desejada for alcançada, e um botão for pressionado, a varredura passará para o modo de colunas como mostra a Figura 9. Finalmente, quando o botão desejado for selecionado, um dos três eventos a seguir pode ocorrer:

- O caractere relacionado com a tecla será enviado para a caixa de texto, ou
- A janela relacionada com a tecla será exibida, ou
- A função relacionada com a tecla será executada, como por exemplo, a função que alterna as letras em maiúsculas e minúsculas.

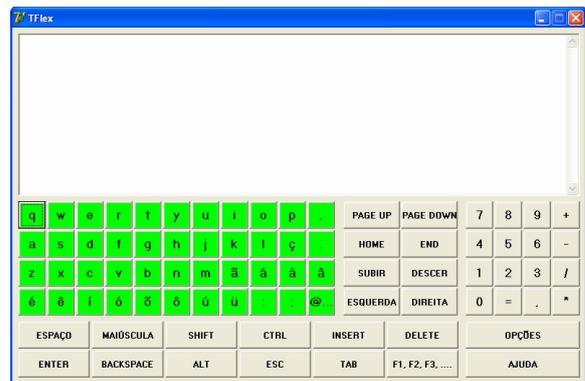


Figura 7. Varredura por grupos. Seleção do grupo de letras.



Figura 8. Varredura por linhas no grupo de letras.



Figura 9. Varredura por colunas no grupo de letras.

Se a varredura percorre duas vezes um conjunto de botões sem que nada seja pressionado, ela automaticamente retorna ao modo anterior. Por exemplo, se todas as linhas de um grupo são percorridas duas vezes, a varredura volta a percorrer todos os grupos a partir do grupo em que a linha se encontrava. Para usuários que possuem menores restrições motoras, foi implementado também um comando de volta. Ao pressionar a tecla *Esc*, a varredura volta ao nível anterior.

### A BUSCA BINÁRIA

Além da varredura tradicional, foi implementada no TFlex a varredura do teclado utilizando-se busca binária.

#### Por que utilizar varredura binária?

Tentando buscar alternativas para a varredura tradicional, propôs-se investigar simuladores de teclado com diferentes modos de busca pelos caracteres a serem digitados pelo usuário, como uma baseada no algoritmo

de busca binária. Nesse tipo de implementação divide-se sucessivamente o teclado em duas partes, sendo que o usuário sempre escolhe uma das divisões, a qual novamente é dividida em duas partes. Dessa forma seleciona-se, no final das divisões, o caractere desejado.

Para justificar a proposição de um novo meio de acesso às letras, faz-se necessário analisar como a maioria dos simuladores de teclado atuais funciona. Em várias implementações tem-se o teclado apresentado na tela e uma linha de teclas é mostrada em contraste (em azul no caso do Teclado Virtual do Windows XP). O usuário deve então selecionar a linha que possui a tecla que se deseja pressionar. Uma vez selecionada a linha, as letras da mesma são varridas em contraste, uma após a outra, até que o usuário pressione ou uma tecla do teclado, ou um botão do mouse, ou outro tipo de acionador/comutador, para selecionar a letra desejada. A velocidade de varredura é configurável, adequando-se assim a capacidade motora do usuário. Tomemos como base para os cálculos de nossa hipótese, um tempo de 0,8 segundos em que uma linha, ou uma tecla permanece em destaque, permitindo que o usuário a selecione. Tomando como base os teclados analisados anteriormente, os quais possuem, em média, cerca de 7 linhas, e que cada linha possui em torno de 15 teclas (colunas). Suponhamos ainda que o usuário seja capaz de selecionar a linha na primeira varredura, ou seja, ele tem tempo hábil para escolher a linha desejada, sem que uma segunda varredura seja efetuada naquela mesma linha. Dessa forma, temos um tempo de seleção médio de linhas de  $0,8 \text{ s} \times 3,5 \text{ linhas}$ , nos resultando em um tempo total de seleção de 2,8 s. Considerando o tempo médio de seleção da coluna, ou seja,  $0,8 \text{ s} \times 7,5 \text{ colunas}$ , nos resulta 6 s para que a letra seja selecionada dentro de uma coluna. No total, para que o usuário digite uma letra, são necessários, no mínimo,  $2,8 + 6 = 8,8 \text{ s}$ .

Considerando-se a busca binária, temos ao invés de linhas e colunas, a alternância entre dois grupos de teclas. Podemos estimar o tempo gasto para a digitação de um caractere quando utilizando esse tipo de abordagem. Para realizar os cálculos, como no teclado tradicional, manteremos em 0,8 s o período de tempo em que cada grupo fica em destaque, e consideraremos o melhor caso, no qual o usuário não espera por outro período para selecionar o grupo de caracteres desejado. Dessa forma, temos que, para um teclado de 104 teclas, serão necessárias  $\log_2(104) = 6,7$  interações. Tomemos como base os mesmos 0,8 s de período. Isso nos resulta em um tempo médio de acesso para um determinado caractere  $6,7 \text{ int.} \times 0,8 \text{ s} = 5,36 \text{ s}$ .

Assim, concluímos teoricamente que a abordagem de busca binária seria algo minimamente interessante de ser implementada. Entretanto, nota-se que para selecionar um caractere utilizando a busca binária, necessita-se de um número maior de interações. Dessa maneira, nota-se que existe uma relação de compromisso entre a rapidez de digitação e o número de interações necessárias. Assim, tomamos o teclado binário como algo interessante para pessoas capazes de realizar um número maior de interações, e tomando proveito desse maior grau de

liberdade obter-se-ia uma maior velocidade na digitação. Para pessoas com maior comprometimento motor continua certamente sendo mais recomendada a utilização do teclado utilizando a busca tradicional.

### Implementação da varredura binária

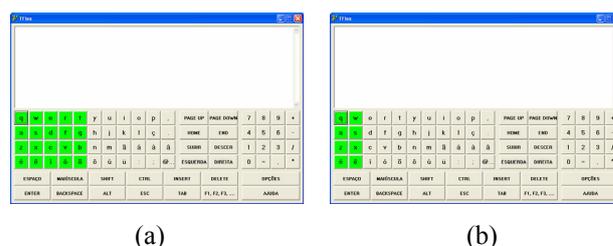
Antes de iniciar a implementação discutiu-se a validade de se utilizar a expansão do grupo de caracteres selecionado, onde a cada iteração da varredura, esse se expandiria para ocupar toda a tela do teclado. Entretanto, decidiu-se que essa não seria uma abordagem interessante pelo fato das teclas ficarem modificando suas posições a cada iteração, obrigando o usuário a procurar pela tecla desejada cada vez que selecionasse um grupo. Mantendo as teclas em suas posições originais, o usuário pode manter o foco de sua visão sempre em um mesmo ponto do teclado, facilitando assim a escolha da tecla desejada e em consequência, agilizando a velocidade de digitação.

Com o intuito de reduzir o campo de busca, optou-se por realizar, como na busca tradicional, uma seleção de um grupo de caracteres de funções parecidas antes de se realizar a varredura binária.

A Figura 10 mostra a seleção do grupo que contém letras, símbolos e pontuação. Após a seleção do grupo, o mesmo é dividido em 2 subgrupos, entre os quais o foco se alterna. A Figura 11(a) mostra a divisão esquerda em destaque. Para que o usuário selecione um grupo, utiliza-se a tecla *Espaço*, como na varredura tradicional. Caso o usuário tenha errado na seleção, ele pode utilizar a tecla *Esc* para voltar para a seleção de grupos, ou ainda esperar que o sistema o faça automaticamente após 3 alternâncias de foco.



Figura 10. Seleção do grupo de letras, símbolos e pontuação.



(a)

(b)

Figura 11. (a) Seleção da seção esquerda da divisão binária na segunda iteração; (b) Terceira iteração.



(a)

(b)

Figura 12. (a) Divisões em metade Inferior e (b) Superior.



Figura 13. Seleção do caractere.

Suponhamos que a tecla a ser digitada seja a letra “a”. O usuário continua selecionando a metade esquerda, como mostrado na Figura 11(b), até chegar a uma única coluna. Nota-se que até esse momento o campo de busca foi dividido sempre verticalmente, em metades esquerda e direita. A partir desse momento, como não se adotou a abordagem de movimentação de teclas por motivos supracitados, o campo de busca será dividido horizontalmente. Ou seja, divide-se a coluna em metades superior e inferior, como mostrado e nas Figuras 12(a) e (b). O usuário continua com as iterações de divisão até chegar ao caractere desejado, como pode ser observado na Figura 13. Em seguida, a busca reinicia no grupo selecionado anteriormente, a fim de que o usuário possa iniciar a busca por um outro caractere, facilitando assim a seleção de teclas de um mesmo grupo, quando na digitação de um texto, por exemplo, caracteres. Esse comportamento tem o intuito de aproveitar o tema da digitação do usuário, facilitando a digitação futura de elementos de mesmo tema dos anteriores. Caso isso não fosse levado em conta, o usuário que acabou de digitar um número, por exemplo, teria de esperar pela varredura passar pelos grupos de caracteres especiais, letras, movimentação até chegar novamente nos números.

### COMPARAÇÃO ENTRE MODELOS

Para analisar a acessibilidade dos dois modelos implementados, primeiro tentamos levantar na literatura *guidelines* que nos possibilitasse fazer uma inspeção de usabilidade, como a avaliação heurística [10], mas com especificidades relativas a características de acessibilidade, que eram as que nos interessava neste primeiro teste.

Atualmente, existem diversos guias de acessibilidade, tanto para software quanto para Internet, havendo uma preponderância maior de guias para a Internet. Dentre estes guias, destacam-se os da WAI/W3C para desenvolvimento de conteúdo para a Web acessível [1] e ferramentas de autoria para Web [13], além das recomendações de acessibilidade da Microsoft [8] para seus sistemas operacionais.

Desenvolver um software acessível a partir de guias como o da WAI/W3C para ferramentas de autoria para a Web pode ser considerada uma tarefa difícil, uma vez que tais guias podem conter informações abstratas, em alto nível, dizendo apenas o que deve ser feito, e não o como, e muitos não levam em conta o contexto de uso do software em questão (se na educação, em um ambiente de trabalho, etc.). Já recomendações como as da Microsoft dependem de ferramentas de desenvolvimento exclusivas da empresa, e são voltadas principalmente para a implementação de software usável com determinadas tecnologias assistivas (leitores de tela, por exemplo). Ainda, estes guias possuem mais dois pontos negativos. Primeiro, nenhum deles considera testes com usuários para se indicar o nível de acessibilidade. Segundo, nenhum guia é aceito universalmente.

Desta forma, avaliar a acessibilidade de um software apenas verificando sua conformidade com tais guias não seria apropriado. E, infelizmente, ainda não existem metodologias formais amplamente difundidas voltadas exclusivamente para a avaliação de acessibilidade.

Optamos, portanto, em iniciar a avaliação dos dois tipos de varredura implementados a partir de testes de usabilidade nos quais a tarefa consistia na digitação de um texto simples: “Parreira levará a campo os mesmos jogadores que golearam a Hungria por 4 a 1.”. Para isso nos baseamos no layout QWERTY, de modo a tomar proveito da experiência que os usuários possuíam nesse layout.

Para a realização dos testes utilizamos 5 usuários de diversas áreas do conhecimento, sendo os mesmos não portadores de qualquer deficiência. Todo o grupo já possuía certa experiência com o uso de computadores, e nenhum havia utilizado simuladores de teclados, bem como não foi realizado qualquer treinamento com os usuários antes da aplicação dos testes. Mantivemos o intervalo de chaveamento de grupos em 0,8 s, de modo a adequar a uma ampla gama de usuários.

O usuário 2 somente testou o TFlex com busca binária, enquanto que o usuário 3 somente utilizou a busca tradicional. O intuito desse tipo de teste foi a de se eliminar a influência do aprendizado nos resultados do teste, uma vez que o usuário poderia aproveitar a experiência adquirida da utilização de um teclado bem como relembrar a disposição das teclas para agilizar a digitação.

Nos outros casos, os usuários testaram os dois modos de varredura, sendo que para os usuários 1 e 4, iniciou-se com a busca tradicional sendo em seguida utilizada a binária. No caso do usuário 5 aplicou-se o teste na ordem inversa. Há uma ressalva quanto ao usuário 5, já que ele faz parte da equipe de desenvolvimento do TFlex, entretanto o resultado desse teste foi interessante pois pudemos analisar os efeitos que a experiência traz na digitação. O tempo e o número de toques que cada usuário utilizou na digitação do texto nos testes estão organizados nas Tabelas 1 e 2.

	Tipo de Busca (Minutos)	
	Tradicional	Binária
Usuário 1	14	23
Usuário 2	-	18
Usuário 3	15	-
Usuário 4	11	17
Usuário 5*	10	12

Tabela 1. Tempo (em minutos) para digitação de um texto elementar.

	Tipo de Busca (Número de toques)	
	Tradicional	Binária
Usuário 1	269	597
Usuário 2	-	824
Usuário 3	Não computado	-
Usuário 4	242	656
Usuário 5*	250	555

Tabela 2. Toques necessários na digitação de um texto elementar.

Nota-se que o aprendizado, como esperado, tem uma influência direta na eficiência de digitação. Observa-se que o usuário 1 utilizou menos toques que o usuário 2 devido a sua experiência prévia com a busca tradicional. Nota-se que o usuário 5 obteve o menor número de toques na busca binária devido a sua ampla experiência com esse tipo de varredura, assim como o tempo utilizado para a digitação foi o menor de todos. Por outro lado, o usuário 3, por sua falta de experiência com a busca tradicional utilizou o maior tempo para digitar o texto.

Um aspecto a ser observado é que a experiência influencia significativamente na eficiência de uso, sendo assim, é de se esperar que o usuário necessite de um tempo de aprendizado. Futuros testes serão conduzidos no sentido de se avaliar esta curva de aprendizagem com relação aos dois tipos de varredura. Principalmente, com relação ao da varredura binária onde a princípio a eficiência pretendida somente será alcançada quando o usuário entender este algoritmo de busca. E isso de alguma forma deverá ser possibilitado por características da interface que salientem os aspectos envolvidos no processo de busca, principalmente com relação a subdivisão e alternância entre grupos.

Através desses testes, utilizamos as opiniões dos usuários para aprimorar o TFlex. Uma das características já alteradas no protótipo foi com relação ao modo de varredura binária. Neste modo de varredura ao pressionar a tecla *Esc* o TFlex retornava a varredura para a seleção de grupos, diferentemente da varredura tradicional, onde a varredura retorna para o modo anterior. Tomemos o

exemplo da varredura das colunas de uma linha para procurar um determinado caractere, mas o caractere se encontra na linha anterior. Ao se pressionar *Esc* a busca retoma a varredura por linhas, fazendo com que o usuário possa se recuperar rapidamente do erro. Esse comportamento ocasiona na busca binária um número superior de iterações no caso do usuário cometer um erro de seleção na varredura.

Outros pontos importantes e que estão sendo analisados dizem respeito à divisão dos grupos e o posicionamento de teclas com elevado uso, como a tecla de ESPAÇO, por exemplo, que devido a sua alta frequência de uso possivelmente deverá ser posicionada junto ao grupo de caracteres e prevê-se a necessidade de incluí-la no teclado com alguma redundância.

Um dos resultados mais efetivos desses testes iniciais foi que a consistência espacial deve ser garantida como uma das características básicas de usabilidade. Por exemplo, o erro mais comum dos usuários no teste foi com relação às teclas acentuadas que estão divididas em mais de uma linha. E sempre a seleção era feita na linha que iniciava um caractere, dada a característica da varredura tradicional ser por linhas. Assim o usuário iniciava a busca de um caractere em uma determinada linha, ainda sem levar em consideração o acento desejado, e quando direcionava sua atenção para o acento do caractere, percebia que o mesmo se encontrava em outra linha.

E finalmente, pode-se concluir que aumentando-se a amplitude dos testes a partir de um melhor refinamento do protótipo, teremos coletado uma série de resultados que esperamos poder compor na forma de um primeiro conjunto de *guidelines* a serem consideradas no design de produtos de software com funcionalidades e objetivos análogos aos do TFlex.

## CONCLUSÕES

As tecnologias assistivas são importantes dispositivos capazes de ajudar no acesso aos recursos computacionais pelas pessoas portadoras de deficiência. Uma destas tecnologias é conhecida por simulador de teclado, um teclado virtual mostrado na tela do computador que possui um sistema que varre as teclas.

Os modelos atuais de simuladores de teclado possuem certas restrições que dificultam a sua adoção por usuários portadores de deficiência motora ou física. Por estas razões, o TFlex foi proposto e está em processo de desenvolvimento, podendo ser obtido gratuitamente em [12].

Apesar de testes com os primeiros protótipos do TFlex terem sido feitos apenas com usuários não portadores de deficiência, alguns aspectos foram notados, como a necessidade de se rever a disposição de teclas específicas (Enter, Backspace, Espaço, etc.) e de se configurar o tempo da varredura de acordo com a habilidade do usuário. Utilizamos voluntários não portadores de necessidades especiais em um primeiro momento com o objetivo de descobrir os principais problemas de usabilidade e refinar o protótipo.

Há de se considerar, também, que os usuários que realizaram os testes com o TFlex estavam acostumados com um modelo direto de entrada de dados no computador, através do teclado físico tradicional. Provavelmente, isto influenciou nos testes e a sensação generalizada de pouca eficiência em escrever com um simulador de teclado. Salienta-se, mais uma vez, que o objetivo principal do TFlex (e de outros simuladores de teclado) é a de se permitir a escrita por aqueles que não podem usar nem o teclado nem o mouse. Ainda assim, o TFlex possui abordagens que tentam melhorar a eficiência, como a proposta de varredura baseada na busca binária e os layouts alfabético e por frequência de uso. Além disso, concluímos que algumas modificações simples podem ser realizadas na busca binária de modo a melhorar a forma de recuperação em caso de erro de digitação.

Planeja-se, com o desenvolvimento de um protótipo mais refinado, testar o TFlex com usuários portadores de deficiência a fim de se comparar de forma mais realista tanto os modos de varredura (tradicional e binária) quanto os layouts de teclado propostos (QWERTY, alfabético e por frequência de uso).

Nota-se ainda que os refinamentos posteriores do TFlex devem levar em consideração a relação de compromisso entre manter um layout de teclado que seja eficiente, mas ao mesmo tempo, mantenha o aspecto de um teclado físico. Além disso, deve-se levar em consideração o tamanho das teclas do TFlex de forma a facilitar a sua leitura, bem como manter um tamanho de janela suficientemente pequeno para não atrapalhar a visualização de outras janelas de programas que por ventura sejam utilizadas junto com o TFlex.

Por fim, espera-se que este projeto resulte em um software de tecnologia assistiva que sirva como uma alternativa às pessoas que não podem utilizar os periféricos tradicionais de entrada de dados no computador.

#### AGRADECIMENTOS

Primeiramente gostaríamos de agradecer a professora Heloísa V. da Rocha do Instituto de Computação da UNICAMP pelo suporte e apoio ao desenvolvimento desse projeto, bem como no auxílio e dedicação aos testes realizados. Não podemos deixar de agradecer a todos os

voluntários que se prontificaram a testar o TFlex, como também pelas preciosas críticas e determinação dos pontos problemáticos do projeto.

#### REFERÊNCIAS

- 1.Chisholm, W.; Vanderheiden, G. Jacobs, I. Web Content Accessibility Guidelines 1.0, maio/1999. <http://www.w3.org/TR/WCAG10/>.
- 2.Hartman, J. How the Typewriter Got Its Keys. <http://www.kith.org/logos/words/upper/Q.html>
- 3.Hogetop, L; Santarosa, L. M. C. Tecnologias Assistivas/Adaptativas: viabilizando a acessibilidade ao potencial individual. set/2001. <http://www.nied.unicamp.br/~proinesp>.
- 4.IBGE. Censo Demográfico 2000 - Tabulação Avançada. População residente, por situação do domicílio e sexo, segundo o tipo de deficiência, Brasília/DF, 2003. <http://www.ibge.gov.br/estatistica/populacao/censo2000>
- 5.Inclusive Technology. Access Devices. <http://www.inclusive.co.uk/catalog/hardware.shtml>.
- 6.Lakefolks. Click-N-Type Virtual Keyboard. <http://www.lakefolks.org/cnt/>.
- 7.Mackay, D. The Dasher Project. <http://www.inference.phy.cam.ac.uk/dasher/>.
- 8.Microsoft. MSDN Library - User Interface Design and Development – Accessibility. <http://msdn.microsoft.com/at>.
- 9.NIEE. Simulador de Teclado. <http://www.niee.ufrgs.br/software/soft1.html>.
- 10.Nielsen, J. Usability Engineering. Academic press Limited, 1993.
- 11.SACI. O Projeto Teclado Amigo. <http://www.saci.org.br/pub/kitsaci2/projeto.html>.
- 12.TFlex. Projeto TFlex. <http://www.nied.unicamp.br/~tflex/>
- 13.Treviranus, J; McCathieNevile, C; Jacobs, I; Richards, J. Authoring Tool Accessibility Guidelines 1.0 – W3C Recommendation, fev/2000. <http://www.w3.org/TR/ATAG10/>.
- 14.WiViK. What's WiViK?. <http://www.wivik.com/Description/description.html>.

