

Komunikacja człowiek-komputer

Wykład 10

Dr inż. Michał Kruk

GUI

- Interfejsy użytkownika
 - 1963 – sketchpad
 - 1964 – Pierwsza mysz - Stanford
 - 1973 – pierwszy komputer z graficznym interfejsem - Alto Xerox Parc
 - 1983 – komputer Lisa firmy Apple – Komputer posiadał okienkowe GUI, Uznane jako pierwowzór tego rodzaju interfejsów

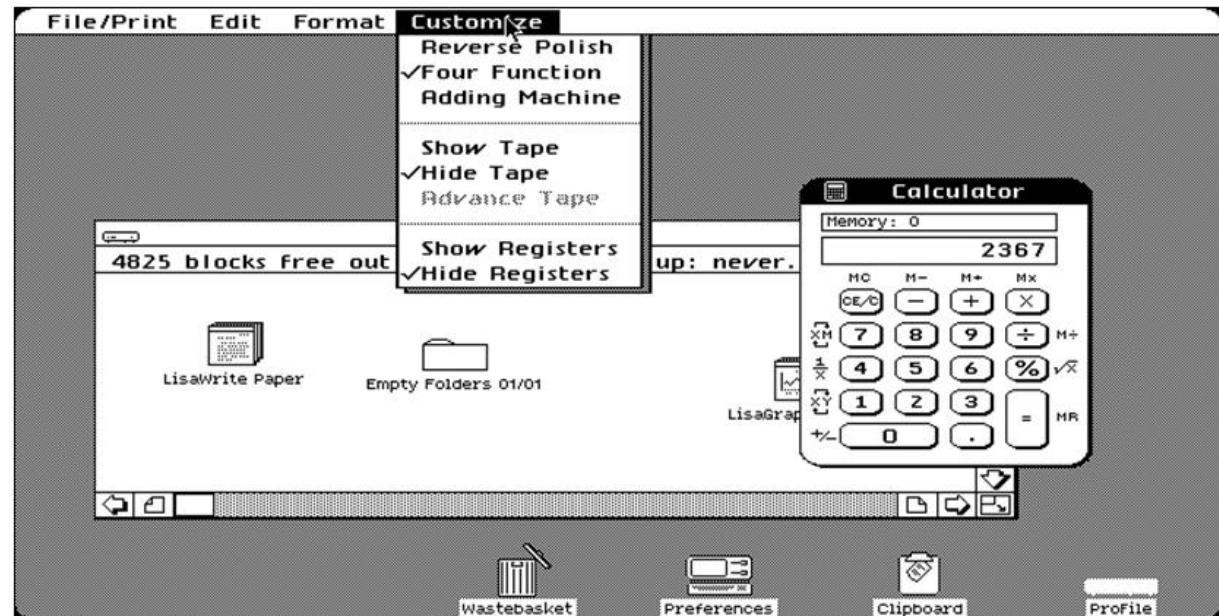
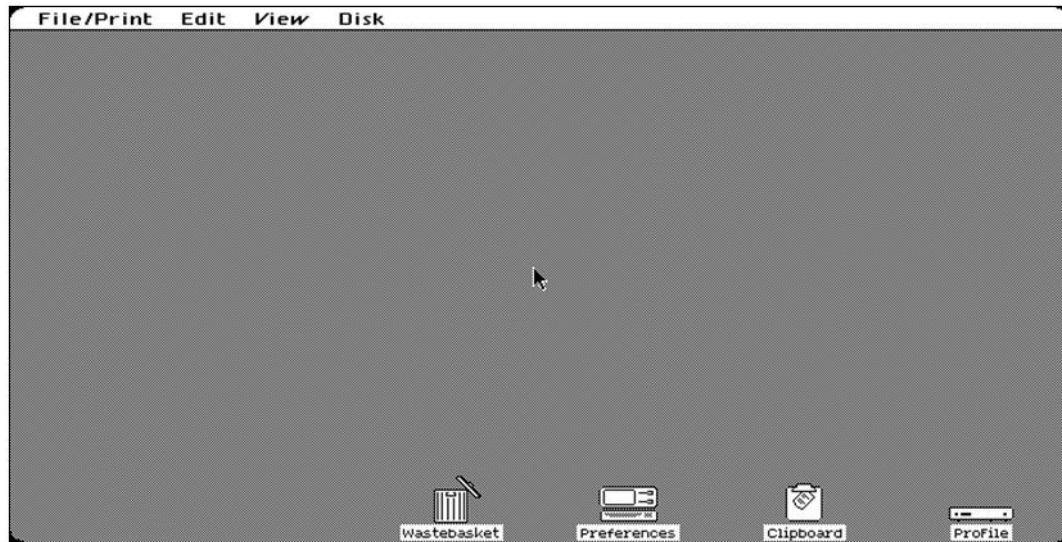


GUI

- Interfejsy użytkownika c.d.
 - Koszt Lisy ok. 10 000\$
 - System Lisa OS
 - Kalkulator
 - Edytor tekstów
 - Kosz (!)
 - Menedżer plików
 - Program do wykresów



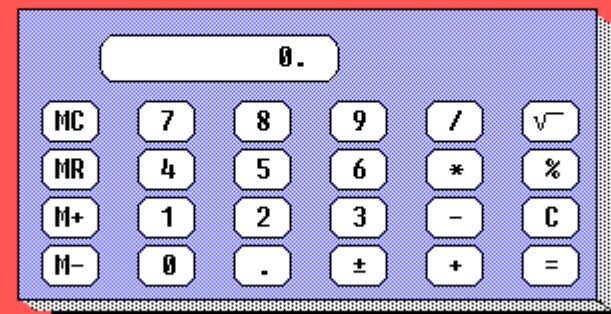
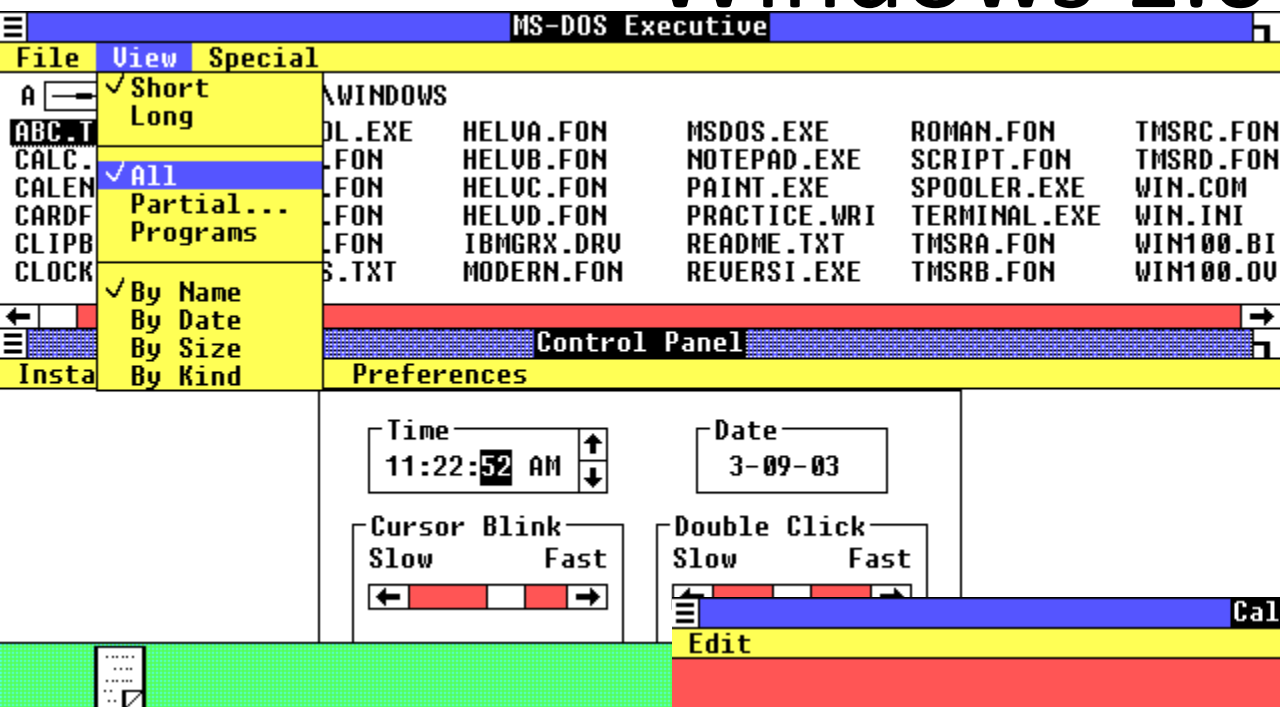
GUI



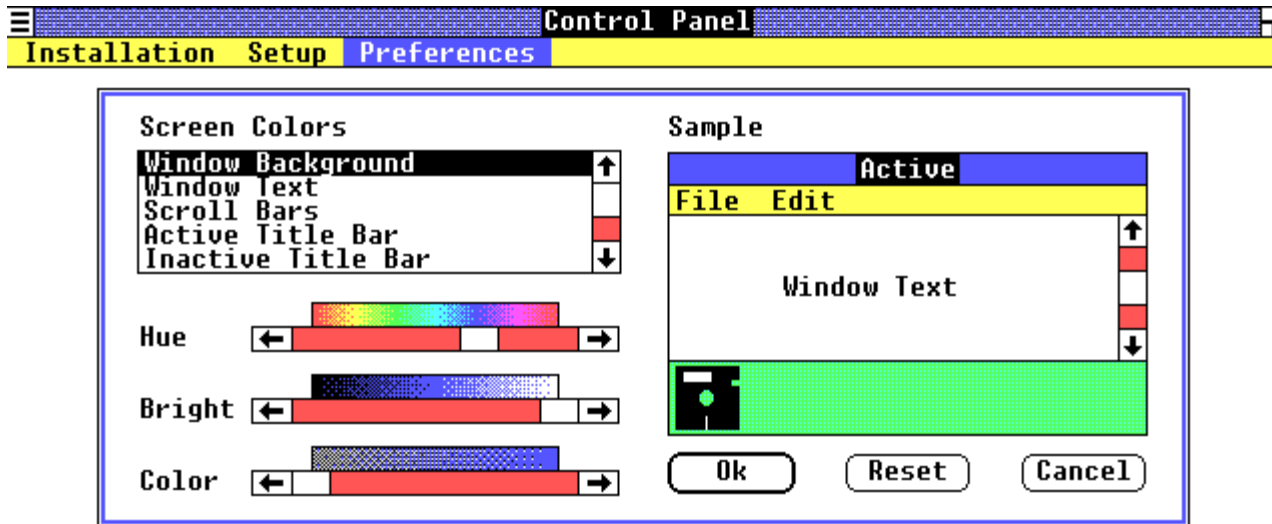
GUI

- 1984 – Macintosh
 - Nakładające się okienka
- 1984 – System X Window
 - Całkowicie przenośny system okienkowy
- 1985 – Windows 1.0
 - Brak możliwości przenoszenia i nakładania się okienek
- www.guidebookgallery.org

Windows 1.0



Windows 1.0



Kategorie UI - Różnorodność

Internet:

- strony domowe,
- sklepy internetowe,
- serwisy korporacyjne, intranet,
- serwisy bankowe,
- serwisy informacyjne,
- wyszukiwarki,
- blogi.

Aplikacje

- biurowe,
- inżynierskie / naukowe,
- grafika komputerowa,
- programiści,
- przemysłowe (magazyny, sterowanie itp.),
- systemy ogólnego przeznaczenia.

Urządzenia:

- multimedialne,
- pda / telefony komórkowe,
- bankomaty.

Ograniczenia Techniczne

Aplikacje internetowe:

Protokół bezstanowy (tzw. cienki klient),
bezpieczeństwo, ograniczone
możliwości dostępu do danych,
problem z nawigacją na stronie
(wygodne do formularzy, ale
trudne wypełnianie danych w
tabelach, dwuklik myszką)

Ograniczenia Techniczne

Aplikacje biurowe: prostota obsługi,
szablony, automatyzacja zadań
biurowych, skróty klawiaturowe,
ergonomia interfejsu,
dostosowanie do preferencji
użytkownika

Ograniczenia Techniczne

Urządzenia: udostępniająca tylko niezbędne funkcje, brak myszki, brak klawiatury, niestandardowe klawisze – brak możliwości dostosowania interfejsu przez użytkownika

Ograniczenia Techniczne

Systemy inżynierskie: duże możliwości,
dopuszczalny skomplikowany
interfejs.

Systemy przemysłowe: ograniczona i
bardzo specjalistyczna
funkcjonalność, bezawaryjność,
obserwowalność: czyli widoczność
wszelkich usterek i awarii,

Postrzeganie

Interfejs wpływa na odbiór programu/strony przez użytkowników.

- Użyteczne oprogramowanie lepiej się sprzedaje.
- Nieużyteczne strony są szybko porzucane.
- Do estetycznego / użytecznego programu z większą przyjemnością się wraca (np. z *przerwy na kawę* :-).

Interfejs jest często niedoceniany, traktowany powierzchownie, bezkrytycznie.

- Użytkownicy często obwiniają siebie za błędy w projekcie interfejsu.
- Osoby decydujące o zakupie/projekcie systemu zazwyczaj nie są końcowymi użytkownikami, dlatego często nie dbają o ważne szczegóły.

Koszty złego projektu

- Koszt czasu użytkownika wcale się nie zmniejsza zgodnie z prawem Moore'a. (natomiast: "moc obliczeniowa komputerów podwaja się co 18 miesięcy",
czyli należy automatyzować jak najwięcej czynności)
- Należy poświęcić więcej czasu na projekt, bo zapłaci się
za niedokładność później.
- Błędny interfejs powoduje wypadki!

Interfejs Użytkownika

- Nie jesteś użytkownikiem.
 - Większa część inżynierii oprogramowania zajmuje się komunikacją między programistami (specyfikacje, modele obiektowe, biblioteki, etc.).
 - UI (Interfejs Użytkownika) zajmuje się komunikacją z użytkownikiem.
- Użytkownik ma zawsze rację...
 - Jeśli użytkownik ciągle popełnia błędy podczas używania programu, to najprawdopodobniej winny jest projekt.
 - ... ale niestety użytkownik czasem nie ma racji, ponieważ ma zbyt małe doświadczenie i wiedzę! Nie traktujmy każdego użytkownika jak 'eksperta'.

Użyteczność jest tylko częścią procesu

Projektanci oprogramowania muszą zajmować się wieloma aspektami systemu:

- Funkcjonalność,
- Wydajność,
- Koszt,
- Bezpieczeństwo,
- **Użyteczność,**
- Rozmiar,
- Wiarygodność,
- Standardy.

Wiele decyzji projektowy stanowi kompromis pomiędzy tymi aspektami.

Podczas kursu przyjmimy krańcową postawę: użyteczność będzie naszym podstawowym celem.

Interfejs Użytkownika Trudno Zrealizować

Interfejs użytkownika pochłania dużą część wysiłku związanego z implementacją.

Szacuje się, że wytworzenie UI stanowi 50%:

- czasu poświęconego na projekt
- czasu na implementację
- kosztów / czasu utrzymania systemu
- rozmiaru całkowitego kodu systemu

Redukcja nakładów tworzenia UI:

- Komponenty (kontrolki, databinding, itp.)
- Ponowne wykorzystanie (widoki obiektów, itp.)
- Maksymalnie uniwersalne komponenty (DBGrid)
- Automatyzacja generowania interfejsów (MS Access, Delphi ECO)
- DOBRY PROJEKT

Interfejs Użytkownika – Czynniki Sukcesu

Funkcjonalność

- ergonomia, prostota obsługi, czytelność prezentowanej informacji, realizacja określonych zadań itp.

Odporność na błędy

- stabilność, powtarzalność, konsekwencja, zgodność...

Estetyka

- odbiór aplikacji przez użytkowników, czy danej aplikacji używa się z przyjemnością i chętnie do niej wraca,
- odpowiedni dobór kolorów i wyróżnień tak aby nie utrudniały odbioru przedstawionej informacji.

Dostępność

- czy nasza aplikacja może być wykorzystywana przez osoby z ograniczonymi możliwościami fizycznymi.

Użyteczność

(ang. usability, web-usability) - nauka zajmująca się ergonomią interaktywnych urządzeń oraz aplikacji. W Polsce pojęcie użyteczności stosowane jest zazwyczaj w odniesieniu do ergonomii serwisów WWW oraz aplikacji użytkowych.

Użyteczność w ich przypadku skupia się na:

- intuicyjnej nawigacji,
- ułatwieniu skanowania w poszukiwaniu informacji,
- zapewnieniu zrozumiałej dla użytkownika komunikacji,
- udostępnieniu odpowiedniej funkcjonalności.

Problemy z terminologią - „funkcjonalność”, „użyteczność” czy może

„używalność”

Mierniki użyteczności

Użyteczność stara się formalizować ocenę funkcjonalności wprowadzając mierniki użyteczności:

1. Skuteczność: czy użytkownik może osiągnąć cel?
2. Nauka obsługi: czy łatwo się jej nauczyć?
3. Ergonomia: po nauczaniu się, czy szybko się używa?
4. „Pamiętalność” - czy łatwo przypominamy sobie to czego się nauczyliśmy?
5. Błędy: czy błędów użytkownika jest niewiele i są odwracalne?
6. Satysfakcja: czy system używa się z przyjemnością?

Mierniki zależą od odbiorcy

Wagi mierników zależą od aktualnego użytkownika:

- początkujący potrzebuje nauki obsługi
- sporadyczni użytkownicy „pamiętalności”
- zaawansowani ergonomii

Niestety żadnego użytkownika nie da się ściśle przydzielić do jednej z kategorii:

- mamy do czynienia z ekspertami z dziedziny w ramach której działa aplikacja, którzy są początkujący w przypadku naszej aplikacji,
- niektóre elementy / właściwości systemu mogą być rzadko wykorzystywane.

Użyteczność to 'filozofia'

Użyteczność z jednej strony jest nauką, a z drugiej filozofią zgodnie z

którą czas/satysfakcja użytkownika są postawione na jednym z

najważniejszych miejsc.

W ramach użyteczności:

- stosujemy się do wytycznych na wszystkich etapach wytwarzania produktu
- szanujemy zdanie użytkownika
- wykonujemy testy
- tworzymy system zgodnie z pewnym procesem

Użyteczność – czym **NIE** jest?

- Testowaniem użyteczności tuż przed wdrożeniem.
- Prostym stosowaniem zaleceń projektowych w fazie projektu.
- Prowadzeniem ewaluacji bez stosowanych zaleceń dotyczącym procesu.
- Własną wypracowaną metodą opartą na doświadczeniu lub heurystykach.
- Nie jest tylko kosmetyką, estetyką projektu.
- Własnością, która zależy tylko od produktu.

Co powinien uwzględniać projekt UI?

Zwiększenie prędkości przyswajania (uczenia się obsługi).

- Miernik: czas potrzebny użytkownikowi do osiągnięcia określonego poziomu zaawansowania? Aspekt ten jest najbardziej istotny w przypadku aplikacji używanych sporadycznie.

Zwiększenie prędkości używania (ergonomia).

- Jak dużo czasu zajmuje zaawansowanemu użytkownikowi wykonanie określonego zadania? Systemy intensywnie wykorzystywane. (Przykład CRM, Kasjer na Poczcie, itp.)

Minimalizacja popełniania błędów.

- Jak dużo błędów popełnia przeciętny użytkownicy podczas typowej sesji z systemem?

Wspomaganie szybkiego przypominania.

- Jak dużo czasu zajmuje sporadycznemu użytkownikowi przypomnienie sobie obsługi?

Podniesienie poziomu atrakcyjności.

- Jaka część użytkowników odbiera system pozytywnie? Ilu osobom podoba się system wizualnie?

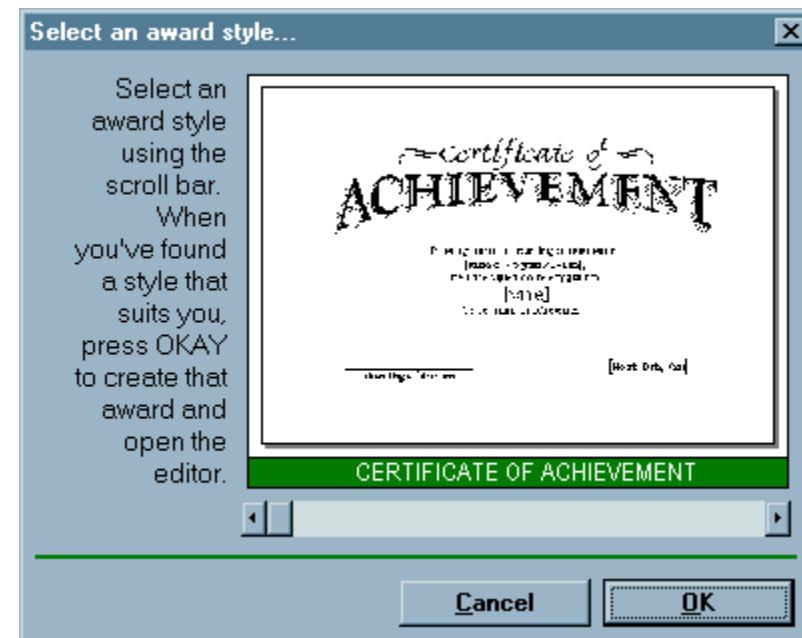
Przykłady

Cel:

- wprowadzenie w problematykę
- wytworzenie wrażliwości na typowe błędy i niekonsekwencje
- przedstawienie błędów w komercyjnych produktach

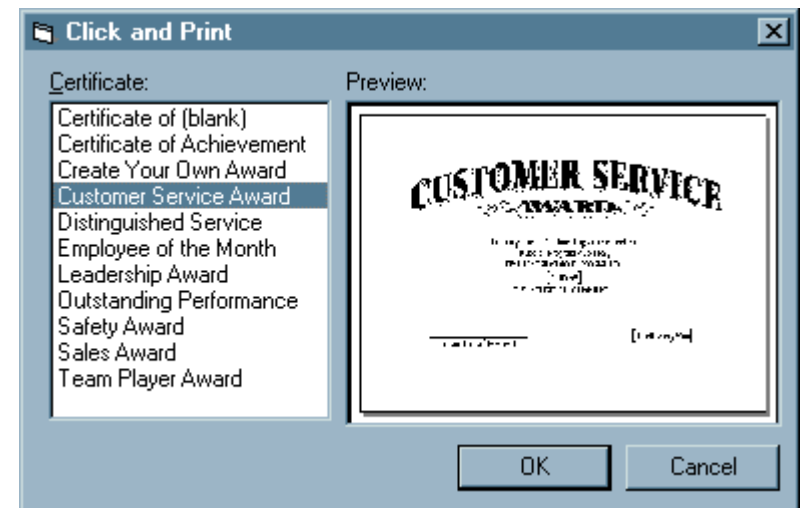
Interfejs Użytkownika - Galeria Niestawy

- W pełni graficzny.
- Myszka.
- WYSIWYG.
- Długi komunikat pomocy.
- Dziwaczny interfejs.
- Scrollbar - Suwak



Poprawiony Interfejs

Lista wyboru (Listbox)
udostępnia zmianę stylu dla
sporadycznych jak i stałych
użytkowników.
Swobodny wybór dla stałych
użytkowników.
Niepotrzebny jest żaden
komunikat pomocy.



Więcej Galerii Niestawy

First Launch Date:	<input type="text" value="09/09/97"/>	<input type="button" value="Set Date"/>
First Launch Time:	<input type="text" value="19:17"/>	<input type="button" value="Set Time"/>

7:17 pm

am 11 12 1 pm

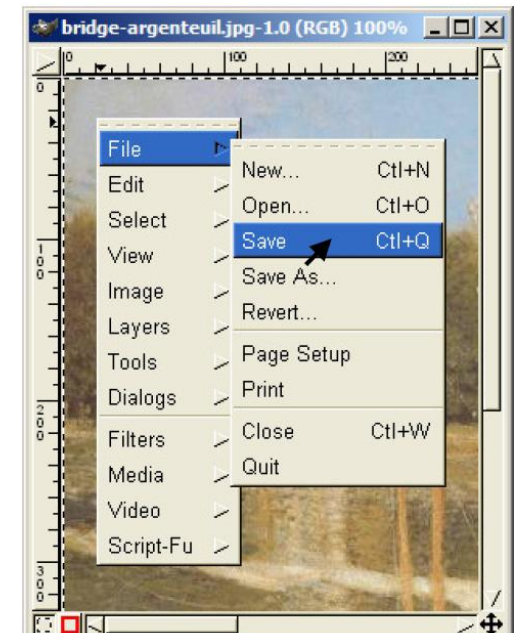
10 9 8 7 6 5 4 3 2

Left Mouse Button: Change Minute
Right Mouse Button: Change Hour

Left Mouse Button: Change Minute
Right Mouse Button: Change Hour

Więcej Galerii Niestawy

- Brak menu.
- Dynamiczne klawisze Skrótów. Czy użytkownik zawsze potrzebuje
- dostosowywać aplikację? Gdzie jest granica rozsądku?



Przykład

- SimCity 2000 firmy Maxis. Przyciski, które mają podmenu dostępne po długim wciśnięciu przycisku myszki.

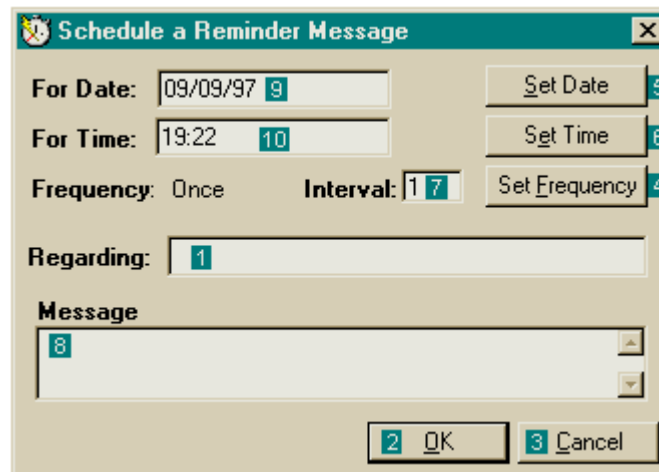


Przykład

Nawigacja za pomocą klawisza Tab.

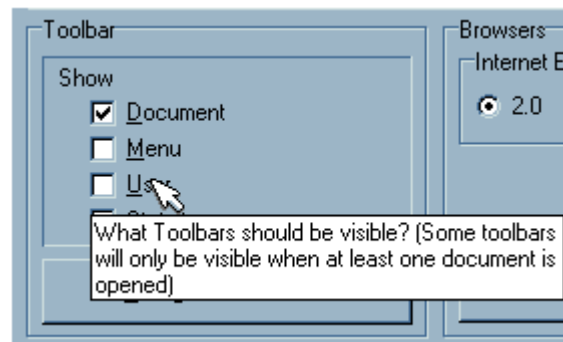
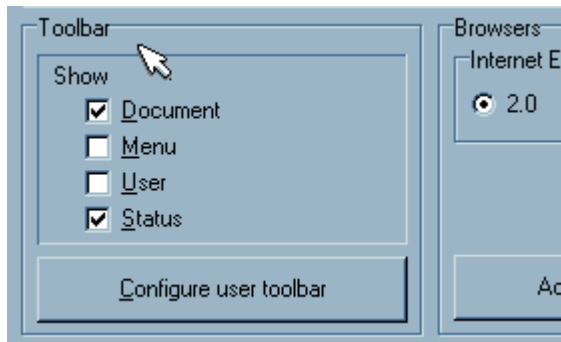
– (Unisyn's Automate Pro – program do automatyzacji zadań dla

Windows NT.)



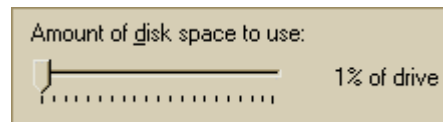
Przykład

- Pomoc kontekstowa jest bardzo potrzebna, ale przesada, a zwłaszcza długie komunikaty mogą doprowadzić do frustracji.



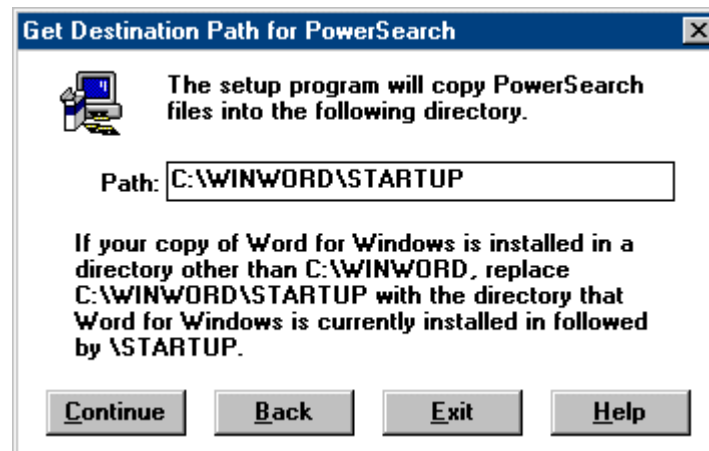
Przykład

- Wybór kontrolki jak najbardziej właściwy, ale jednostki pozostawiają wiele do życzenia.
(Internet Explorer 3.0 – Rozmiar Cache)



Przykład

Instalacja programu PowerSearch. Nie ma problemu, gdy użytkownik wybierze katalog domyślny? A co gdy będzie chciał zainstalować gdzie indziej? A tak w ogóle, czy nie dałoby się sprawdzić gdzie jest zainstalowany Word?



Przykład

- Użytkownicy zdecydowanie szybciej przeglądają listy w kierunku od góry do dołu (heurystyka). Tutaj Microsoft jednak zmusił ich do przeglądania dodatkowo od lewej do prawej. (Internet Explorer 3.0)



Przykład

- Program: HTML Transit
- Poświęćcie 25% powierzchni okna na przyciski komend?
- – Staraj się nigdy nie mieć dwóch przycisków o takim samym znaczeniu.

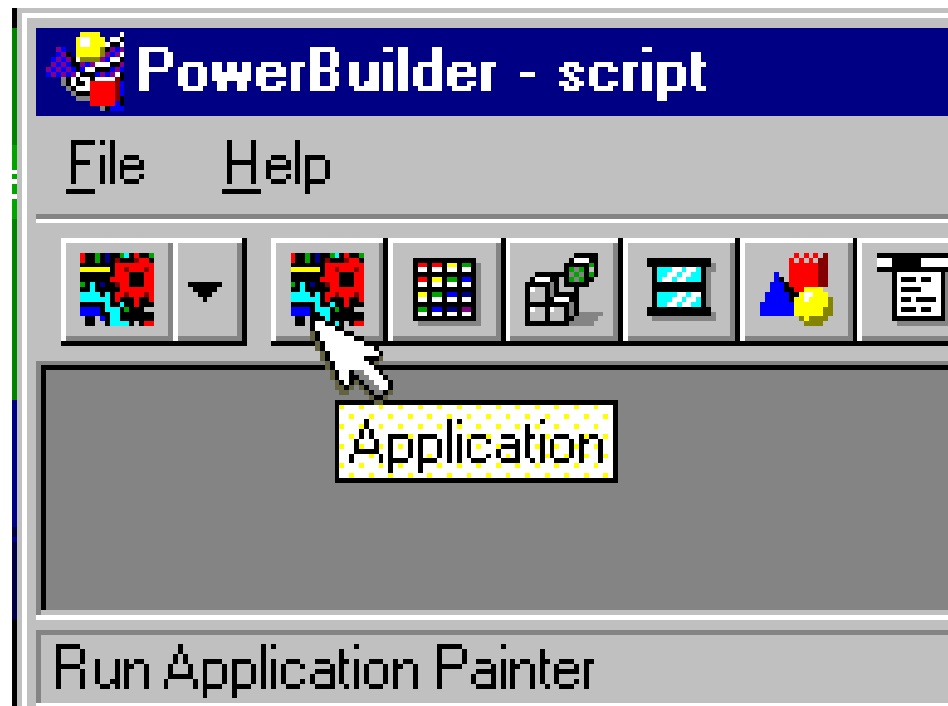
Link to	Icon path		
<input type="checkbox"/> <u>P</u> revious page		Browse...	Gallery...
<input type="checkbox"/> <u>N</u> ext page		Browse...	Gallery...
<input type="checkbox"/> <u>T</u> op of page		Browse...	Gallery...
<input type="checkbox"/> <u>I</u> OC		Browse...	Gallery...
<input type="checkbox"/> <u>I</u> ndex		Browse...	Gallery...
<input type="checkbox"/> <u>P</u> revious item		Browse...	Gallery...
<input type="checkbox"/> <u>N</u> ext item		Browse...	Gallery...
<input type="checkbox"/> <u>S</u> pecific page		Browse...	Gallery...
<u>F</u> ile name:		Browse...	

Przykład

PowerBuilder – udostępniał pomoc kontekstową i pomoc w Pasku Statusu.

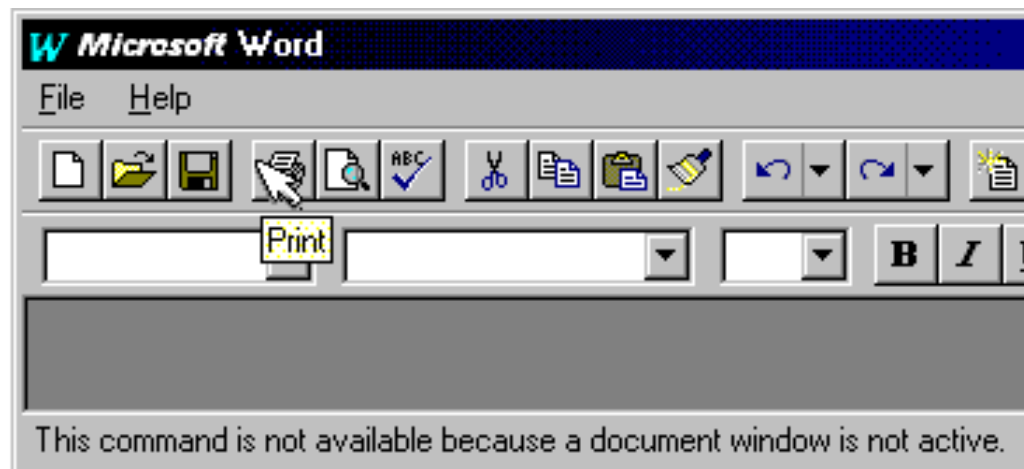
- Problem: pomoc kontekstowa była pokazywana z opóźnieniem ok 2s, ale również pomoc w pasku status była pokazywana z opóźnieniem.
- Skutek: po zmianie pozycji myszy, w pasku statusu była nieaktualna informacja.

Przykład



Przykład

- Przyciski powinny być nieaktywne!
- Możemy być pewni, że programiści Microsoft potrafią dynamicznie zmieniać kolory ikon. Czy zatem wpływ na wyeliminowanie 'szarych', nieaktywnych ikon mieli graficy Microsoft?



Przykład

- Microsoft Excel – Wytnij i Wklej
- Po wciśnięciu wytnij, fragment pozostaje nienaruszony do momentu aż go nie wkleimy w nowym miejscu.

Region	January	February
North	10111	13400
South	22100	24050
East	13270	15670
West	10800	21500

Przykład

- Okna z suwakami! Horror Accessa?

A screenshot of a Windows-style application window titled "Customer Information". The window contains several input fields for customer data: "First Name:", "Last Name:", "Middle", "Address1:", "Address2:", "City:", "State:", "Zip", "Account Number:", and "Mother's Maiden Name:". The fields are arranged in a grid-like fashion. A vertical scrollbar is visible on the right side of the window, indicating that the content is scrollable.

A screenshot of a Windows-style application window titled "Human Resource Administration". The window displays employee information: "Name: SMITH, JOHN Q", "Status:", and "Payroll: G". Below this, there is a scrollable area containing a table with the following data:

Mgr ID	Date	Amount

The table is part of a section labeled "Awards". A vertical scrollbar is visible on the right side of the scrollable area, indicating that the table content is scrollable.

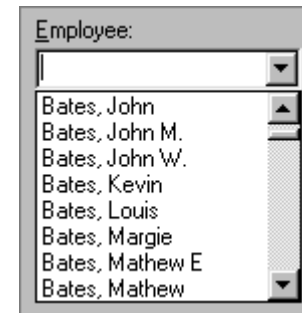
Przykład

- Czyba ktoś nie przeczytał o kontrolce listy wyboru? (Listbox?)



Przykład

Znajdź użytkownika w rozwijalnej liście zawierającej kilka tysięcy nazwisk?



- Są lepsze sposoby walidacji, czy użytkownik wpisał rzeczywiście cyfry! (np.: *InputMask*)



Przykład

- MS Word: Wykluczające się opcje. Dlaczego nie zostały użyte komponenty RadioButton. Należy zaznaczyć, że bardzo nam to nie przeszkadza i szybko jesteśmy się w stanie tego nauczyć, ale jest to jednak niespójność.

Model:

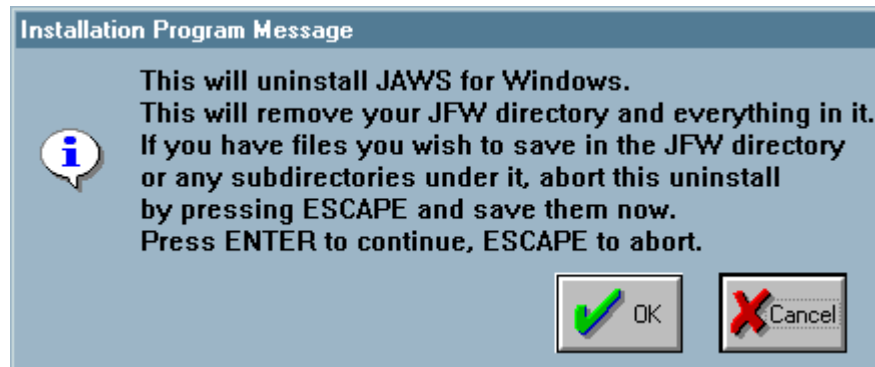
<input type="checkbox"/>	Sport Coupe
<input checked="" type="checkbox"/>	Sedan
<input type="checkbox"/>	Sport Utility
<input type="checkbox"/>	Pickup Truck

Effects

<input type="checkbox"/> Strikethrough	<input type="checkbox"/> Shadow	<input checked="" type="checkbox"/> Small caps
<input checked="" type="checkbox"/> Double strikethrough	<input type="checkbox"/> Outline	<input type="checkbox"/> All caps
<input type="checkbox"/> Superscript	<input type="checkbox"/> Emboss	<input type="checkbox"/> Hidden
<input checked="" type="checkbox"/> Subscript	<input checked="" type="checkbox"/> Engrave	

Przykład

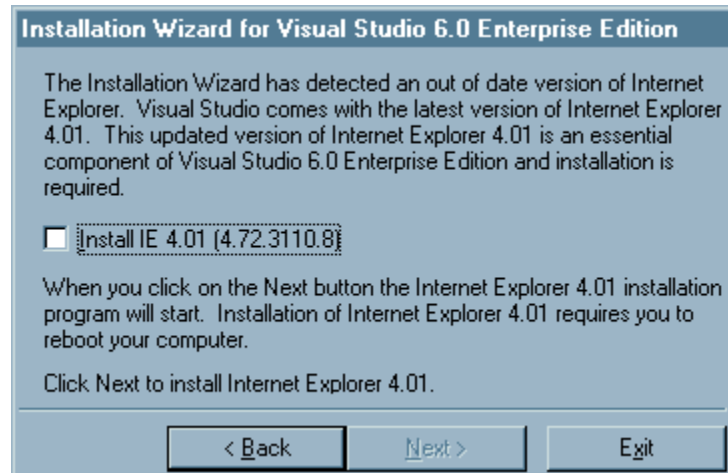
- ENTER? ESCAPE? A w oknie są przyciski OK i Cancel?



- MacOS rozwiązał to inaczej: często klawisze nie są obsługiwane. Tylko kliknięcie myszką zamyka okno.

Przykład

- Internet Explorer, Visual Studio.
- Czy kontrolka Checkbox jest niezbędna?
Przecież i tak nie mamy żadnego wyboru?



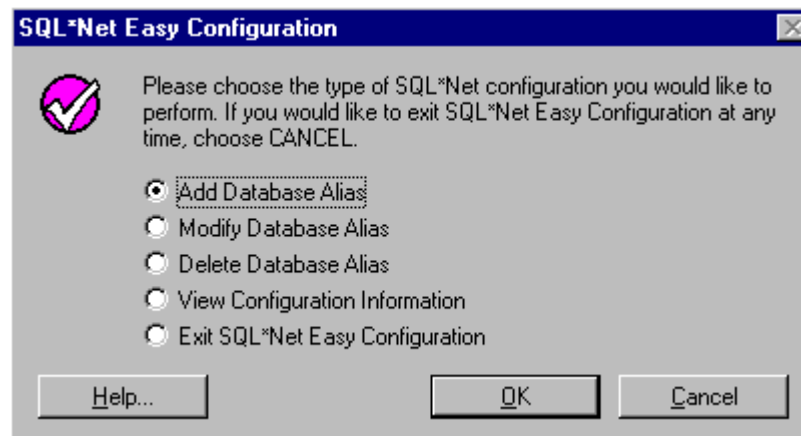
Przykład

- Quiz: Ilu użytkowników wciśnie Enter? Ilu Next? A ilu Finish?



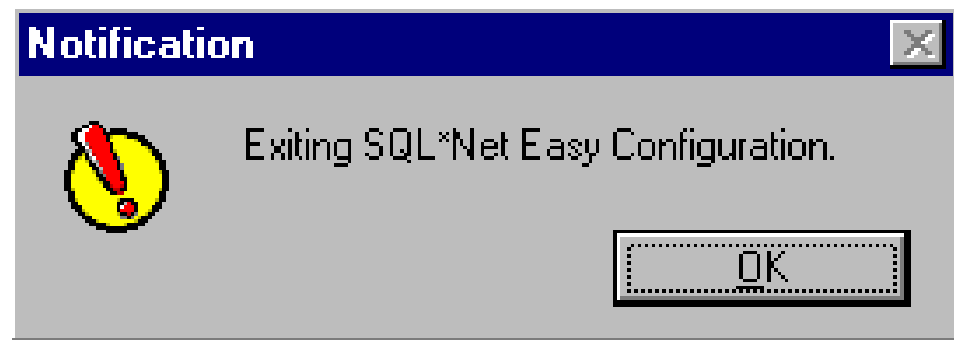
Przykład

- Oracle's SQL*Net Easy Configuration
- Quiz 1: Jak wyjść z aplikacji?



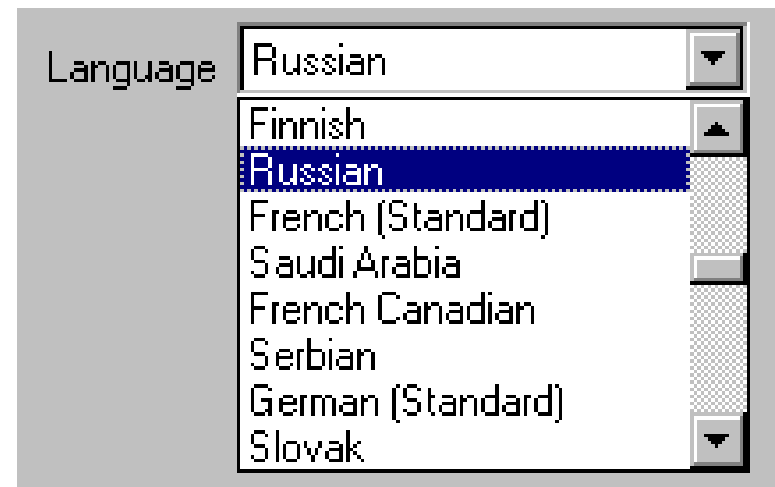
Przykład

- Quiz 2: Po co ten komunikat
- jeśli nie można anulować
- decyzji?



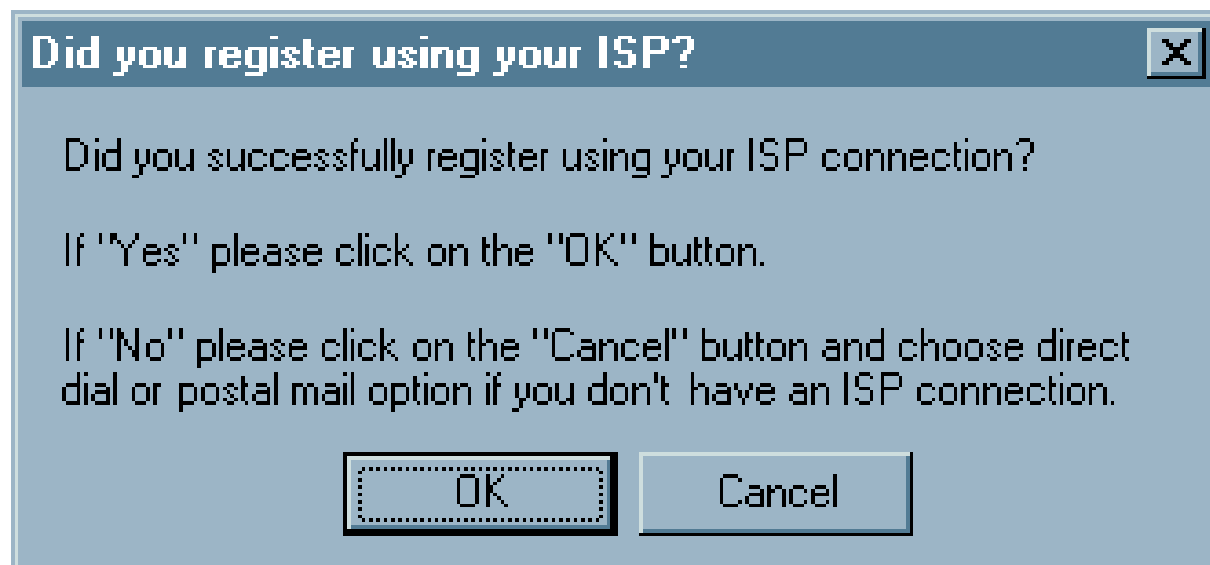
Przykład

- Windows Help Designer.
- Programista zapomniał
- zaznaczyć opcji
- automatycznego
- sortowania.



Przykład

- Instalacja modemu
- U.S. Robotics:



Przykład

- Kontynuuj używanie, czy
- proces zamykania?
- Czym różni się Abort od
- Cancel?



Przykład

- Xblock – gra.
- Kto chciałby zająć
- miejsce numer 0?



A screenshot of a 'High Scores' window from a game. The window has a blue title bar with the text 'High Scores' and a close button. The main content area is a table with three columns: 'Player', 'Score', and 'Remain'. The table lists five entries, each with a rank number in parentheses, a player name, a score, and a remaining value.

	Player	Score	Remain
0)	tommie	15006	25
1)	heidi	14608	18
2)	tommie	14460	50
3)	tommie	14370	9
4)	tommie	14200	11

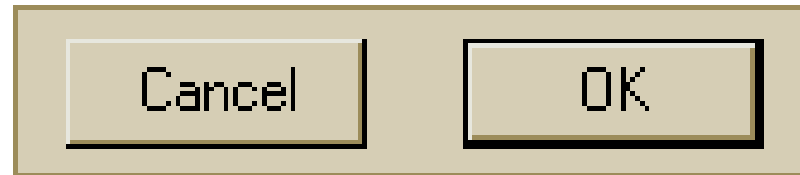
Przykład

- IBM's Audiostation
- Quiz: Co się stanie gdy
- wciśniemy Power?

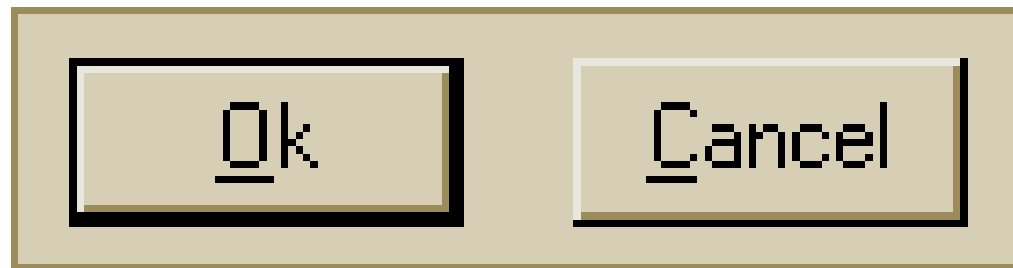


Przykład

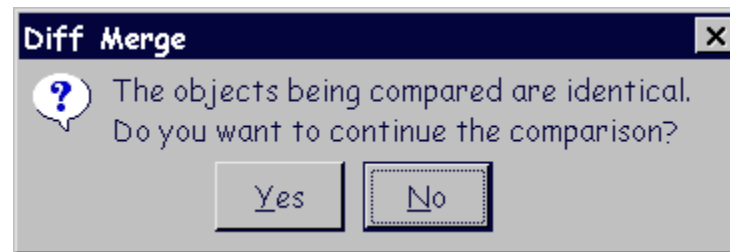
- Rozmieszczenie przycisków.



- Ok czy OK? Po co używać ALT+O i ALT+C, przecież i tak większość będzie wciskać ENTER i ESC.



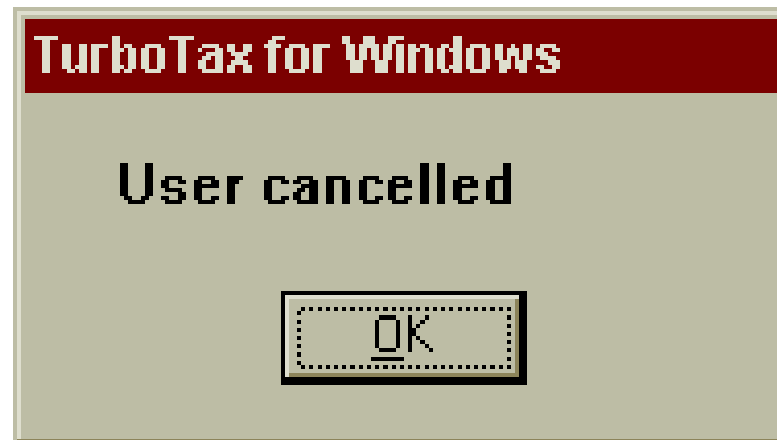
- Idiotyczne komunikaty:



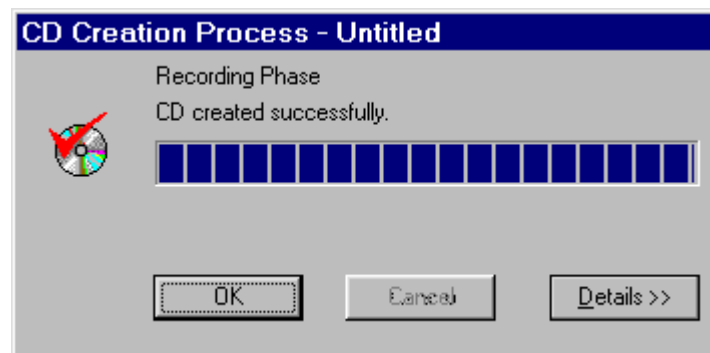
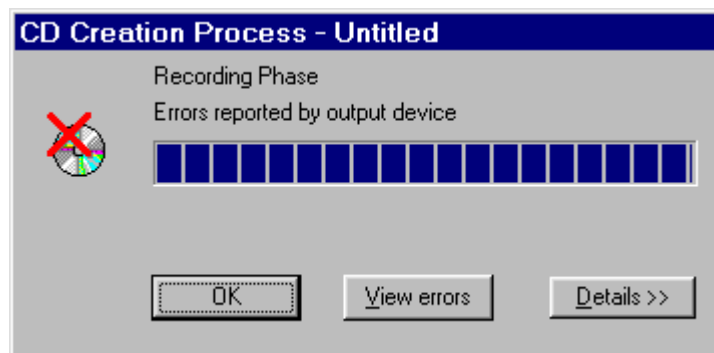
- NT workstation



- Quicken's Turbo Tax – anulowanie niemalże każdej czynności:



- Odpowiedni dobór kolorów...
- (Easy CD Creator)



- Zbyt mały kontrast utrudnia używanie aplikacji dla niepełnosprawnych. (Accessibility) (Apple's QuickTime 4.0 Player)

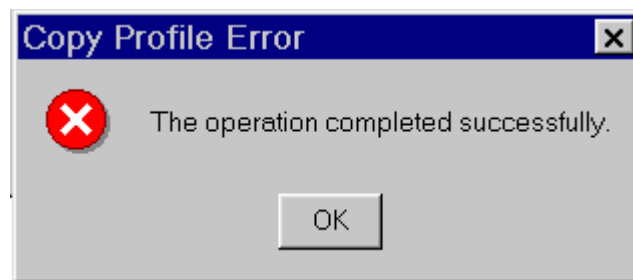


„Classified”, a shareware diary program:

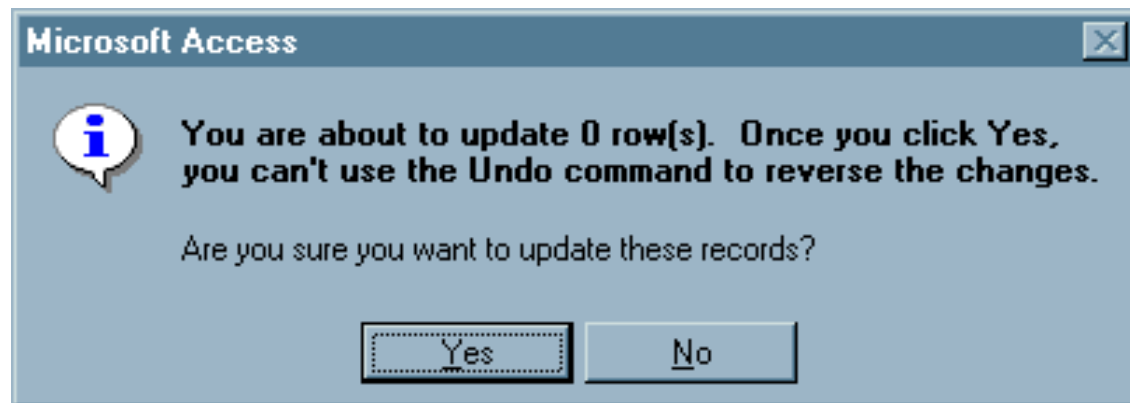
- nie można wprowadzać daty za pomocą klawiatury
- zmienić rok można tylko za pomocą paska



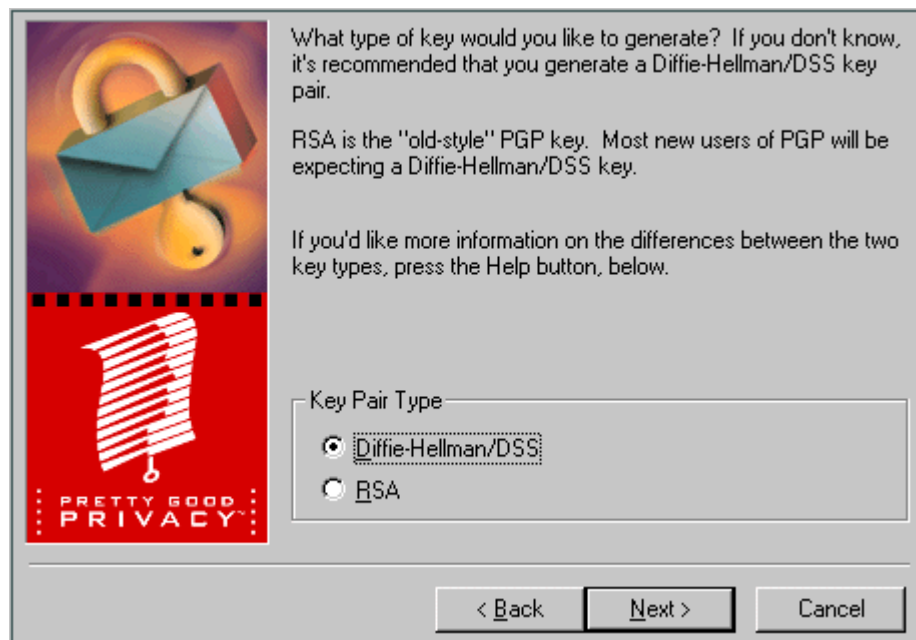
- – Od kiedy sukces jest błędem?



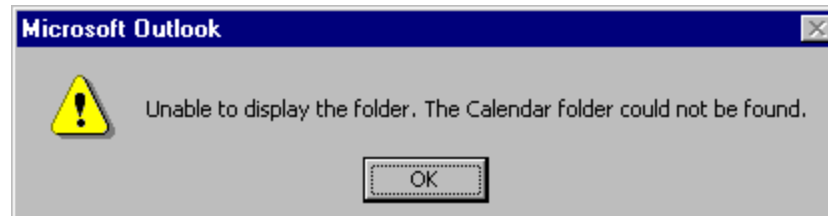
- – Komunikat pojawia się przy próbie uruchomienia kwerendy, która w wyniku nie modyfikuje żadnych wierszy.



- A gdzie przycisk Help?



- Nie istniejący obiekt...
- – nie, to tylko brak uprawnień do odczytu



- Moodle – system do tworzenia serwisów dydaktycznych
- Spróbuj zmienić opis dotyczący zajęć...

Plan tygodniowy

Organizacja zajęć:

- Układ zajęć:
 1. Pierwsza część zajęć: teoretyczny wykład wprowadzający
 2. Druga część zajęć: praktyczne wykorzystanie systemów operacyjnych i sieci komputerowych
- Zaliczenie:
 1. 60 punktów do zaliczenia to referat wraz z obroną ustną. Tematy referatów zostaną przedstawione na 3 zjeździe.
 2. 40 punktów można zdobyć za wykonane zadania na zajęciach. Punkty można otrzymać w pakietach po 8. W trakcie semestru na 5 zjazdach będzie można otrzymać punkty za wykonane zadania.



Forum aktualności →



Dodaj zasób...



Dodaj składową...



