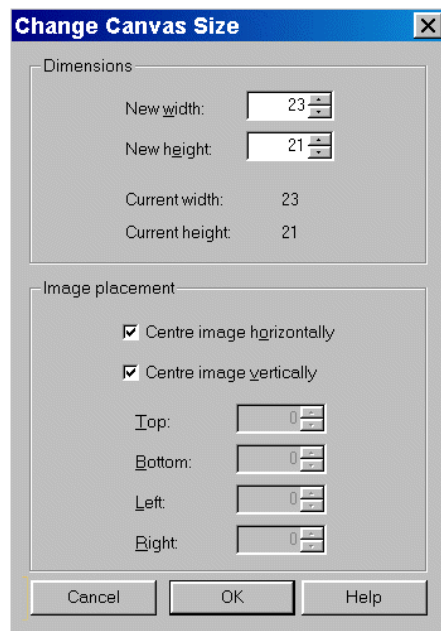


11

Käyttöliittymän rakennusaineet

Etsi virhe tästä kuvasta



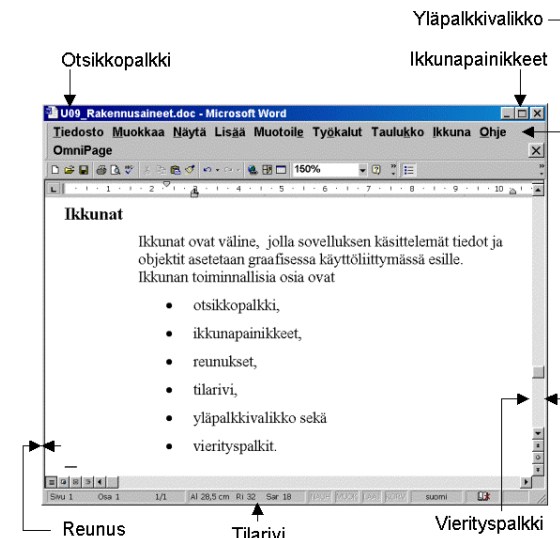
Kokeneemmat tietokoneen käyttäjät huomaavat varmaankin, että Ok-painike ja Cancel-painike ovat totutusta poikkeavassa järjestyksessä. Olemme tottuneet näkemään Ok-painikkeen ennen Cancel-painiketta. Jos painikkeet ovat vierekkäin, kuten tässä, Ok-painike on Cancel-painikkeen vasemmalla puolella. Jos painikkeet ovat allekkain, Ok-painike on ylimpänä.

Tämä sijoitteluvirhe ei ole aito esimerkki vaan kuvaeditorilla väärennetty. Jos joku oikeasti sijoittelisi painikkeet näin, monet käyttäjät painaisivat joskus tottumuksen voimasta Cancel-painiketta halutessaan painaa Ok-painiketta.

Tämä esimerkki havainnollistaa, että sovelluksen rakennusaineiden käyttötavoilla on väliä! Tässä luvussa käymme läpi graafisten käyttöliittymien yleisimpiä rakennusaineita, ja teemme huomioita niiden hyvistä käyttötavoista.

Käytämme esimerkkijärjestelmänä Windowsia. Esitetyt huomiot ovat suurelta osin sovellettavissa muihinkin graafisiin ympäristöihin.

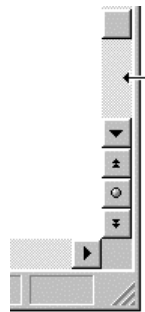
Ikkunat



Ikkunan ulkoasussa ei mikään vihjaa käyttäjälle, että ikkunaa voi siirtää vetämällä hiirellä yläpalkista. Hiiren kohdistin ei myöskään muuta ulkonäköään otsikkopalkin päällä, joten sekään ei vihjaa siirtämisen toimintamahdollisuudesta.

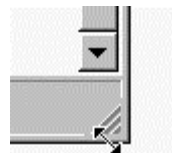
Sovellusten tekijät voinevat kuitenkin lähteä siitä, että ikkunoiden sulkemiset ja siirtämiset kuuluvat kaikille käyttäjille tuttuihin yleistaitoihin.

Ikkunoiden koon muuttamisen mahdollisuudet ovat paremmin esillä.



Kaikki käyttäjät huomaavat varmaankin ennemmin tai myöhemmin, että hiiren kohdistin muuttuu aina reunuksen päällä venytysmahdollisuuksia kuvaavaksi kaksipäiseksi nuoleksi.

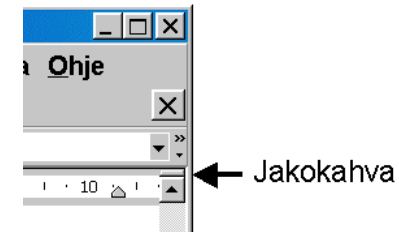
Alakulman muotoilu saattaa vihjaista käyttäjälle, että siihen liittyy joku toimintamahdollisuus. Jos käyttäjä vie kohdistimen sen päälle, hän näkee jälleen kaksipäisen nuolen, mutta nyt eri asennossa:



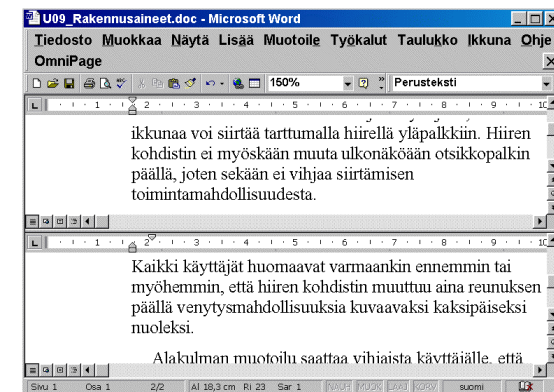
Tämä on varsin hyvä vihje siitä, että kulmasta tarttumalla ikkunaa voi venyttää haluttuun kokoon.

Vierityspalkkien käyttö kuuluu niihin taitoihin, jotka käyttäjät yleensä osaavat. Tekstinkäsittelyohjelmat ovat totuttaneet käyttäjät vierittämään ikkunan sisältöä pystysuunnassa. Käyttäjät eivät kuitenkaan yleensä pidä siitä, jos he joutuvat vierittämään sisältöä vaakasuunnassa. He haluavat nähdä ainakin joitain osakokonaisuuksia (esimerkiksi tekstin kappaleita) kerrallaan kokonaisina.

Monissa sovelluksissa vierityspalkkeihin liittyy jakokahva (split box):



Siitä hiirellä vetämällä voi avata toisen näkymän samaan dokumenttiin:



Tämä on hyvä esimerkki huonosti esillä olevasta toimintamahdollisuudesta. Se on kyllä periaatteessa esillä, mutta niin hyvin maastoutuneena, että esimerkiksi minä käytin Windowsia vuosikausia ennen kuin hoksasin sen.

Aktiiviset ja passiiviset ikkunat

Vain yksi ikkuna kerrallaan voi olla aktiivinen. Windowsissa aktiivisen ikkunan otsikkopalkki on sininen ja muiden ikkunoiden otsikkopalkit ovat harmaita. Windows ohjaa näppäimistön painallukset aktiiviselle ikkunalle.

Ikkunat voivat olla joko **modaalisia** tai **moodittomia**. Nämä termit eivät oikein juurikaan kuvaa, mistä on kysymys.

Modaalinen ikkuna on aina sovelluksen ainoa aktiivinen ikkuna. Kun modaalinen ikkuna on auki, sovelluksen muut ikkunat eivät reagoi hiiren klikkauksiin tai näppäinten käyttöön. Jos sovellus avaa toisen modaalisen ikkunan, tämä uusi varastaa jälleen kaiken huomion, ja edellinen modaalinen ikkuna jähmettyy.

Modaalisia ikkunoita käytetään sellaisissa tilanteissa, joissa käyttäjän täytyy suorittaa joku selkeästi rajoitettu tehtäväkokonaisuus.

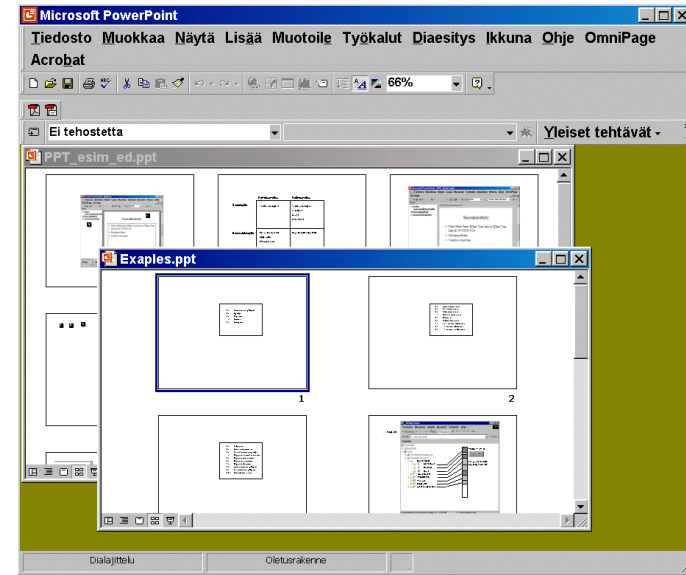
Dokumentin tallennus on tästä hyvä esimerkki.

Modaalisia ikkunoita käytetään myös ilmoitusikkunoina: käyttäjän on kuitattava ilmoitus esimerkiksi Ok-painikkeella ennen kuin sovelluksen käyttö pääsee jatkumaan.

Sovellus voi avata moodittomia ikkunoita rajoituksetta, ja käyttäjä voi käyttää niitä rinnakkain – valiten hiirellä klikkaamalla kulloinkin aktiivisen ikkunan. Moodittomia ikkunoita käytetään esimerkiksi silloin, kun käyttäjälle halutaan tarjota useita erilaisia näkymiä samaan dokumenttiin.

MDI: sovelluskohtainen työpöytä

Microsoftin Powerpoint luo Windowsin työpöydälle oman työpöytänsä. Kaikki käyttäjän avaamat Powerpoint-esitykset avautuvat tälle sovelluksen työpöydälle.

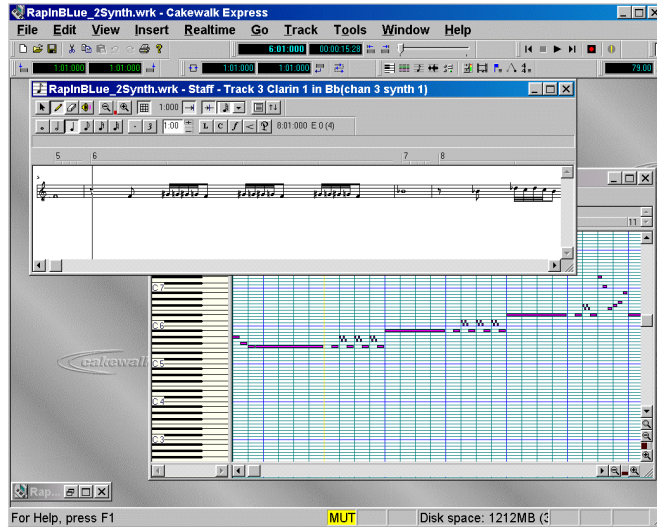


Tämän sovelluksen toimintamallin nimi on Multiple Document Interface eli MDI.

Sovelluksen työpöydällä olevat ikkunat käyttävät työpöytäikkunan valikkoa. Ne voivat lisätä tähän valikkoon omia ikkunakohtaisia toimintojaan.

Mutta miksi eri dokumenttien pitäisi olla samalla sovelluksen omalla työpöydällä? Miksi eri dokumentit eivät voisi aueta Windowsin työpöydälle omiin ikkunoihinsa? Minä en tiedä tähän kysymyksen mitään järkevältä tuntuva vastauksista!

Jotkut sovellukset käyttävät MDI:ia toisenlaisella tavalla:



Cakewalk on elektronisten soitinten ohjaamiseen tarkoitettu ns. sekvensseriohjelma. Sen työpöydällä voi olla useita erilaisia näkymiä samaan musiikkitiedostoon. Yllä näemme saman kappaleen saman soitinäänän sekä nuotteina että ns. pianorullana. Tuntuu ymmärrettävältä sitoa samaan dokumenttiin liittyvät eri näkymät samalle työpöydälle.

Sovellus tarvitsee pääikkunan

Jokaisella sovelluksella täytyy olla pääikkuna, jonka sulkeminen päättää sovelluksen toiminnan. Välinesovelluksissa (esimerkiksi tekstinkäsittelyohjelmissa) pääikkuna sisältää käsiteltävän dokumentin.

Välinesovelluksen pääikkuna on yleensä venytettävä. Otsikkopalkin alla on sovelluksen päävalikko omassa palkissaan. Lisäksi tällaisessa ikkunassa on erilaisia työkalu- ja komentopainikepalkkeja. Alimpana on yleensä sovelluksen tilarivi:



Tilarivi soveltuu dokumentin ja sovelluksen tilaa kuvaavien tietojen esittämiseen.

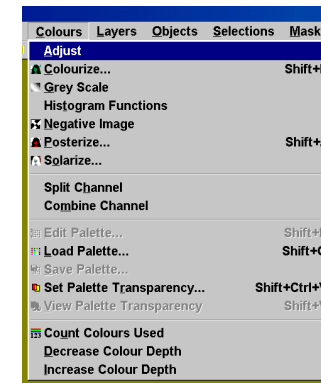
90-luvun alkupuolella käyttöliittymissä harrastettiin tilarivin paikalla ns. ohjeriviä. Se esitti kohdistettuun kontrolliin tai objektiin liittyviä selityksiä tai ohjeita. Tämä ratkaisu ei toiminut kovin hyvin, koska ikkunan alareuna on yleensä niin kaukana toiminnan kohteista, että ne eivät mahdu samaan näkökenttään. 90-luvun lopulta lähtien kelluvat opasteet (puhekuplaopasteet) ovat oikeutetusti syrjäyttäneet ohjerivit.

Palvelusovelluksessa pääikkuna sisältää usein sovelluksen päävalikon. Tällaiset ikkunat ovat yleensä kiinteän kokoisia.

Valikot jäsentävät sovelluksen

Ostin 2000-luvun alussa ensimmäisen digitaalikamerani. Ensimmäisistä kuvista totesin, että kameralla oli ongelmallinen tapa valottaa kuvia. Kuvan kirkkaista osista – kuten esimerkiksi taivaasta – tuli liian kirkkaita, ja taas varjoisista osista tuli liian tummia.

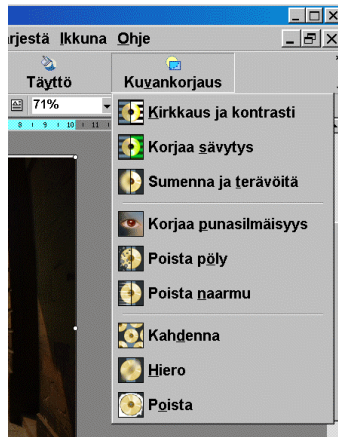
Ei hätää. Koneessani sattui olemaan monipuolinen kuvankäsittely-ohjelma Paintshop Pro. Sillä voi tehdä kuville hyvin monenlaisia asioita:



Kokeilin erilaisia toimenpiteitä. Minulle parhaimman tuloksen antavat välineet löytyivät valikkovalintojen ”Colours / Histogram Functions / Histogram Adjustments” takaa. Yksikään näistä valinnoista ei sano

mitään valotuksesta ja sen virheistä! Tämä sovellus ei siis keskustele käyttäjän kanssa hänen tavoitteistaan vaan niiden saavuttamisen keinoista.

Tällainen valikko löytyy Microsoftin kuvankäsittelyohjelma PhotoDraw'sta:



Tämä valikko keskustelelee paremmin käyttäjän kanssa hänen tavoitteisiinsa liittyvillä käsitteillä. Tämä PhotoDraw tarjoaa kuitenkin paljon Paintshopia vähemmän erilaisia valotuksen korjailuun sopivia toimintoja. Paintshopin monien mahdollisuuksien järjestäminen käyttäjän tavoitteista keskusteleviin valikoihin olisi varmaankin haastavaa, mutta tuskin mahdotonta.

Sovelluksen kehittäjä joutuu päättämään, että kenen tarpeiden ja näkökulman mukaan sovelluksen jaottelu ja valikot rakennetaan. Esimerkiksi Paintshopin tyyppiset keinokeskeistä näkökulmaa noudattavat valikot voivat olla perusteltuja, jos sovellus on tarkoitettu nimenomaan eksperttikäyttäjille.

Valikkojen sisällöt kuvastavat siis sovelluksen rakennetta. Helposti ymmärrettävän ja opittavan sovelluksen jaottelu noudattaa käyttäjälle tuttua ja ymmärrettävää näkökulmaa.

Päävalikko eli ikkunavalikko

Etsi virhe tästä kuvasta!



”Valokuvien korjailu” on kahden sanan mittainen valinnan otsikko. Ikkunavalikoissa ei juuri koskaan näe monisanaisia valinnan otsikkoja. Tähän on hyvä syy: otsikot ovat samalla rivillä peräkkäin, ja niiden välissä ei ole mitään visuaalisia erottimia. Jos valikossa on useita monisanaisia valintoja, käyttäjän voi olla vaikeaa hahmottaa, missä kukin otsikko alkaa.

Valikkopalkissa olevat valintojen otsikot ovat jatkuvasti näkyvillä. Tässä suhteessa se on hyvä paikka panna käyttäjän toimintamahdollisuuksia näkyvästi esille. Valinnan otsikon rajaaminen yhteen sanaan haittaa kuitenkin valikoiden ymmärrettävyyttä.

Ikkunavalikot soveltuvat erityisesti välinesovelluksiin: sovelluksen pääikkunassa on käsiteltävä dokumentti, ja yläpalkissa on esillä siihen liittyviä toimintamahdollisuuksia.

Ikkunavalikko ei ehkä ole hyvä ratkaisu palvelusovelluksen päävalikoksi. Esimerkiksi sovelluksen pääikkunaan komentopainikkeista rakennettu valikko voi olla parempi ratkaisu. Sellaisessa valikossa voidaan käyttää kaikilla tasoilla monisanaisia, kuvailevia otsikoita. Painikkeiden asettelulla voidaan havainnollistaa sovelluksen hierarkkista rakennetta. Hyvin järjestetystä painikevalikosta käyttäjä voi löytää nopeasti haluamansa toiminnon hyvinkin monien vaihtoehtojen joukosta.

Ikkunavalikoihin tulee usein keräytyä niin monenlaisia asioita, että niiden järjestäminen loogiseksi valikkorakenteeksi voi olla haasteellista. Näistä valikoista voi yleensä löytää mm.

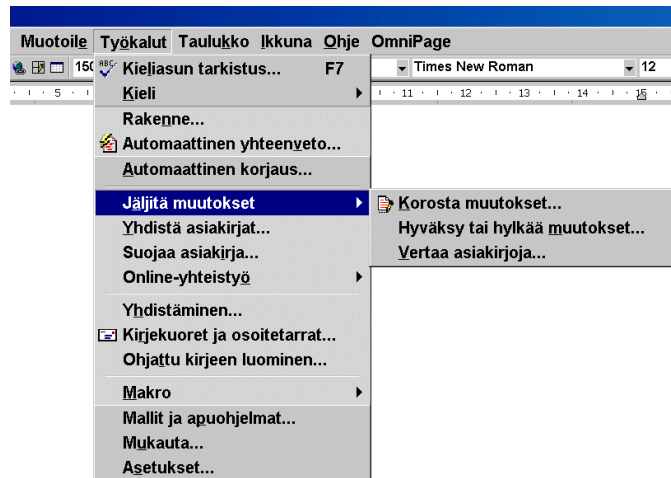
- ohjeita,
- sovelluksen asetuksia,

- käsiteltäviin objekteihin liittyviä toimintoja ja asetuksia,
- dokumentin hallintaan liittyviä toimintoja ja asetuksia
- jne.

Valikko on käyttäjälle ymmärrettävin, jos se jäsentää yhden ja vain yhden asian kattavasti (esimerkiksi: kevät/kesä/syky/talvi). Jos katsomme tyypillistä välinesovelluksen päävalikkoa, niin siellä on perinteisesti ensimmäisinä valintoina ”File” ja ”Edit”. ”File” on substantiivi, ja ”Edit” on verbi, joten ne eivät voi kuulua samaan jäsenyyteen.

Käytännössä välinesovellusten päävalikkoihin sisältyy aina useiden eri asioiden jäsenyyksiä. Valikkopalkki on niin arvokasta näkyvää tilaa, että sovelluskehittäjällä ei ole varaa olla käyttämättä sitä moniin tarkoituksiin.

Ikkunavalikkojen ominaisuudet



Ikkunavalikkoihin liittyy joukko vakiintuneita tapoja tehdä asioita:

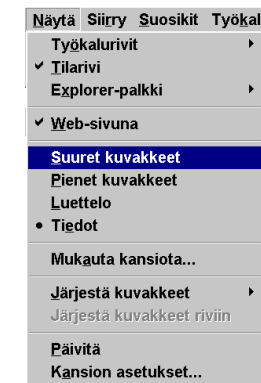
Pikanäppäimet: Jokaiseen valinnan otsikkoon liittyy alleviivattu kirjain (esimerkiksi kuvan valikossa Kieli). Ylimmällä valikkotasolla valinta voidaan tehdä näppäimillä painamalla ko. kirjainta yhdessä Alt-näppäimen kanssa. Esimerkiksi kuvan valikossa näppäinyhdistelmä Alt-ö

avaa Työkalut-valikon. Seuraavilla tasoilla pelkkä alleviivatuun kirjaimen painaminen riittää – Alt-näppäintä ei tarvita. Esimerkiksi S-näppäin valitsee tässä valikossa viimeisenä olevan valinnan ”Asetukset”.

Oikonäppäimet: Kuvan valikossa toimintonäppäin F7 avaa kieliasun tarkistusdialogin. Tämä oikonäppäin toimii, vaikka Työkalut-valikko ei olisi auki. Käyttäjä pääsee siis aina yhdellä näppäilyllä tähän toimintoon. Oikonäppäimiksi voidaan määritellä toimintonäppäinten lisäksi Ctrl-näppäinyhdistelmiä.

Navigointi: Kolme pistettä valinnan otsikon perässä tarkoittaa, että valinta avaa uuden ikkunan (esimerkiksi ”Kieli...” edellisessä kuvassa).

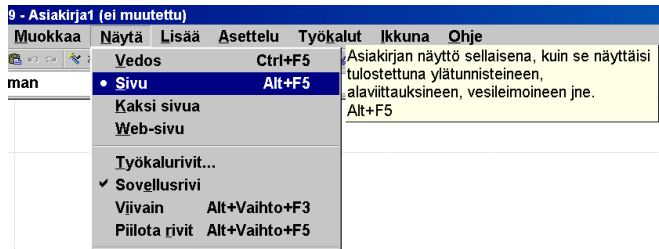
Alivalikko: Oikealle osoittava nuolenpää valikon oikeassa reunassa ilmaisee, että valinta avaa alivalikon (esimerkiksi ”Järjestä kuvakkeet” seuraavassa kuvassa).



Ruksivalinta ilmaisee kyllä/ei –valintaa. Esimerkiksi ruksi valinnan ”Tilarivi” vasemmalla puolella ilmaisee kyllä-valintaa, jolloin tilarivi näytetään.

Vaihtoehtovalinta ilmaisee valintaa kahdesta tai useammasta toisensa poissulkevasta vaihtoehdosta. Esimerkiksi edellisessä kuvassa pallukka valinnan ”Tiedot” kohdalla kertoo, että mikä neljästä vaihtoehdoisesta näkymästä on valittu.

Kelluvia opasteita (seuraavassa kuvassa) käytetään nykyisin säännöllisesti komentopainikkeiden yhteydessä, mutta harvemmin valikkojen yhteydessä. Ne ovat hyvä tapa antaa valintaan liittyvää lisätietoa.



Millainen valikko on nopea käyttää?

Valikon suunnittelutavoitteita ovat

- ymmärrettävyys ja
- käytön nopeus.

Ymmärrettävyyttä käsitelimme jo edellä. Sitä edistetään parhaiten käyttämällä käyttäjälle tuttuja ja hänen tarpeisiinsa ja tavoitteisiinsa liittyviä käsitteitä.

On muutamia hyviä syitä tehdä valikoista nopeita käyttää:

- Ihmisen työmuisti (pikamuisti) vuotaa pahasti! Työn alla olevia asioita alkaa hävitä sekunneissa. Tyypillisesti puolet työmuistin sisällöstä häviää noin kuudessa sekunnissa. Mitä pitempään valinta valikosta kestää, sitä vaikeampaa käyttäjälle on jatkaa siitä mihin jäätin.
- Toistuvissa tehtävissä sekuntienkin viiveistä voi kertyä kokonaisviiveitä, joilla voi olla merkitystä esimerkiksi asiakaspalvelun tehokkuudelle.

Pika- ja oikonäppäimet ovat hyviä tapoja parantaa valikon käytön nopeutta. Satunnaiset tai ajoittaiset käyttäjät tuskin muistavat kovin monia sovelluskohtaisia oikonäppäimiä. Sovelluksen säännölliset

käyttäjät oppinevat kuitenkin ajan mittaan hyödyntämään itselleen tärkeitä pika- ja oikonäppäimiä.

Valikkojen laatijat joutuvat aina pohtimaan, että tehdäänkö pitkiä vai syviä valikkoja. Pitkä valikko sisältää paljon valintoja peräkkäin samassa listassa, kun taas syvä valikkorakenne sisältää paljon sisäkkäisiä alivalikoita.

Syvä valikkorakenne auttaa sovelluksen käyttöä aloittelevaa käyttäjää. Jos meillä on esimerkiksi kuusitoista toimintoa järjestettynä neljään alivalikkoon, käyttäjä joutuu valintaa tehdessään lukemaan keskimäärin neljän valinnan otsikot. Jos sama määrä toimintoja on yhdessä valikossa, käyttäjä joutuu lukemaan keskimäärin kahdeksan valinnan otsikot. Suuremmilla toimintojen määrillä ero syvän valikon hyväksi on vielä suurempi.

Pitkä valikko taas auttaa sovelluksen ahkeraa käyttäjää: hän oppii vähitellen muistamaan, että kuinka pitkällä vaihtoehtojen listassa jokin hänelle tärkeä valinta on. Jos esimerkiksi valinta on listan loppupäässä, hän ei edes lue listan alkupäätä.

Pitkän valikon käytön nopeutta voidaan parantaa

- järjestämällä valikko sopivasti, sekä
- auttamalla kohdan löytämistä sopivilla visuaalisilla keinoilla.

Jos valinnoilla on jokin **luonnollinen järjestys** (esimerkiksi kevät/kesä/syky/talvi), käyttäjä oppii sen nopeasti ja käyttää myös sitä nopeasti.

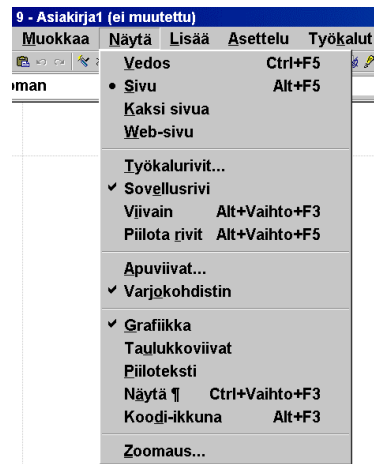
Aakkosjärjestetty valikko on melko nopea käyttää sitten, kun käyttäjä on keksinyt tämän järjestyksen. Sitä voi olla vaikeaa hahmottaa, jos valintoja ei ole kovin paljoa. Peräkkäin olevat, samalla kirjaimella alkavat valinnan otsikot antavat käyttäjälle vahvan vihjeen aakkosjärjestyksestä.

Aakkosjärjestys sopii erityisesti sisällöltään muuttuviin valikoihin. Niiden käyttäjä ei voi oppia muistamaan valintojen sijainteja, koska niillä

ei ole pysyvää sijaintia. Tällöin aakkosjärjestys on oikeastaan ainoa keino auttaa käyttäjää löytämään haluamansa valinnat.

Aiheen mukainen ryhmittely auttaa käyttäjää sitten, kun hän on sen oppinut.

Seuraavan kuvan esimerkissä on tyypillinen valikko, jossa on useaan eri asiaan liittyviä valintoja. Kaksi ensimmäistä ryhmää ovat luontevasti toisiinsa liittyviä, kun taas lopussa on sitten sekalaisempia asioita.

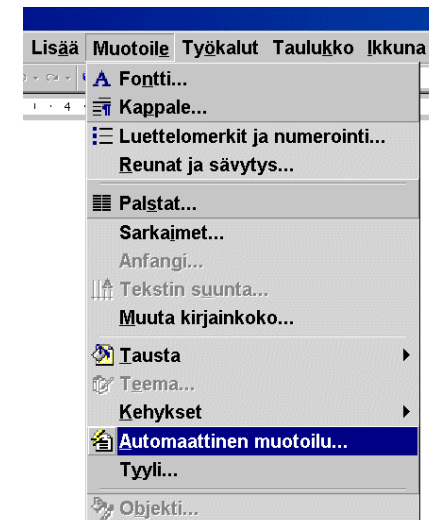


Tässä esimerkissä ryhmät on erotettu väliviivoilla toisistaan. Tällaisia kolmesta viiteen valintaan pitkiä visuaalisia ryhmiä kannattaa muodostaa, vaikka valinnoista ei erottuisikaan mitään luontevia ryhmiä. Tällainen jako muodostaa valikosta visuaalisen kokonaisuuden, joka auttaa käyttäjää muistamaan valintojen sijainnit suhteessa tähän hahmoon. Tämän hahmon toimivuutta auttaa, jos nämä ryhmät ovat vaihtelevan mittaisia.

Monien sovellusten valikoissa on kuvakkeita auttamassa oikean valinnan paikallistamista.



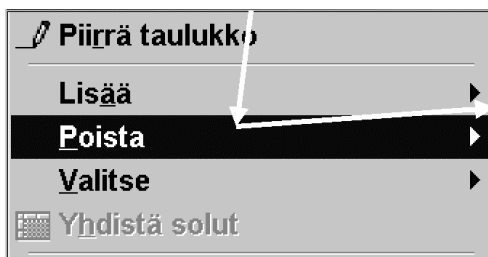
Luulen, että edellisen kuvan kaltainen kuvakkeiden riemunkirjava risuaita ei juuri auta halutun valinnan paikallistamisessa.



Tässä esimerkissä kuvakkeita on käytetty maltillisemmin. Ne toimivat nyt kiintopisteinä valikon visuaalisessa hahmossa. Ne auttavat käyttäjää muistamaan valinnan sijainnin suhteessa tähän hahmoon.

Oikean valinnan paikantamista pitkistä valikoista voidaan siis auttaa monella tavalla. Nämä kaikki keinot parantavat pitkien valikkojen kilpailukykyä suhteessa syviin valikkoihin.

Syviin ikkunavalikkoihin liittyy vielä eräs huomion arvoinen käytettävyysspulma. Siirtyminen hiirellä alivalikkoon vaatii sellaista tarkkaa näpertämistä, jota ei yksinkertaisesti voi tehdä vauhdilla. Hiiren liike ylhäältä alas täytyy ensin pysäyttää oikean valinnan kohdalle. Sitten täytyy lähteä oikealle täsmälleen suorassa kulmassa. Pienikin suuntausvirhe saattaa aiheuttaa väärän valinnan:



Jos hiiri kuvan esimerkissä liikkuu valkoisten nuolten mukaisesti, valituksi tulee valinta ”Lisää”, eikä tavoiteltu ”Poista”.

Jokainen uusi alivalikkotasoa lisää näiden hankalien suunnanmuutosten määrää. Jos ikkunan valikkopalkista avattavalla valikolla on esimerkiksi kaksi alivalikkotasoa, käyttäjä joutuu tekemään neljä suunnan muutosta! Oma kokemukseni on, että niin syvissä valikossa sattuu merkittävän usein väärän valinnan aiheuttavia lipsahduksia.

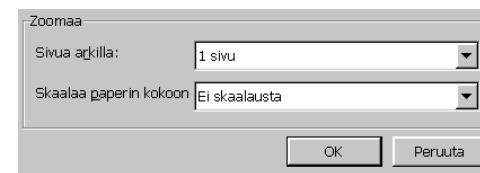
Moni asia siis puoltaa sitä, että tehdään mieluummin pitkiä kuin syviä valikkoja. Mutta valikkojen pituudella on tietenkin käytännölliset rajansa. Joskus alivalikkojen käytölle on hyviä syitä. Mutta silloinkin on syytä välttää useita alivalikkotasoja sisältäviä rakenteita.

Putkahdusvalikot ovat toinen graafisissa käyttöliittymissä yleinen valikkotyyppi. Niistä puhutaan edempänä, kun käsitellään oliopohjaisen käyttöliittymän erityiskysymyksiä.

Komentopainikkeet

Komentopainikkeilla on joukko vakiintuneita käyttötapoja.

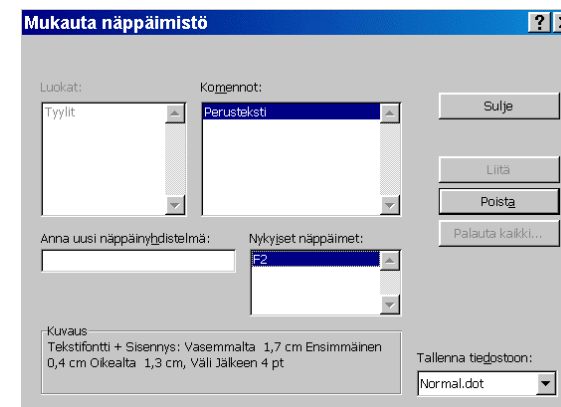
Lähtöpainikkeet ovat niistä yleisin:



Ok-painike ja Peruuta-painike (englanniksi Cancel) esiintyvät aina yhdessä dialogeissa, joissa käyttäjä antaa toisiinsa liittyviä tietoja tai valintoja. Jos dialogi suljetaan Ok-painikkeella, nämä tiedot ja valinnat tallennetaan ja/tai toteutetaan kaikki yhdessä. Jos poistuminen tapahtuu Peruuta-painikkeella, tietoja ja valintoja ei tallenneta/toteuteta.

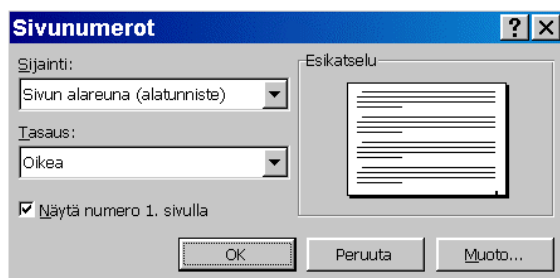
Kun tällainen dialogi on auki, Enterin painaminen näppäimistöllä vastaa lähtöä Ok-painikkeella, ja Esc-näppäimen painaminen vastaa lähtöä Peruuta-painikkeella.

Jos muutokset tallennetaan tai toteutetaan sitä mukaa, kun käyttäjä muuttaa tietoja lomakkeella, käytetään lähtöpainikkeena Ok-painikkeen sijasta Sulje-painiketta (englanniksi Close):



Komentopainikkeita käytetään myös dialogin sisäisiin toimenpiteisiin. Esimerkiksi edellisessä kuvassa Poista-painike poistaa viereisestä listasta valitun kohteen.

Komentopainikkeilla voidaan myös avata lisädialogi, joka sisältää sellaisia valintoja ja tietoja, joita kaikki käyttäjät eivät tarvitse.

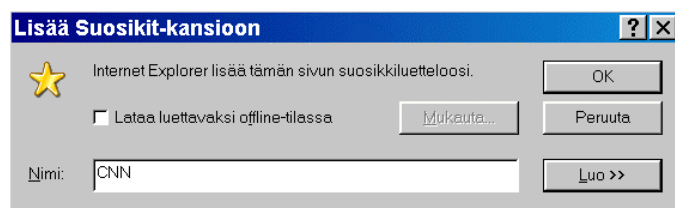


Tässä esimerkissä painike ”Muoto...” vie dialogiin, jolla käyttäjä voi muuttaa sivunumeron esitystapaa.

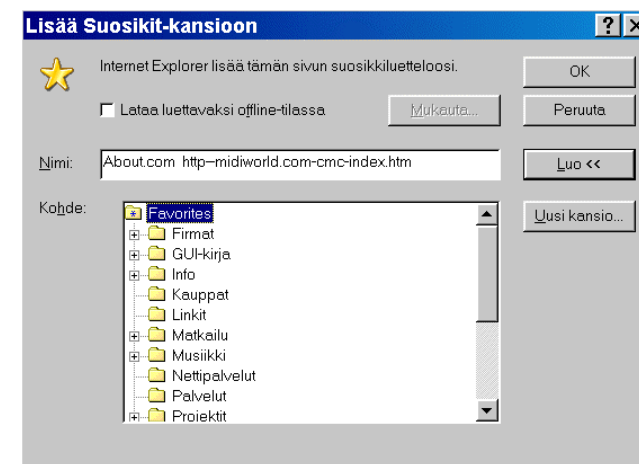
Myös komentopainikkeisiin voidaan liittää pikanäppäinyhdistelmä. Kuvan esimerkissä näppäily Alt-M liipaisee tämän komentopainikkeen.

Kolme pistettä painikkeen otsikon perässä on vakiintunut tapa ilmaista, että painike avaa uuden ikkunan.

Merkintä ”>>>” komentopainikkeen otsikossa ilmaisee, että painike avaa dialogin laajennuksen.



Kun käyttäjä klikkaa painiketta ”Luo>>>”, dialogiin avautuu laajennus:

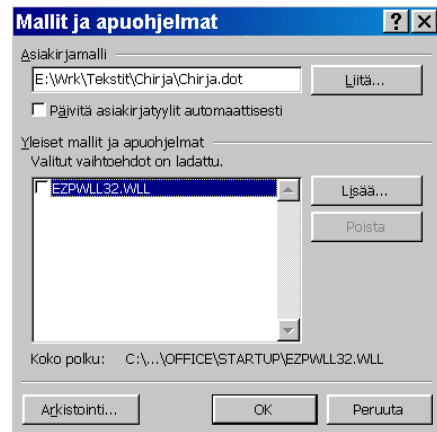


En ole oikein koskaan ymmärtänyt, että miksi näitä dialogeja ei voisi alun perin avata laajennetussa muodossaan. Ehkä tämän esimerkin tapauksessa on tavoiteltu yksinkertaisuutta. Olen tässä kovasti saarnannut asioiden näkymisen merkityksestä – en siis osaa uskoa asioiden piilottamisen siunauksellisuuteen!

Edellisessä luvussa puhuttiin kerroksittaisista käsitelmillestä. Voisiko aloittelevaa käyttäjää auttaa, jos muut kuin yksinkertaisimpaan käsitelmällisiin liittyvät asiat piilotetaan tällä tavalla?

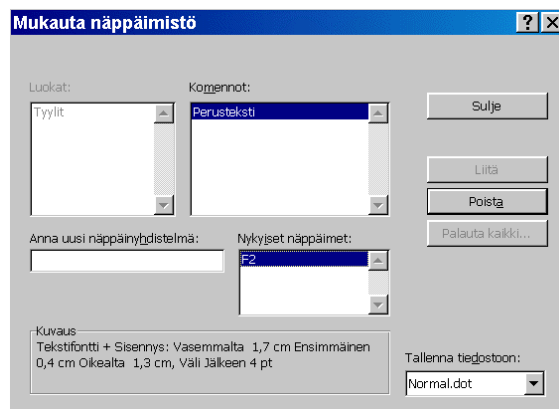
En usko, että siitä olisi mitään hyötyä. Yksinkertaisimpaan malliin liittyvät asiat voidaan sijoittaa dialogin alkuun, ja dialogin läpikäyntijärjestystä voidaan vielä ohjata erilaisilla visuaalisilla keinoilla. Kun satunnainen käyttäjä on dialogin alusta löytänyt tarvitsemansa asiat, hänen ei tarvitse välittää muusta. Toisaalta aloittelevakin käyttäjä voi olla utelias, jolloin asioiden esillä oleminen auttaa häntä sovelluksen oppimisessa.

Komentopainikkeilla voidaan myös käynnistää johonkin dialogin yksittäiseen tietoon liittyvä toimenpide – esimerkiksi tiedon haku:



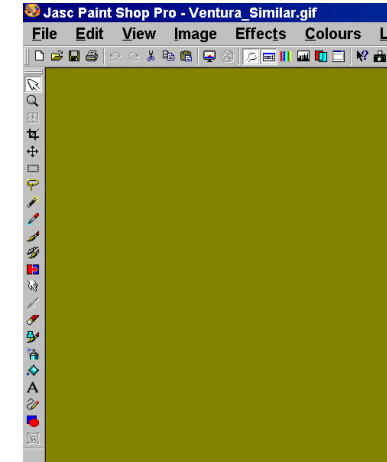
Tässä esimerkissä painike ”Liitä...” avaa dialogin, jolla Wordissa voidaan valita käytettävä asiakirjamalli.

Jos painike liittyy johonkin dialogin tietoon, sen täytyy sijaita niin lähellä tätä tietoa, että käyttäjä osaa vaivatta yhdistää ne toisiinsa. Esimerkiksi seuraavassa dialogissa painikkeen ”Poista” sijainti on niin epämääräinen, että ei ole aivan itsestään selvää, että se liittyy nimenomaan listaan ”Nykyiset näppäimet”.



Painikepalkit

Windowsin välinesovelluksissa on yleisesti kahdenlaisia painikepalkkeja:



- **Pikavalintapalkit** sijaitsevat ikkunavalikon alla. Siihen on yleensä sijoitettu valikon tärkeimpiä valintoja vastaavia painikkeita.
- **Työkalupalkit** sijaitsevat yleensä ikkunan vasemmassa reunassa.

Näissä painikkeissa ei yleensä ole tekstiä vaan ainoastaan painikkeen toimintaa jollain tavalla kuvaava kuvake. Tällaiset painikkeet vaativat tietenkin tuekseen kelluvat opasteet. Käyttäjät eivät arvaa painikkeen toimintaa pelkästä kuvakkeesta.

Painikkeiden käyttö on huomattavasti nopeampaa kuin valikkojen käyttö – sitten kun kuvakkeiden merkitykset on opittu. Painikkeet ovat jatkuvasti esillä, kun taas valikko on avattava. Kuvien tunnistaminen on myös nopeampaa kuin lukeminen – täytyyhän ennen lukemista tunnistaa kirjaimet!

Kuvakkeet

Kuvakkeiden tekemiseen on parasta käyttää visuaalisen viestinnän ammattilaista. Tässä ei pyritä antamaan valmiuksia niiden suunnitteluun. Tarkoituksena on vain antaa sovelluskehittäjälle eväitä kuvakkeiden

ideointiin osallistumiseen sekä ehdotettuja kuvakkeita koskeviin kannanottoihin.

Seuraavassa on joukko esimerkkejä sekä niiden takaa löytyviä ajatuksia:



Voit näyttää käyttäjälle tutun välineen, jota voidaan käyttää vastaavaan tehtävään – tässä piirtämiseen!



Voit kuvata toimintaan liittyvää tapahtumista, kuten tässä avaamista!



Voit kuvata toimenpiteen suorittavaa välinettä, kuten tässä kirjoitinta!



Voit kuvata toiminnan tai valinnan lopputulosta, kuten tässä tekstin kursivointia!



Voit käytä liikettä oikealle kuvaamaan tehtävän etenemistä. Länsimaissa liike vasemmalta oikealle kuvaa etenevää järjestystä. Tässä esimerkissä kuvataan toimenpiteen peruuttamista (Undo) kääntämällä etenevä liike takaisin vasemmalle.



Voit yhdistellä asioita. Tässä plussa ja miinus kertovat suurennuksen lisäämisen ja vähentämisen mahdollisuudesta.



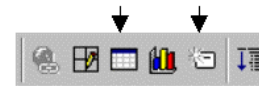
Voit käyttää osittaista samanlaisuutta korostamaan eroavuutta! Jos vierekkäin on useita jossain suhteessa samanlaisia kuvakkeita, käyttäjän huomio kohdistuu niiden eroavuuksiin. Tässä kaksi osittain samanlaista kuvaketta kertoo erilaisista listan kohtien esittämisen mahdollisuuksista.



Pidä kuitenkin huoli, että erilaisuus on riittävän erottuvaa. Näiden kahden kuvakkeen erot eivät kyllä erotu, jos ne ovat hyvin pieniä kuvaruudulla.



Huomio vakiintuneet esitystavat! Esimerkiksi välinesovellusten File ja Edit -valikoista yleisesti löytyvien vakiotoimintojen kuvakkeet ovat varsin vakiintuneet.



Tee kuvakkeista hyvin toisistaan erottuvia! Ääriviivat ja yleisväri ovat keskeisiä tekijöitä kuvakkeiden tunnistamisessa. Niiden merkitys korostuu, jos kuvakkeet ovat hyvin pieniä. Tässä on PowerPointin painikepalkista kaksi ääriviivoiltaan ja yleisväriltään hyvin samanlaista kuvaketta, jotka vielä sijaitsevat painikepalkissa aivan liian lähellä toisiaan.

Uskalla tehdä yksinkertaista! Parhaat kuvakkeet ovat muodoiltaan ja väreiltään hyvin pelkistettyjä.

Kuvaruutujen piirtotarkkuus eli resoluutio on parantunut nopeammin kuin ruutujen koot ovat kasvaneet. Jos esimerkiksi 19:n tuuman ruudulla käytetään 1600 x 1200:n pisteen piirtotarkkuutta, tyypilliset painikepalkin kuvakkeet ovat ruudulla varsin pieniä (noin 4 x 4 mm!). Runsaat yksityiskohdat ja riemunkirjava väritys eivät paranna näin pienen kuvakkeen tunnistettavuutta – ne vain suttaavat sen visuaalista ilmettä!

Kuvakkeiden ymmärrettävyyden testaus on suhteellisen helppoa. Siihen ei tarvita valmista sovellusta tai edes sen prototyyppiä. Testi voidaan tehdä täysin paperilla esimerkiksi seuraavasti: Järjestä paperille tulostetut kuvakkeet seinätaululle ikään kuin painikepalkkiin. Kirjoita erillisille lapuille kuhunkin kuvakkeeseen liittyvä toiminto tai komento.

Pyydä käyttäjää sijoittamaan laput taululle sellaisen kuvakkeen alle, joka parhaiten kuvaa lapussa mainittua toimintoa.

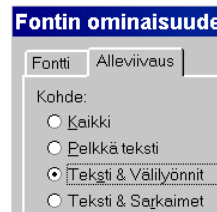
Valintojen tekeminen

Sovellukset kyselevät käyttäjiltä erilaisia valintoja:

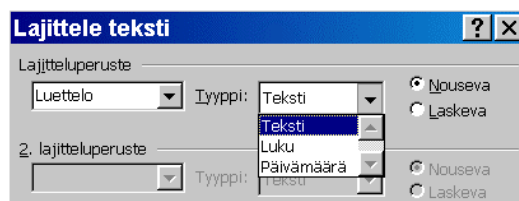
- Valinta mahdollisten arvojen joukosta: esimerkiksi lentolipun luokka voi yleensä olla turistiluokka, bisnesluokka ja ensiluokka.
- Kyllä/ei –valinta: esimerkiksi haluaako hotellivieras tupakoitsijoille tarkoitetun huoneen.
- Joukon valinta: esimerkiksi mitkä valinnaiset lisävarusteet uuteen autoon halutaan.

Valintojen tekemiseen on tarjolla erilaisia kontrolleja. Valinnan mahdollisten arvojen joukosta voi toteuttaa:

- valintapainikkeella (eli radionapilla)



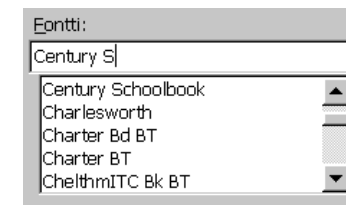
- pudotusvalikolla, tai



- listakentällä.



Sekä pudotusvalikkoon että listakenttään voi liittyä portaittainen hakukenttä:



Kun käyttäjä kirjoittaa etsimänsä vaihtoehdon nimeä, listasta haetaan jokaisen näppäilykirjaimen jälkeen ensimmäinen näppäilytekstiä vastaava valinta. Tällainen haku on tarpeellinen, jos vaihtoehtoja on kymmeniä, ja aivan välttämätön, jos vaihtoehtoja on satoja.

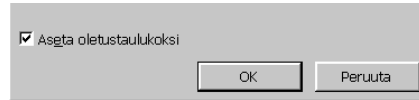
Joskus mahdollisten arvojen joukko vaihtelee. Esimerkiksi autovuokraamossa tarjolla olevien automallien valikoima vaihtelee. Tällaisissa tapauksissa täytyy käyttää pudotusvalikkoa tai listakenttää. Listakentät vievät enemmän tilaa kuin pudotusvalikot. Ne ovat pudotusvalikkoja parempia ainoastaan sellaisessa tilanteessa, jossa käyttäjän täytyy nähdä useampia vaihtoehtoja silloinkin, kun kohdistus on jossain toisessa kontrollissa.

Valintapainikkeet ovat käyttökelpoisia silloin, kun valinnan mahdollisten arvojen joukko on muuttumaton ja melko suppea. Esimerkiksi viikonpäivien joukko on toki muuttumaton, mutta seitsemän vaihtoehtoa vie lomakkeella aika paljon tilaa!

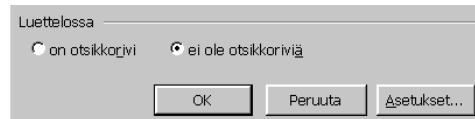
Valintapainikkeet vievät lomakkeella enemmän tilaa kuin pudotusvalikot, mutta ne ovat myös huomattavasti nopeampia käyttää.

Niiden käyttö voidaan tehdä hyvin nopeaksi, jos jokaiselle valinnalle määritellään oma pikanäppäinyhdistelmänsä.

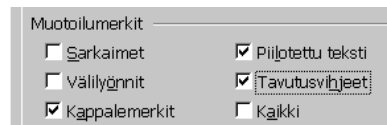
Kyllä/ei valinnan ainoa oikea väline on valintaruutu.



Tällaisia valintoja ei pitäisi tehdä esimerkiksi valintapainikkeilla, kuten seuraavassa Wordin dialogissa on tehty:



Valintaruutuja voidaan käyttää myös joukon valinnassa.



Tällainen valintaruutujen käyttö on mahdollista vain, jos valintojen joukko on muuttumaton. Jos niitä on hyvin paljon, niiden mahdollinen lomakkeelle voi olla käytännössä mahdotonta.

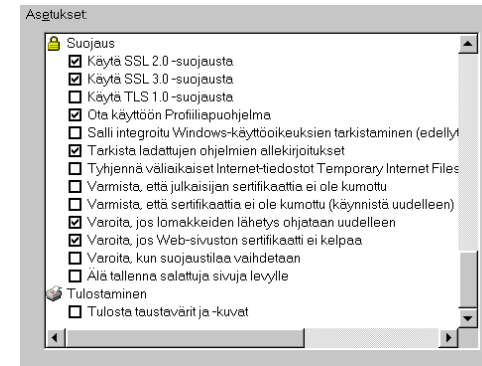
Jos joukko valitaan muuttuvasta joukosta, täytyy käyttää monivalintaista listakenttää:



Tällaisiin listakenttiin liittyy muutama käytettävyysohjelma.

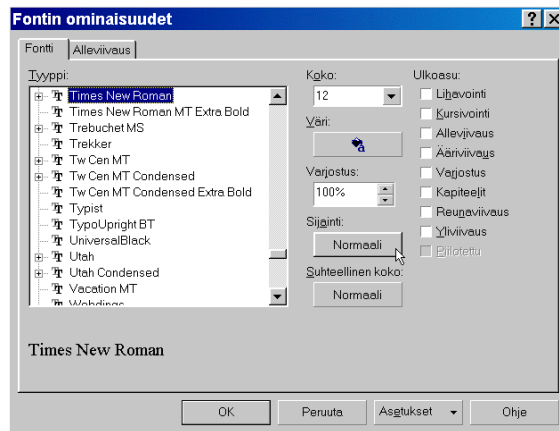
- Kenttä itsessään ei tarjoa mitään visuaalista vihjetä siitä, että käyttäjä voi valita useampia listan kohtia.
- Jos käyttäjä klikkaa valintaa hiirellä ilman Ctrl-näppäintä, tämä valinta korvaa kaikki aikaisemmat valinnat. Tämä saattaa jäädä käyttäjältä huomaamatta, jos kaikki valinnat eivät ole näkyvissä.

Tällaisiin muuttuviin monivalintalistoihin on parempikin kontrolli:

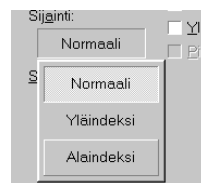


Tässä on valintaruudut sijoitettu listaan. Valintaruudut antavat käyttäjälle selvän visuaalisen vihjeen mahdollisuudesta valita useampia kohtia listasta. Valintojen tekemiseen ei tarvita mitään hankalia näppäinyhdistelmiä. Listan sisältöä voi muuttaa ohjelmallisesti. Tämä ratkaisu siis yhdistää monivalintaisen listakentän ja valintaruutujen parhaat puolet!

Valintojen tekemiseen on tarjolla käyttökelpoisia kontrolleja. Joskus sovellusten tekijät eivät tyydy niihin. Olen nähnyt useammassa sovelluksessa, että komentopainikkeita käytetään valintoihin - kuten esimerkiksi tässä WordPerfectin dialogissa.



Kun käyttäjä klikkaa painiketta Sijainti/Normaali, hänelle avautuu seuraava vaihtoehtojen lista:



Miksi tässä ei voitaisi käyttää aivan tavallista pudotusvalikkoa?
Komentopainike viestii toimenpiteiden suorittamisen mahdollisuutta eikä valinnan tekemistä!