

ŠIAULIŲ UNIVERSITETAS
TECHNOLOGIJOS FAKULTETAS
ELEKTRONIKOS KATEDRA

Tomas Kalinauskis

**TEKSTO REDAGAVIMO PROGRAMŲ
VALDYMAS ŽVILGSNIU**
Magistro darbas

Vadovas

doc. dr. G. Daunys

ŠIAULIAI, 2007

ŠIAULIŲ UNIVERSITETAS
TECHNOLOGIJOS FAKULTETAS
ELEKTRONIKOS KATEDRA

TVIRTINU

Katedros vedėjas

doc. dr. G. Daunys

2007 06

**TEKSTO REDAGAVIMO PROGRAMŲ
VALDYMAS ŽVILGSNIU**
Magistro darbas

Recenzentas

ŠU Technologijos fakulteto
Elektronikos katedros
prof. V. Lauruška

2007 06

Vadovas

ŠU Technologijos fakulteto
Elektronikos katedros
doc. dr. G. Daunys

2007 06

Atliko

RM5 gr.stu.
T. Kalinauskis

2007 06

SUMMARY

Kalinauskis T. Control of Text Editing programs by Eye Gaze: Master thesis of electronics engineer/ research advisor doc. dr. G. Daunys; Šiauliai University, Technological Faculty, Electronics Engineering Department. – Šiauliai, 2007. – 76p.

A human being, who has lost a possibility to control surrounding environment because of different reasons and to communicate with other people, faces many problems. Frequently, the people, who can not move or speak, become isolated from society and similarly. In such cases they feel themselves unhelpful, dependent on the people round about. The problem of the disabled people is the problem of society, which needs to be solved.

To print and edit text in Microsoft Word programme, Word Control programme for the disabled was created, i.e., the people, who have a big disability, i.e. cannot move at all.

The research was carried out at a computer class of Šiauliai University. A computer “LC Technologies Eye gaze System” was used for the research. The researched of 20-50 years old have participated in the research. The made research made to clear out what is printing speed and the number of made mistakes using different printing and editing methods.

TURINYS

| | |
|--|-----------|
| ĮVADAS | 8 |
| 1. TEKSTO ĮVEDIMO IR REDAGAVIMO ĮRENGINIŲ APŽVALGA | 9 |
| 1.1. Gyvenimo kokybė – kas tai? | 9 |
| 1.1.1. Vaikų ir neįgaliųjų gyvenimo kokybė netiriama | 9 |
| 1.1.2. Kiekviena sritis vertina savaip | 9 |
| 1.1.3. Sveikatos kokybė – ne klausimas, bet uždavinys | 9 |
| 1.1.4. Žmonės vadovaujasi asmeniniais prioritetais | 10 |
| 1.1.5. Universalus gyvenimo kokybės modelis | 10 |
| 1.1.6. Sveikatos kokybės standartas | 10 |
| 1.2. Kompiuteriai neįgaliesiems | 11 |
| 1.3 Kokios alternatyvos žvilgsnio valdymui? | 11 |
| 1.3.1. Įrenginiai su tiesiogine mechanine sąveika | 12 |
| 1.3.2. Įrenginiai su optine sąveika | 13 |
| 1.3.3. Elektrinė sąveika | 15 |
| 1.4. Prietaiso funkcijų panaudojimas | 15 |
| 1.5. Valdymas žvilgsniu | 16 |
| 1.5.1. Žvilgsnio krypties naudojimo metodai | 19 |
| 1.5.2. Akies pelės naudojimas | 20 |
| 1.5.3. Žvilgsnio sąveikos komunikacija | 20 |
| 1.6. Teksto rinkimo akimis sistemos | 21 |
| 1.6.1. Bendrieji teksto įvedimo metodai | 22 |
| 1.6.2. Dasher projektas | 23 |
| 1.6.3. Kaip dirba Dasher? | 24 |
| 1.6.4. Kaip tikimybių modelis sąlygoja ženklų išsidėstymo schemą (išplanavimą) | 26 |
| 1.6.4.1. Kalbos modelis | 27 |
| 1.6.4.2. PPM teksto suspaudimo algoritmas | 27 |
| 1.6.5. Dasher koncepcija | 28 |
| 2. SUKURTOS TEKSTO REDAGAVIMO PROGRAMOS ALGORITMAS | 29 |
| 3. SUKURTOS TEKSTO REDAGAVIMO PROGRAMOS VALDYMAS | 38 |
| 4. EKSPERIMENTINIS TYRIMAS | 43 |
| 4.1. Tiriamųjų pasirinkimas | 43 |

| | |
|----------------------------------|-----------|
| 4.2 Tyrimo eiga | 44 |
| 5. REZULTATAI | 47 |
| IŠVADOS IR SIŪLYMAI | 50 |
| LITERATŪRA | 51 |
| PRIEDAI | 52 |

LENTELĖS

| | |
|---|-------|
| 4.2.1. Rezultatų fiksavimo lentelė | 46 p. |
| 5.1. Tyrimo metu užfiksuotų rezultatų lentelė | 47 p. |

PAVEIKSLAI

| | |
|---|-------|
| 1.3.1.1 pav. „Siurbimo/pūtimo jungtukas“ | 12 p. |
| 1.3.1.2 pav. Klaviatūra su apsauga | 13 p. |
| 1.3.2.1 pav. Principinė veikimo schema ir sukonstruotas įrenginys | 13 p. |
| 1.3.2.2 pav. Galvos pelė naudojanti infraraudonąjį spinduliavimą | 14 p. |
| 1.3.2.3 pav. Kameros pelė | 14 p. |
| 1.3.3.1 pav. Kibernetinė sistema aptinka smegenų ir kūno signalus galvos sensoriuje | 15 p. |
| 1.6.1. pav. Teksto rinkimo akimis sistema Dasher | 22 p. |
| 1.6.3.1 pav. Spausdinimas įvairiomis programomis ir įvedimo įrenginiais. Vertikaliai – spausdinimo greitis (žodžiai per min.). Horizontaliai – laikas panaudotas įvairioms sąsajoms | 24 p. |
| 1.6.3.2 pav. Dasher programa pagal raidžių pasirodymo tikimybes pasiūlo sekantį ženklą | 25 p. |
| 1.6.4.1 pav. Aritmetinis alfabeto kodavimas raidėms {a, b, c} | 26 p. |
| 1.6.4.2. pav. Sąsajos dinamika | 27 p. |
| 2.1 pav. Laiko ant klavišo praleidimo skaičiavimo algoritmas | 29 p. |
| 2.2 pav. Teksto redagavimo klaviatūros algoritmas | 30 p. |
| 2.3 pav. Teksto redagavimo klaviatūros algoritmas (tęsinys) | 31 p. |
| 3.1 pav. Pagrindinė programos „WordControl“ valdymo klaviatūra | 38 p. |
| 3.2 pav. Atidarytas Microsoft Word programos langas su atidarymo klaviatūra ir pagrindine valdymo klaviatūra | 39 p. |
| 3.3 pav. Atidarytas pasirinktas Microsoft Word dokumentas su pagrindine valdymo klaviatūra. | 40 p. |
| 3.4 pav. Teksto redagavimo klaviatūra pasirinktame Microsoft Word dokumente | 41 p. |
| 3.5 pav. Teksto redagavimo klaviatūra su įjungtu „Shift“ klavišu pasirinktame Microsoft Word dokumente | 42 p. |
| 4.1.1 pav. Tiriamojo darbo vieta | 43 p. |
| 4.2.1 pav. Priartintas akies vaizdas ir visas kameros vaizdas | 45 p. |
| 4.2.2 pav. Kalibravimas | 45 p. |
| 5.1 pav. Tiriamiesiems pateikto teksto rinkimo ir redagavimo standartine klaviatūra, pelė ir žvilgsniu valdoma WordControl programa, laiko rodikliai | 47 p. |
| 5.2 pav. Tiriamiesiems pateikto teksto rinkimo ir redagavimo standartine klaviatūra, pelė ir žvilgsniu valdoma WordControl programa, padarytų klaidų skaičiaus rodikliai | 48 p. |
| 5.3 pav. Sąryšis tarp teksto rinkimo ir redagavimo metodų | 49 p. |

IVADAS

Pasaulyje yra labai daug neįgaliųjų žmonių. Neįgalūs žmonės, kaip ir visi kiti, yra unikalūs. Daugelis jų gebėjimų primena „vidutiniškų“ žmonių, laikomų neturinčiais negalių. Vis dėlto negalia – tai bruožas, būdingas nedaugeliui žmonių. Svarbu suprasti, kas yra negalia ir tinkamai ją priimti, bet negalima pamiršti, kad neįgalūs žmonės turi ir įvairių gebėjimų – kitų bruožų, vienijančių juos su dauguma [1].

Žmogus, dėl įvairių priežasčių praradęs galimybę valdyti jį supančią aplinką, praradęs galimybę bendrauti su kitais žmonėmis, susiduria su daugeliu problemų. Dažnai asmenys, negalintys judėti ar kalbėti, tampa atstumtais, izoliuotais nuo visuomenės ir pan. Tokiais atvejais jie jaučiasi nenaudingi, priklausomi nuo aplinkinių. Aplinkinių požiūris į žmones su negalia taip pat nėra pozityvus. Neįgaliųjų žmonių problema yra visuomenės problema, kurią reikia spręsti.

Visuomenė privalo siekti, kad aplinka, kurioje gyvena neįgalieji, taptų jiems draugiškesnė, saugesnė, kad toje aplinkoje jie galėtų jaustis, kiek galima savarankiškesni. Tinkamiausias šios problemos sprendimas būtų bent iš dalies sugrąžinti jiems tai, ką jie prarado. Pasitelkiant į pagalbą sparčiai besivystančias technologijas (telekomunikacijų ir kompiuterinės technikos srityje), yra įmanoma išspręsti šią problemą.

Darbo tikslas – sukurti teksto redagavimo programą valdomą žvilgsniu ir atlikti tyrimą, kurio tikslas – parodyti, kokie yra teksto rinkimo greičiai, daromų klaidų skaičius, išsiaiškinti ar patogi sukurtos programos klaviatūra ir jos valdymas.

Tyrimo uždaviniai:

- atlikti tyrimą, pagal kurį būtų galima nustatyti teksto rinkimų greičius ir daromų klaidų skaičių;
- rasti sąryšį tarp teksto rinkimo standartine klaviatūra ir žvilgsniu valdoma sukurta WordControl programa;
- rasti sąryšį tarp teksto rinkimo standartine klaviatūra ir pele valdoma sukurta WordControl programa;
- palyginti gautus rezultatus ir pateikti išvadas.

1. TEKSTO ĮVEDIMO IR REDAGAVIMO ĮRENGINIŲ APŽVALGA

1.1. Gyvenimo kokybė – kas tai?

1.1.1. Vaikų ir neįgaliųjų gyvenimo kokybė netiriama

Iš [2] medicininių straipsnių, kurių per pastarąjį dešimtmetį pasirodė spaudoje apie 4500, matome, kad keičiasi gyvenimo kokybės medicininio aspekto turinys. Ankstesniuose buvo analizuojama gyvenimo kokybės sąsaja su liga ir gydymo poveikiu, pastarojo laikotarpio straipsniuose pastebima tendencija nagrinėti ryšį tarp gyvenimo kokybės ir sveikatos – kaip visuomenės galimybės žmogui fiziškai, protiškaiai ir socialiai veikti, gyventi aktyvų ir produktyvų gyvenimą, pajusti pasitenkinimą ir laimę.

Svarbiausi gyvenimo kokybės elementai turėtų būti vienodi visai visuomenei – tiek sveikiesiems, tiek neįgaliesiems ir lygiai tenkinti tiek pagrindinius (pvz. sveikatos), tiek aukštesnius (tarkim, draugystės, saviraiškos) visų grupių žmonių poreikius.

Tačiau mūsų visuomenės ateitis ir viena pažeidžiamiausių jos grupių – vaikai retai įtraukiami į gyvenimo kokybės tyrimų projektus. Lietuvoje iki šiol dar nepateiktas vaikų ir neįgaliųjų gyvenimo kokybės vertinimas.

1.1.2. Kiekviena sritis vertina savaip

Mokslinio [2] požiūrio į gyvenimo kokybę užuomazga buvo Allard socialinė studija „Turėjimas, mylėjimas ir buvimas“. Materialiniai poreikiai ir jų įgyvendinimas buvo pavadinti gyvenimo kokybės lygiu. Šiandien jo klausimai studijuojami daugelyje mokslo sričių, ir kiekviena pateikia savitą sampratą. Pavyzdžiui, sociologija aiškina, kad gyvenimo kokybė – tai būti laimingam. Ekonomikos atstovai gyvenimo kokybę vertina kaip buvimą turtingam. Jų naudojamas tradicinis ekonomikos augimo matas – bendrasis nacionalinis produktas bei 1970 m. Pasaulio banko sukurtas fizinis gyvenimo kokybės indeksas, apimantis socialinius bei su sveikata susijusius kintamuosius dydžius. Medicinoje gyvenimo kokybė vertinama kaip pusiausvyra tarp sveikatos ir ligos.

1.1.3. Sveikatos kokybė – ne klausimas, bet uždavinys

Nuo [2] 1977, kai PSO pradėjo vykdyti globalią sveikatos politiką „Sveikata visiems“, svarbi reikšmė suteikiama gyvenimo kokybei. Teigiama, kad sveikatos kokybė – ne klausimas, kaip

prailginti gyvenimą, o uždavinys, kaip metams suteikti pilnatvę. Studijuojami žmogaus gebėjimai, kaip, netgi nepaisant negatyvių gyvenimo aplinkybių, išlaikyti sveikatą. Svarbu gebėti suvokti situaciją, kontroliuoti jausmus, rasti gyvenimo prasmę.

1.1.4. Žmonės vadovaujasi asmeniniais prioritetais

Vis [2] dėlto iki šiol dar nėra suformuluotas vieningas gyvenimo kokybės apibrėžimas, kadangi žmonės, dalyvaujantys apklausoje, pirmiausia vadovaujasi asmeniniais prioritetais, o tai priklauso nuo individo kultūros ir jo asmenybės bruožų.

Pasaulinė sveikatos organizacija siūlo tokį gyvenimo kokybės apibrėžimą: „Tai individų vietos gyvenime suvokimas kultūriniam ir jų užimamos padėties kontekste, susijęs su jų tikslais, viltimis, standartais ir požiūriais. Tai plati koncepcija, kuriai didelės įtakos turi asmens fizinė sveikata, psichologinė būklė, nepriklausomybės laipsnis, socialiniai ryšiai ir santykiai su aplinka“.

1.1.5. Universalus gyvenimo kokybės modelis

Šiuo [2] metu yra nustatytas universalus gyvenimo kokybės modelis, kuris suskirstytas į 4 sferas: globalinę, išorinę, tarpasmeninę ir asmeninę. Pirmajai, globalinei, priklauso žmogaus aplinka, jo teisės ir politika. Išorinei – darbas, šeimos pragyvenimo lygis ir gyvenamoji vieta, būstas. Tarpasmeninei – šeima, artimi santykiai ir tarpasmeniniai ryšiai. Ketvirtajai, asmeninei, priklauso fizinė, psichinė ir dvasinė dimensijos.

1.1.6. Sveikatos kokybės standartas

Pagal [2] šį universalųjį modelį penkiose šiaurės šalyse - Švedijoje, Suomijoje, Danijoje, Norvegijoje ir Islandijoje, kurių aukštas bendras gyvenimo lygis - atlikti 2-18 metų vaikų, ir vaikų, sergančių itin sunkiomis ligomis – cistine fibroze ir mielomeningocele (galvos smegenų auglys), gyvenimo kokybės tyrimai. Buvo atrinkta po 3 tūkstančius vaikų, jų šeimoms išsiųsti klausimai raštu ir gauti atsakymai. Labai aktyviai apklausoje dalyvavo šeimos, auginančios neįgalius vaikus, kiti susidomėjimo rodė kiek mažiau. Po to buvo suskaičiuota vaikų sveikatos kokybės bazinė vertė (reikalingiausiomis sąlygomis pasitenkinimui pasiekti). Šių bazinių verčių rinkinys ir suformavo sveikatos kokybės standartą.

1.2. Kompiuteriai neįgaliesiems

Negalia [3] neturi būti kliūtimi siekti išsilavinimo ir integruotis į visuomenę, ieškoti informacijos, ją gauti, skleisti. Šią kliūtį didele dalimi gali pašalinti pati visuomenė, padėdama savo nariams. Labai svarbų vaidmenį šioje srityje vaidina informacijos technologijos. Vienas iš svarbiausių žmogaus poreikių – bendravimas. Neįsivaizduojame savęs negalėdami suprasti kito. Aklieji bendrauja Brailio raštu, kurtieji – rankų gestų kalba. Fiziškai nesveiki žmonės dažnai nevaldo tam tikrų raumenų grupių ar galūnių. Šie žmonės gali kalbėti ir naudodami alternatyvią klaviatūrą, jungiklius bei specialią programinę įrangą. Jau šiandien įvairiose pasaulio šalyse bendrauti su neįgaliaisiais naudojama simbolių kalba, dar vadinama augmentatyviosios komunikacijos kalba. Tai įvairių simbolių kalba. Ji vartojama tada, kai neįgalusis turi kalbėjimo problemų. Lietuvoje tai dar neįprasta bendravimo forma, todėl specialių techninių priemonių jai įgyvendinti turime labai mažai. Neįgaliųjų gyvenime vis didesnę vaidmenį atlieka kompiuteris ir speciali programinė įrangą. Ji padeda lavinti įvairius pojūčius ir moko geriau orientuotis aplinkoje, ugdo neįgaliųjų savarankiškumą.

Siekiant pagelbėti silpnaregiams bei akliesiems sukurtos specialios programos šriftui didinti, rašybai Brailio raštu, tekstams skaityti balsu.

Viena iš perspektyviausių balso technologijų naudojimo sričių – suteikiamos įvairios galimybės neįgaliesiems žmonėms. Balso technologijos dažnai esti net vienintelis tokių žmonių integravimo būdas. Neįgaliesiems skirtos balso technologijos iš esmės pagrįstos tais pačiais kalbos apdorojimo technologijų pasiekimais: kalbos atpažinimu, kalbos sinteze ir kartais asmens atpažinimu.

Lietuvos neįgalieji dar nedaug kuo gali pasidžiaugti. Lietuvių kalbai pritaikytų sistemų dar yra labai mažai, bet darbai jau pradėti. Jeigu nekreipsime dėmesio į lietuviškų garsų iškraipymus, skaitymui galima pasinaudoti užsienyje sukurtomis programomis.

Šiuo metu dauguma aklųjų ir silpnaregių Lietuvoje naudoja būtent šias programas. Lietuvoje egzistuojantys sintezatoriai (programiniai) su kitomis programomis bendrauja naudodami Microsoft SAPI ir ekrano skaitymo programa JAWS. Nuo 2001 metų Lietuvoje naudojamas čekų kompanijos Rosasoft sukurtas lietuvių kalbos sintezatorius, kuris kartu su JAWS programa veikia Windows aplinkoje.

1.3 Kokios alternatyvos žvilgsnio valdymui?

Yra [4, 5] daug įrenginių, kurie gali padėti žmonėms su negalia. Kai kurie žmonės gali naudoti tik dalį savo kūno, pavyzdžiui kaip kumštis, pirštas ar pėda. Kai kurie gali tik akimis mirksėti ar galvą judinti. Aišku čia reikalingas sąmoningas judesys o ne atsitiktinis, kaip drebulys. Ginčijamasi kuris

įvedimo įrenginys gali būti efektyvus naudojant mažiausiai pastangų, nesuteikiant vartotojui diskomforto, santykiui su pasiektais rezultatais.

Kompiuteriniai įvedimo įrenginiai žmonėms su negalia yra klasifikuojami pagal skirtingus ypatumus. Pagal Shein, 1992, yra trys pagrindinės grupės:

- Jungikliai
- Klaviatūros
- Sekimo įrenginiai

Kita vertus, yra trys būdai kuriais šie įrenginiai gali būti valdomi:

- Mechaninė sąveika
- Optinė sąveika
- Elektrinė sąveika

Mechaninė sąveika reikalauja, kad vartotojas atliktų apgalvotą judesį jungikliui aktyvuoti, pavyzdžiui naudoti putimo slėgį, pūtinėti arba įkvėpti. Su „optinė sąveika“, sąveikai tarp žmogaus ir įvedimo įrenginio suveikimo, naudojamos matomos šviesos bangos arba infraraudonieji spinduliai. Su „elektrinė sąveika“, įrenginiai naudoja tiesiogiai įgytus elektrinius signalus nuo vartotojo odos ar audinio.

1.3.1. Įrenginiai su tiesiogine mechanine sąveika

Vartotojai [4, 5], kuriems sudėtinga valdyti rankas, bet kurie gali efektyviai valdyti galvą, gal būti uždedamas specialus šalmas. Prie šalmo pritvirtinta paprasta lazdelė, kuria žmogus gali valdyti įprastą klaviatūrą. Naudojamas dvigumas jungtukas „siurbimo/pūtimo jungtukas“ aktyvuojamas įkvėpiant/iškvepiant. Vartotojas naudojantis tokį įrenginį pavaizduotas 1.3.1.1 paveiksle. Siurbimo/pūtimo jungtukas“ susideda iš ant galvos pritvirtinto rėmo su nuvestu vamzdeliu į burną, vamzdelis sujungtas su jungtuko dėžės plastikiniu vamzdeliu. Siurbimas ir pūtimas yra paverčiami „užbaigtu jungikliu“ „Jungtuko dėžės“ viduje.



1.3.1.1 pav. „Siurbimo/pūtimo jungtukas“

Vietoj įtaiso pritvirtinimo prie vartotojo, įtaisas gali būti pritvirtintas prie pagrindinio įtaiso kas padarytų jį labiau prieinamą. Pavyzdžiui, prie įprastos klaviatūros gali būti pritvirtinta klaviatūros apsauga, pavaizduota 1.3.1.2 paveiksle. Tai gali padėti tiems kurie turi drebulį ar kitus sunkumus prieinant prie klaviatūros, įdubimai klavišų apsaugoje padeda tiksliau nuteikti pirštus į taikomas klavišus.



1.3.1.2 pav. Klaviatūra su apsauga

1.3.2. Įrenginiai su optine sąveika

Technologija [4, 5] aptinkanti sąmoningų mirksėjimų atspindžių kiekį nuo atmerktos ir užmerktos akies. Aptikimui naudojami infraraudonųjų spindulių siūstuvai ir imtuvai, jie tvirtinami abiejose akinių pusėse, pavaizduota 1.3.2.1 paveiksle.



1.3.2.1 pav. Principinė veikimo schema ir sukonstruotas įrenginys

Ši sistema pakeičia kompiuterio klaviatūrų mygtukus, ir liežuviu paliečiama panelė aktyvuojanti infraraudonąjį spindulį.

Galvos pelė naudoja infraraudonąjį spinduliavimą, norint paversti galvos judesius kursoriaus valdymu, pavaizduota 1.3.2.2 paveiksle. Pelės funkcijos gali būti kontroliuojamos delsimo virš mygtuko tam tikru laiko tarpu.



1.3.2.2 pav. Galvos pelė naudojanti infraraudonąjį spinduliavimą

Gali būti įrengtas išorinis jungiklis ar jungikliai, kurie sekioja pelės klavišus. Lyginant galvos pelę su šalmais, elektrodais, akiniais ir burnos lazdelėmis gali būti nepatogūs nešioti ar naudoti. Dažniausiai parduodami galvos įrenginiai dažnai yra nepritaikyti vaikų galvos dydžiui. Bet, kai kurie vartotojai, ypač maži vaikai, nemėgsta prisilietimų veido srityje, jaučia diskomfortą ir gali priešintis bet kurių įrenginių tvirtinimui prie galvos.

Yra daugybė įrangos, kuri naudoja kamerą, kuri seka kūno judesius, ypač galvos judesius. Kameros pelė pavaizduota 1.3.2.3 paveiksle, tai įranga, kuri užtikrina kompiuterio kontrolę be rankų, naudojant videokamerą sekancią kūno judesius (pvz., galvos) ir paversti šiuos judesius į kursoriaus judesius kompiuterio monitoriuje.



1.3.2.3 pav. Kameros pelė

Įrankių juosta ekrane leidžia vartotojui sekti visą pelės kontrolę. Kameros pelė veikia su visa standartine įranga ir nereikalauja laidų, infraraudonųjų spindulių. Naudojamas nosies galiuko sekimas, videokamera naudoja sekamo ypatumo judesį, taip tiesiogiai kontroliuoja pelės rodiklį kompiuteryje. Dabar „kamerų pelės“ sistemą sudaro du kompiuteriai, kompiuteris „vaizduotojas“

ir kompiuteris “vartotojas”. „Kompiuteris vaizduotojas“ atlieka vizualinio sekimo algoritmą ir siunčia apdorotus duomenis kompiuteriui vartotojui.

1.3.3. Elektrinė sąveika

Žmogaus-Kompiuterio [4, 5] Sąveikos (Human-Computer Interaction) sistema EMG/EEG (electromyographic/electroencephalographic) pavaizduota 1.3.3.1 paveiksle. EMG/EEG pagrindu yra alternatyvi akių sekimui sistema, ir kada ji yra sujungta su ekrano klaviatūra, ji visiškai valdoma be vartotojo inicijuojamų fizinių judesių. Sistema transformuoja biosignalus į dviejų dimensijų kursoriaus judesių valdymą. Priešingai negu akies sekimo sistemos, HCI sistema yra potencialiai pigesnė ir mažų gabaritų, ji panaikina kai kuriuos nepatogumus, sukeltus akių sekimo sistemų, tokius kaip “gyvenimo laikas”, vartotojo treniravimas ir kalibravimo praradimas.



1.3.3.1 pav. Kibernetinė sistema aptinka smegenų ir kūno signalus galvos sensoriuje

1.4. Prietaiso funkcijų panaudojimas

Žmonės [5, 6] su fizine negalia, kurių rankos paralyžuotos ar sužalotos nelaimingų atsitikimų metu, negali naudoti įprastų įvedimo prietaisų. Tai reiškia, kad jie negali valdyti prietaisų, turinčių daugiau nei vieną funkciją, tai yra negali išrinkti norimos funkcijos, nes tam reikia pozicionuoti ranką, pirštą ar pelės žymeklį virš norimo objekto ir jį aktyvuoti. Todėl kokios nors prietaiso funkcijos panaudojimą galima suskaidyti į dvi fazes: pozicionavimą ir aktyvavimą.

Pozicionavimas gali būti atliekamas dvejopai:

1. Tiesiogiai išrenkant norimą elementą (pirštu, vairalazde, pele ir t.t.)
2. Panaudojant elementų skenavimą.

Pozicionavimo ir aktyvavimo funkcijoms atlikti galimi šie būdai:

1. Panaudojant balsu komandas.
2. Panaudojant koki nors elementarų fizinį veiksma.
3. Panaudojant žvilgsnio kryptį.

Panaudojant balso komandas, pagrindinė problema, su kuria susiduriama naudojant šį būdą yra identifikacinių žmogaus balso parametrų išskyrimas, kuriems neturėtų įtakos vartotojo emocinė ar fizinė būklė.

Labiausiai paplitęs žmogaus judesio panaudojimas, kurį jis gali atlikti valingai. Tai gali būti rankos krustelėjimas, jungiklio spustelėjimas (ranka, koja, liežuviu), akies mirksnis ir kita. Svarbu, kad veiksmas turėtų 2 būsenas, atitinkančias padėtis „įjungta“ ir „išjungta“. Šių padėčių pagalba įmanoma atlikti ir tiesioginį elemento išrinkimą, ir panaudoti skenavimo sistemose.

Žvilgsnio krypties panaudojimą apžvelksime vėliau.

1.5. Valdymas žvilgsniu

Videookulografinis [10] metodas yra pagrįstas registruojamo akies vaizdo analize. Akių judesiams nustatyti naudojami įvairūs vaizdo požymiai: vyzdžio centro, ragenos atšvaito, 4-ojo Purkinje atvaizdo koordinatės, ribos tarp rainelės, odenos padėtis ir deformacija. Tikslūs metodai gaunami tada, kai derinami keli požymiai, pavyzdžiui, apskaičiuojamas vektorius tarp vyzdžio centro ir ragenos atšvaito. Tada matavimo metu nesvarbūs maži galvos poslinkiai; naudojant vieną kurį požymį ir dėl mažiausio galvos judesio gali būti didelės akių judesių matavimo paklaidos (Daunys, Laurutis, 1999, p. 76-79).

Jeigu tiriamojo galva eksperimento metu nejuda, tada geriausias yra vyzdžio centro sekimo metodas. Pranašumai: ryški vyzdžio kontūro riba nuo rainelės, darant kampinius posūkius, poslinkiai būna didesni negu ragenos atšvaito. Derinant vyzdžio centro ir ragenos atšvaito metodus galima kompensuoti mažus galvos poslinkius, tačiau prarandamas tikslumas, kai galva nejuda. Metodo paklaidos būna dėl šių pagrindinių trikdžių:

- akies mirksniai arba dalinis vyzdžio uždengimas akių vokais ir blakstienomis;
- vyzdžio skersmens kitimas nuo 2 iki 8 mm, veikiant psichologiniams ir fiziologiniams reiškiniams;
- pašalinių kryptingų šviesos šaltinių skleidžiama šviesa.

Vyzdys gali padėti matuojant akių judesius; čia remiamasi „raudonų akių“ efektu. Akis yra apšviečiama lygiagrečiais žiūros linijai spinduliais. Stebėtojai, esančiam toje pačioje žiūros linijoje, vyzdys atrodo šviečiantis. Apšvietus infraraudonaisiais spinduliais kampu, juodo vyzdžio kontrastiškumas labiau padidėja. Tiriamasis netrikdomas, be to, sumažėja aplinkinių šviesos šaltinių įtaka. Taip susidaro ragenos atšvaitas.

Ketvirtas Purkinje atvaizdas yra beveik to paties dydžio ir suformuluotas beveik toje pačioje plokštumoje kaip ir pirmasis atvaizdas. Kadangi šviesos lūžio rodiklis pasikeičia daug mažiau galinėje lęšiuko sienelėje nei nuo oro ir ragenos sąlyčio, tai ketvirtojo Purkinje atvaizdo

intensyvumas sudaro tik 1% pirmojo atvaizdo. Todėl sekti tokį atvaizdą yra sunku. Be to, reikia griežtai kontroliuoti aplinkos apšvietumą. Šiuo metu negalima registruoti mažos amplitudės akies judesių.

Viena iš problemų yra vyzdžio radimas įrašytame vaizde. Ebisawa (1998, p. 948-955) pasiūlė metodą, kaip lengviau rasti vyzdį. Panaudota savybė – apšvietus akį tiesiai, vyzdys pasidaro šviesus. Apšvietus šonine šviesa, vyzdys yra tamsus. Kaitaliojant apšvietimą, vyzdys paeiliui būna čia šviesus, čia tamsus, o kitos akies sritys nuo apšvietimo keičiasi ne taip žymiai. Skaičiuojant dviejų vaizdų skirtumą, vyzdžio vietoje pasiekiamas didžiausias kontrastas.

Tiriant operatoriaus okulomotorinę sistemą, kita problema yra galvos judesiai. Brangesnės žvilgsnio krypties matavimo sistemos naudoja judančius veidrodžius, kuriais akis taip pakreipiama, kad patektų į registruojamą vaizdą. Didelės skiriamosios gebos kameros leidžia išsiversti be judančių dalių (Zhu, Yang, 2002, p. 124-129). Rega gali būti registruojama iš lėto didinant žiūros gebą. Vidinis akies kampas yra pastoviausia dalis, pagal kurią galima registruoti galvos judesius. Pritaikius subpikselinį vaizdo apdorojimą kompiuteriu, galima pasiekti 1,4 laipsnio matavimo neapibrėžtį.

Ilgą laiką akių judesių matuokliai buvo gaminami tik pagal vienetinius užsakymus ir buvo brangūs. Šiuo metu atsiranda pigesnių matuoklių, susidedančių iš masiškai gaminamų dalių. Geriausia naudoti masiškai gaminamas apsaugos kameras ir vaizdo įvedimo į kompiuterį plokštes, kompiuteriams pritaikytas kameras arba buitines vaizdo kameras. Abu variantai turi kelis tuos pačius trūkumus. Labiausiai paplitęs vaizdo kadru skaičius per sekundę yra 30. toks kadru dažnis duoda per mažą laikinę skiriamąją gebą sakadoms tirti. Kitas trūkumas – kamerų optinė sistema. Ji pritaikyta filmuoti platų matymo lauką. Šiuo atžvilgiu ypač blogos yra kompiuterinės kameros, kurių aprėpiamo vaizdo keitimo galimybės labai ribotos. Trečias trūkumas – jose naudojami spalvoti vaizdo sensoriai, todėl nėra tokie jautrūs, kaip juodo ir balto vaizdo matricos.

Akių judesių naudojimas žmogaus ir kompiuterio sąsajose. Akių judesių tyrimai visuomet teikė naudingos informacijos ergonomikai. Paplitus kompiuteriams, iš ergonomikos išsirutuliojo atskira mokslo šaka – žmogaus ir kompiuterio sąveika. Tarp kitų klausimų ji nagrinėja informacijos įvedimo į kompiuterį būdus. Tradicinės kompiuterio dialoginio valdymo priemonės yra klaviatūra ir pelės įtaisas. Tačiau kompiuterių pajėgumas nuolat didėja, jie gali pateikti gerokai daugiau informacijos, negu operatorius gali įvesti. Akių judesius žmogus dažniausiai gali atlikti ir tada, kai jam yra sutrikęs judėjimas.

Šiuo metu žmogaus ir kompiuterio dialogas dažniausiai organizuojamas WIMP (akronimas nuo anglišku žodžių *windows, icons, menu, pointers*) sąsaja. Naudojantis šia sąsaja, kompiuterio valdymą galima atlikti šiais elementariais veiksmais:

- kursoriaus pozicionavimu (angl. *pointing*);

- komandos patvirtinimu.

Vienas pirmųjų apie akių judesių panaudojimą žmogaus ir kompiuterio sąsajos srityje pranešė R. A. Boltas iš Masačusetso technologijos instituto (Bolt, 1982, p. 360-362). Šiame institute buvo kuriama multimodalinė žmogaus ir kompiuterio sąsaja. Objekto pasirinkimas akimis ir rankų gestais. R. A. Boltas siūlė komandų patvirtinimą duoti balsu. Praktinis idėjų įgyvendinimas buvo išbandytas akių judesius matuojant videookulografiniu metodu.

1984 m. Virdžinijos universitete (JAV) buvo pradėtas įdiegti „Erica“ (*Eye-gaze-Response Interface Computer Aid*) projektas (Hutchinson, White, 1989, p. 1527-1533). Jo vykdymo metu buvo sukurta kompiuterio valdymo žvilgsniu sistema, skirta neįgaliesiems vartotojams. Akių judesių matavimui panaudota infraraudonųjų spindulių kamera, įtaisyta po monitoriumi. Šia kamera buvo galima matuoti akių judesius, panaudojant vyzdžio centro, gauto pagal jo šviesų atvaizdą, ir ragenos atšvaito koordinates. Kompiuterio žymiklis (kursorius) buvo perkeliamas akių judesiais. Komandos patvirtinimas gaunamas ilga fiksacija. Apie aptiktos fiksacijos pradžią duodamas ženklas garsiniu signalu ir perkeliant kursorių į žiūros tašką. Pasibaigus nustatytam fiksacijos laikui, duodamas kitas garsinis signalas, žymintis komandos pasirinkimą. Toks vizualinis ir girdimasis grįžtamasis ryšys suteikia galimybę atsisakyti pasirinktos komandos, jeigu nenuskambėjo antrasis garsinis signalas. 1988 m. pasirodė komerciniai „Erica“ egzemplioriai. Sistema leido pasinaudoti keturiomis funkcijomis: aplinkos valdymo, komunikacijos, rekreacijos, teksto skaitymo. Esminis komunikacijos funkcijos elementas yra teksto rinkimas. Jis buvo realizuotas panaudojant virtualią klaviatūrą kompiuterio ekrane ir WIMP sąsają. Ši su akių judesiais suderinta sąsaja negali užtikrinti greito teksto rinkimo, nes akuatorius yra tik akis, tuo tarpu naudojantis klaviatūra dalyvauja keli pirštai. Teksto rinkimas neįgaliajam vėliau tapo ta sritimi, kuriai skiriama bene daugiausia mokslinių tyrimų. „Erica“ sistema tobulinama iki šių dienų (Lankford, 2000, p. 23-27). Joje dabar naudojamos kitos akių judesių sąsajos naujovės, aprašytos toliau.

Reikšmingus akių judesių panaudojimo tyrimus atliko R. Jakobas. Jis, (1991, p. 152-169), būdamas JAV karinio jūsus laivyno mokslinis tyrėjas, vienas iš pirmųjų suformulavo šūkį „Į ką žiūri, yra tai, ką gauni“ (What You Look At is What You Get). R. Jakobas numatė sistemų, panaudojančių akių judesius, pranašumus ir trūkumus. Toks žymiklio perkėlimo būdas turėtų būti greitesnis. Paprastai, prieš perkeliant žymiklį ant ikonos, pirmiau į ją pažiūrima. Tuo tarpu visi mechaniniai prietaisai turi didesnę inerciją. Pagrindinė problema, dėl kurios negalimas greitas darbas, buvo pavadinta Mido prisilietimu (graikų mitologinis personažas Midas turėjo galią viską, prie ko tik prisilies, paversti auksu). Kur pažiūrės operatorius, ten bus sužadinta komanda. Norint to išvengti, reikia fiksuoti komandą. Naudojant pelės įtaisą, tai daroma klavišo paspaudimu. Yra keli akių žvilgsnio sąsajų naudojimo variantai. Vienas toks pat, kaip ir naudojant pelę, - klavišo paspaudimas. Tačiau šiuo metodu negali naudotis sunkūs neįgalieji. Kitas būdas – ilga fiksacija ant

pasirinktos ikonos (angliškai vadinama *dwell time*). Dar vienas variantas – patvirtinimą parodyti akies mirksniu.

1.5.1. Žvilgsnio krypties naudojimo metodai

Žvilgsnio [5, 6] krypties panaudojimo srityje dabar atliekama labai daug mokslinių tyrimų. Išradimai šioje srityje panaudojami netik neigaliųjų integravimui į visuomenę, bet ir karinėje pramonėje (taikinio pozicionavimas).

Pagal skenavimo pobūdį galima išskirti tokius metodus:

1. Autoskenavimas. Šiuo atveju elementai paeiliui išskiriami automatiškai, tam tikrą laiką laukiant vartotojo reakcijos. Po paskutinio elemento vėl išskiriamas pirmasis.
2. Žingsninis. Sekantis elementas išskiriamas po numatytos vartotojo reakcijos – vieno ar kelių signalų „įjungta“ suformavimų jungikliu. Tuo tarpu elemento išrinkimui naudojama kita jungiklio tipo įvedimo prietaiso aktyvavimo seka.
3. Bėgantis. Šiuo atveju elementai paeiliui išskiriami tol, kol jungiklis laikomas „paspaustas“. Vos tik jungiklis yra atleidžiamas, skenavimas sustoja ir išskirtas lieka tas elementas, ties kuriuo skenavimo procesas buvo nutrauktas.

Skenavimo metodus galima suskirstyti pagal skenuojamų elementų matricos lygių skaičių.

1. Vieno lygio. Jam priklauso tiesinis ir apskritiminis skenavimo metodai. Tiesiniame metode elementai yra išskiriami paeiliui, nors išdėstyti gali būti ir ne viena eilute. Apskritiminis – tai tas pats tiesinis metodas, tik elementai išdėstyti ratu.
2. Dviejų lygių. Tokio metodo pavyzdžiu galėtų būti eilutės – stulpelio skenavimas. Šiame metode, paeiliui yra išskiriama elementų matricos eilutėje (arba stulpelyje) esantys elementai. Kai kuris nors elementas išrenkamas, paeiliui išskiriami tos eilutės stulpelių (eilutės) elementai.
3. Kelių lygių. Vienas iš tokių yra grupės metodas. Grupės atveju, elementai saugomi daugiamatėje matricoje. Tai panašu į kompiuterio programos meniu, kuri turi kelis submeniu, tai yra į gylį besidriekiančią hierarchiją. Šis metodas gali naudoti vieną iš pirmų dviejų tipų, atskiro sluoksnio skenavimo matricai realizuoti.

Pagal realizaciją, skenavimo sistemas galima suskirstyti į 2 grupes:

1. Vartotojui matomas tik vienas elementas (tas, kuris šiuo metu yra išskirtas).
2. Vartotojui matoma elementų grupė, dažniausiai visa to lygio elementų matrica.

1.5.2. Akies pelės naudojimas

Žvilgsnio [5] panaudojimas akies pelei ilgai buvo žmogaus ir kompiuterio sąveikos tyrimų tikslas. Žvilgsnio panaudojimas ir privalumai turi potencialą, nes jis yra natūrali sąveikos forma, kadangi žmogus žiūri į objektą ir sąveikauja su juo. Aptinkant objektą, žvilgsnis gali būti labai greitas palyginti su kitais įrenginiais, tokiais kaip galvos pelė. Natūralūs akies judesiai greitai nepavargsta ir gali suteikti sąveiką beveik be nuovargio. Žvilgsnio valdymas, gali būti naudojamas kaip kompiuterio valdymo modalumas tiems žmonėms, kurie neturi kito naudotino sąveikos su kompiuteriu metodo.

1.5.3. Žvilgsnio sąveikos komunikacija

COGAIN [4, 5] (Žvilgsnio sąveikos komunikacija) jungia cutting-edge sąsajos technologijų ekspertizę siekiant naudoti vartotojams su negalia. COGAIN siekia pagerinti gyvenimo kokybę tų žmonių, kurie turi judėjimo kontrolės sutrikimų.

Pagalbinės COGAIN technologijos įgalins tikslinę grupę komunikuoti panaudojant turimus gebėjimus ir siūlant prastėjančių gebėjimų kompensavimą. Vartotojai galės naudotis prietaisais, kurie padės jiems valdyti aplinką, arba pasiekti visiškai naujo patogumo lygio ir žvilgsniu grindžiamų ryšių. Naudojant tinkle išvystytą technologiją, tekstas gali būti greitai kuriamas žvilgsniu, taip pat kaip ir balsu. Be to, tinklas suteiks pramogų pritaikymą, suteikiant vartotojų gyvenimui daugiau malonumo ir lygiateisiškumo. COGAIN teigia, kad pagalbinės technologijos geriausiai suteikia galimybes, kurios įgalina ir pramogauti. Šiame tyrime ir atliekamame darbe COGAIN siekia įveikti apsunkinančias kompiuterio priemones ir programinę įrangą. Dėl tyrimo veiksmų fragmentacijos ši programinė įranga dažnai veikia tik su atitinkamu akies sekimo prietaisu. Tas pats taikoma ir įrankiams, analizuojantiems efektyvumą ir praktiškumą, ir kurie yra sukurti tik atitinkamam įrenginiui. COGAIN veiks tik modularizacijai ir standartizacijai, kurie jungia esančias dalis į aplanką suderinamų ir tinkančių naudoti įrankių ir įtaisų. Praktiškumas, patogumas ir rezultatų priėmimas bus užtikrintas didelio vartotojų bendruomenių ištraukimo į projektą. Speciali veikla yra planuojama ir tam, kad vartotojai būtų įtraukti į projektą ir tam, kad rezultatai būtų skleidžiami tarp tų, kuriems šios technologijos yra reikalingos. COGAIN taip pat prisidės prie akies sekimo technologijų turimų įrenginių kainos mažinimo ir alternatyvų kasdieniam naudojimui namuose ir kitoje veikloje plėtojimo.

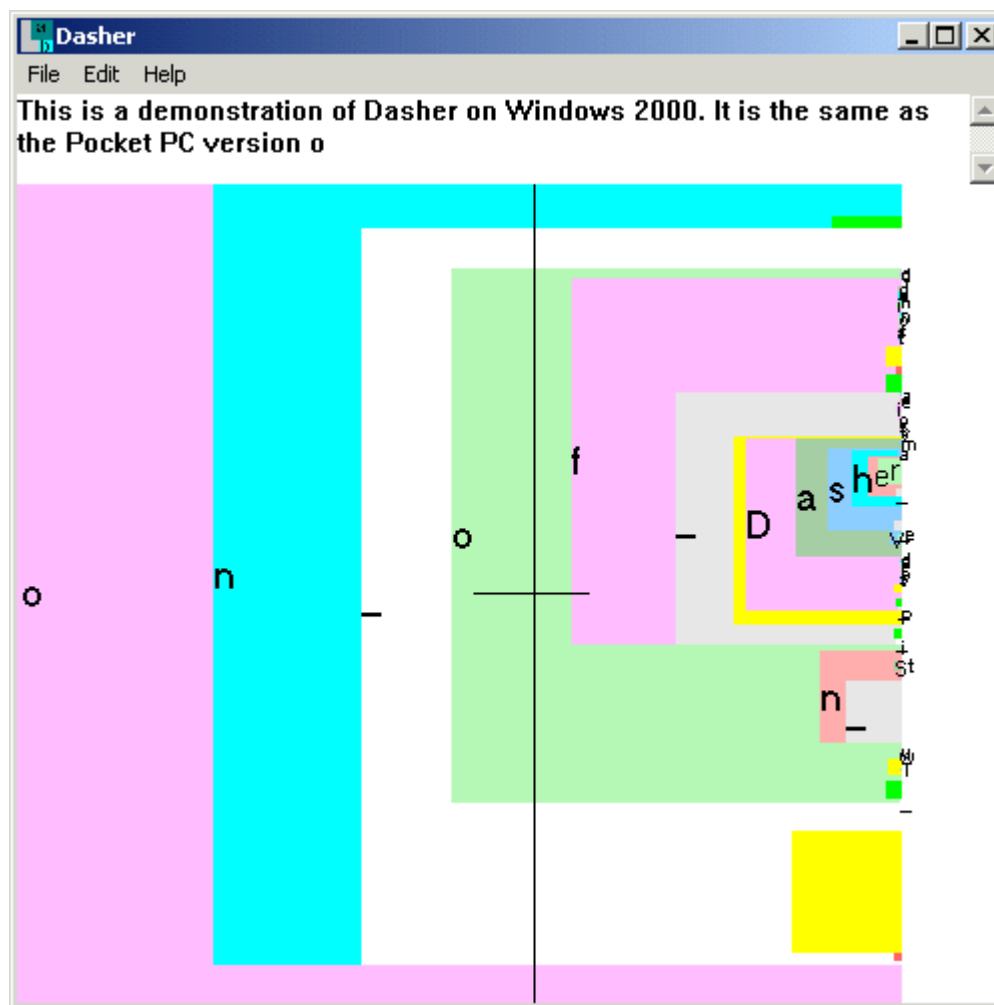
Galimybė pasakyti greitai ir efektyviai tikslia kalba yra fundamentalus gyvenimo kokybės rodiklis. Kai kurie žmonės su specialiaisiais poreikiais negali palaikyti sklandaus tarpasmeninio bendravimo.

1.6 Teksto rinkimo akimis sistemos

Vartotojams [5, 6], praktiškai neturintiems galimybės judėti, optimaliausia yra akies bliko metodu pagrįsta sistema ASL 4250R PLUS . Ji pagrįsta spindulio atspindžio kampu nuo akies lęšiuo. Prieš vartotoją įtvirtintas infraraudonųjų spindulių šaltinis spinduliuoja koncentruotą, siaurą spindulių pluoštelį, kuris atsispindėjęs nuo akies, patenka į registruojantį įrenginį. Pagal spindulio atspindžio kampą yra nustatoma tiksli vieta ekrane, į kurią žiūri vartotojas. Ši sistema pasižymi dideliu tikslumu, nes ji „atsižvelgia“ į paklaidą dėl žmogaus galvos nevalingo judesio. Šiuo principu veikiančios sistemos nelabai patogios, nes infraraudonųjų spindulių siūstuvai/imtuvai dažniausiai tvirtinamas ant akinių rėmo ir užstoja dalį matymo lauko. Todėl sukurtos televizinės žvilgsnio krypties nustatymo sistemos. Jose žvilgsnio kryptis fiksuojama televizinės kameros pagalba. Vieną tokių sistemų sukūrė Liuveno Karališkojo Universiteto mokslininkai . Žvilgsnio krypties nustatymui sistema naudoja vyzdžio centro metodą. Po kompiuterio monitoriumi tvirtinama videokamera pastoviai seka akį. Nedidelis, mažos galios IR spindulių diodas, įtaisytas šalia kameros, šviečia į akį. Jo spindulys nukreiptas į rageną. Iš kameros vaizdas perduodamas į personalinį kompiuterį, kur programiškai yra paskaičiuojamas skirtumas tarp taško, atsispindėjusio nuo ragenos ir vyzdžio centro. Akiniai ar kontaktinės lizės sistemai dirbti netrukdo. Didžiausias šios sistemos trūkumas yra tai, kad vartotojas turi laikyti galvą fiksuotoje padėtyje.

Yra sukurta sistemų, neturinčių šio trūkumo. Objekto, į kurį žiūri vartotojas, išrinkimas atliekamas, jei vartotojas laiko sufokusavęs ties tuo objektu žvilgsnį tam tikrą laiką (iki 0,5 s). Kitos sistemos analogiškam uždaviniui (išrinkimui) naudoja dantų tiko, akies mirksnio ar kitą registratorių.

COGAIN partneriai nepriklausomai išvystė penkias skirtingas teksto rinkimo akimis sistemas per kelis pastaruosius metus, o būtent „DASHER“, „GazeTalk“, „UKO-II“, „iKey“, „ECKEY“. Sistemos turėjo individualius privalumus ir trūkumus. Dasher yra greičiausia AAC sistema pavaizduota 1.6.1 paveiksle.



1.6.1. pav. Teksto rinkimo akimis sistema Dasher

Dasher naudoja algoritmo variantą, vadinamą PPM5D+, kuris gali suspausti Anglišką tekstą iki 2 bitų ženklui. Yra ir kiek geresnių algoritmų, tačiau PPM5D+ buvo pasirinktas, nes jis yra paprastas ir greitas. Kalbos modelis mokosi visą laiką: jei panaudojate naują žodį bent vieną kartą, jį lengviau parašyti kitą kartą.

1.6. 1. Bendrieji teksto įvedimo metodai

Teksto [5, 7] įvedimas panaudojant mažus prietaisus be klaviatūros ar naudojant žvilgsnį turi daug bendrų apribojimų. Nauji būdai, tokie kaip parodyta Dasher sistema atitinka abi šias kriterijų sritis.

Kembridžo Dasher projektas bendradarbiauja su ACE platinamas sistemą vartotojams su negalia.

Aišku, kad akies sekimo įranga turi pagelbėti kuo didesniai vartotojų skaičiui. Todėl turi būti užtikrinta galimybė naudotis įranga kuo daugiau įvairiomis kalbomis kalbantiems žmonėms. Pavyzdžiui, Dasher tinka daugiau kaip 80 kalbų.

Kembridžo universitetas sukūrė Dasher, greičiausiai tekstą renkantį žvilgsniu sistema kada nors aprašyta literatūroje, kurios spausdinimo greitis yra pasaulyje rekordinis - nuo 39 žodžių per minutę. Jų darbas buvo publikuojamas Nature (2002) straipsnyje, jis apibūdintas kaip vienas retų HCI tyrimo pavyzdžių, paminėtų šiame žurnale. Dasher grupė įdėjo dideles pastangas versdama sistemą į kelias kalbas. Dasher grupės žmonės gilinasi į akies sekimo algoritmų vystymą vidutinės ir mažos raiškos kameroms.

1.6.2. Dasher projektas

Dasher [5, 7] projektas buvo pradėtas Inferenc grupės Cavendish laboratorijoje. Projektui vadovavo IT kompanijos įkūrėjas, Fizikos Departamento lektorius Deivid Makkey (David MacKay). Deivid 1997 m. sukūrė pirmą Dasher prototipą.

Deivid Vard (David Ward) nuo 1998 iki 2002 m. kūrė ir gerino Dasher versijas. Rašydamas savo disertaciją, jis Dasher „apvilko“ programiniu apvalkalu, pastoviai ją tobulino, vykdė daugybę eksperimentų, kurie pademonstravo visas programos galimybes. Šiuo metu Deivid dirba Kembridžo universitete.

Alan Blekvel (Alan Blackwell), kompiuterinės laboratorijos lektorius padėjo vykdyti eksperimentus.

Darbai grupėje taip pat buvo pakviesti ir kiti tyrėjai, kurie manė kad Dasher būtinas visam pasauliui. Kuriant Dasher ypatingas dėmesys buvo skirtas tam, kad juo galėtų naudotis nedarbingi vartotojai. Be viso šito, dar buvo skiriamas dėmesys tokiems tikslams kaip nešiojamų kompiuterių sąsajos, japoniškos programos versijos sukūrimas, specializuotų versijų sukūrimas specifiniams kompiuteriams.

Išbandant Dasher sistemą kompanija Xybernaut suteikė galimybę naudotis jų nešiojamais kompiuteriais. Pradžioje buvo kuriamos versijos, kurios palaikytų skandinavų šalių, vokiečių ir prancūzų kalbą.

Taip pat Fil Kauens (Phil Cowans) ir Tim Hospedales (Tim Hospedales) kūrė akių pelę (eyetracker). Su Linux versija dirbo Timas Xospedales (Tim Hospedales). Fil Kauens (Phil Cowans) taip pat dirbo su savireguliuojančia akių pele (eyetracker). Ayn Merrei (Iain Murray) kūrė kalbos modelius.

Yra kuriamas įtaisas kuris sugebėtų atpažinti kalbą, sekti kaip teisingai sistema nuspėja galutinį rezultatą. Esant reikalui, kai sistema neteisingai supranta kalbą, Dasher pagalba atlikti sakinių korekciją. Tai žymiai paprasčiau nei taisyti klaidas balso komandomis.

Taip pat yra kuriama programa, kuri leistų Dasher programai dirbti ir kaip automatinė vertimo programa. Tai yra kai paprasta vertimo sistema, kuri nekokybiškai verčia iš vienos kalbos į kitą

būtų kontroliuojama vartotojo. Jis išverčia tekstą ir su Dasher pagalba, kuris turi įtaisytą kalbos modelį, ištaiso klaidas.

Dasher versijos:

Tcl – originalo prototipas. Jis parodo, kaip atrodo pradinis kodas. Kūrėjas - Deivid Makkey (David MacKay).

C ir tcl – skirta Linux ir Windows operacinėms sistemoms. Kaip kalbos modelį naudoja ppm. Valdoma pele. Kūrėjas - Deivid Vard (David Ward).

C – skirta kišeniniams personaliniams kompiuteriams. Valdymas taškuojant. Kūrėjas - Deivid Vard (David Ward).

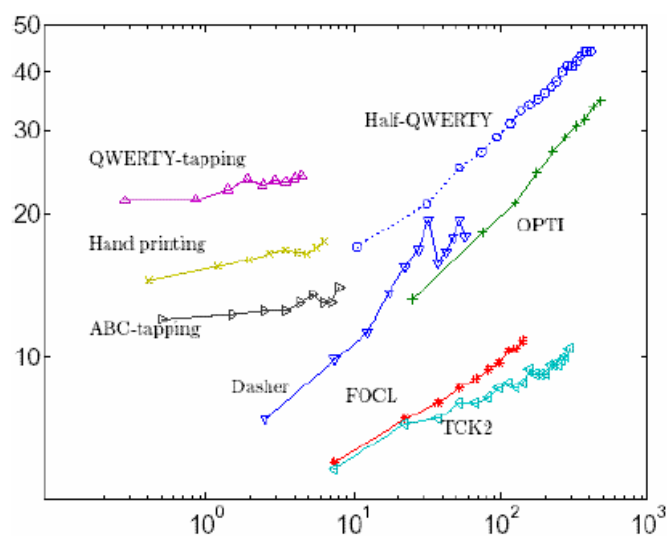
Eyetracker Dasher – valdoma pele arba akių sekimo įrenginiu. Kūrėjas - Deivid Vard (David Ward).

JDasher – palaiko japonų kalbą. Įeina C ir tcl versijos.

Java - pritaikyta Java kalbai. Kūrėjas – Džon Vin (John Winn).

1.6.3. Kaip dirba Dasher?

Dasher [5, 8] tai visiškai nauja sąsaja, sujungianti kalbos modeliavimą ir supaprastintą, dviejų dimensijų nepertraukiamą valdymą. Bandymai parodė, kad su Dasher pagalba vartotojai gali tekstą spausdinti iki 39 žodžių per minutę 1.6.3.1 pav. (su įprasta mums klaviatūra teksto rinkimo greitis siekia iki 60 žodžių per minutę). Dasher programa yra pritaikyta rašyti trumpus tekstus, žinutes, susirašinėti elektroninio pašto pagalba.

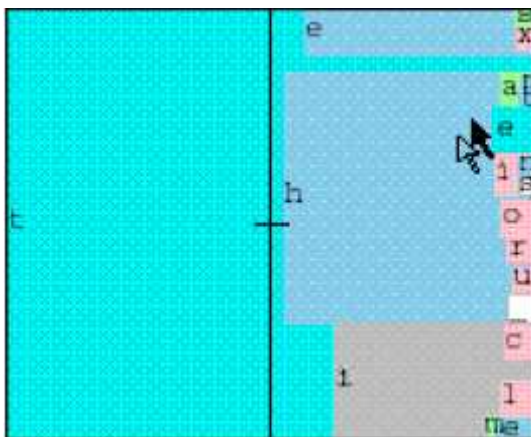


1.6.3.1 pav. Spausdinimas įvairiomis programomis ir įvedimo įrenginiais. Vertikaliai – spausdinimo greitis (žodžiai per min.). Horizontaliai – laikas panaudotas įvairioms sąsajoms

Kaip rašo Financial Times, Dasher programa panaši į arkadinį žaidimą „abėcėlės – žmogžudės ataka“. Dasher sąsaja grindžiama didinimu. Mes paprasčiausiai parodome į tai kas mums yra reikalinga, ir ta sritis didėja. Ekraną vieta, su kurios pagalba renkame tekstą, susideda iš raidžių ir ženklų, kurie geriausiai tinka pagal prasmę renkamam tekstui. Kuo ilgiau „žaidžiame“ su tekstu, tuo gaunamas ilgesnis tekstas. O vieta, kur yra „žaidžiama“ apsprendžia renkamo teksto prasmę.

Norint padaryti sąsają dar efektyvesnę, Dasher panaudojo specialų kalbos modelį, kuris leidžia dalinai atspėti vienos ar kitos raidės pasirodymo tekste tikimybę. Didžiausią tikimybę turintys raidžių junginiai yra „siūlomi“ iš karto. Tai leidžia juos labai greitai rinkti. Reti raidžių junginiai (arba tie junginiai, kurie yra skaitomi kaip klaidingi) užima mažai ekrano vietos ir juos rinkti yra žymiai sunkiau. Kalbos modelis visą laiką pats apsimoko. Jei mes vieną kartą surinkome kurį nors neįprastą žodį, tai kitą kartą šį žodį surinkti bus žymiai lengviau.

Raidės išrinkimas vyksta stačiakampio srityje, kurioje juda kursorius. Taigi, jei kursorius juda pavyzdžiui link „t“ stačiakampio, tai atsiranda naujas stačiakampis, ant kurio dešinėje pusėje vertikaliai yra išsidėsčiusios raidės, kurios siūlo surinkti raidžių junginį „ta“, „tb“, „tc“, ..., „tz“. „Siūlomų“ raidžių stačiakampių aukščiau parodo tų raidžių tikimybę būti surinktai po prieš tai esančios raidės, ženklo ar viso jų junginio. Anglų kalboje raidžių junginys „ti“ yra pakankamai dažnai vartojamas. „te“ – labia retai, o „th“ labai dažnai. Pagal raidžių pasirodymo tikimybes atitinkamas yra ir stačiakampių dydis, kuriuose yra tos raidės, pavaizduota 1.6.3.2 paveiksle

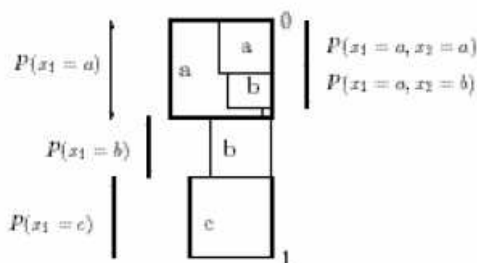


1.6.3.2 pav. Dasher programa pagal raidžių pasirodymo tikimybes pasiūlo sekantį ženklą

Bet didžiausias Dasher privalumas prieš kitas sąsajas dirbančias spėjimo metodu yra tai, kad Dasher neturi režimų. Ji yra berežimė ir vartotojui nereikia persijunginėti tarp režimų „rinkimas“ ir „spėjimas“. Kitas privalumas – rašymo bet kuriuo stiliumi apmokymo paprastumas. Reikia tik į reikalingą bylą įrašyti pavyzdžius ir viskas.

1.6.4. Kaip tikimybių modelis sąlygoja ženklų išsidėstymo schemą (išplanavimą)

Duotame [5, 8] kontekste parodyta galimų tęsinių abėcėlė, išreikštą kaip ženklų stulpelis (parodyta 1.6.4.1pav.). Dešinysis vertikalusis sektorius yra analogiškas aritmetiniam kodavimui.



1.6.4.1 pav. Aritmetinis alfabeto kodavimas raidėms {a, b, c}

Tegul alfabetas bus $Ax = \{a_1, a_2, a_1\}$. Intervalą suskirstome į I intervalus, kurių ilgiai lygūs tikimybėms $P(x_1=a_i)$. Papildomai suskirstome intervalą a_i į intervalus pažymėtus $a_ia_1, a_ia_2, a_ia_3, \dots, a_ia_l$. Tada intervalo a_ia_j ilgis yra:

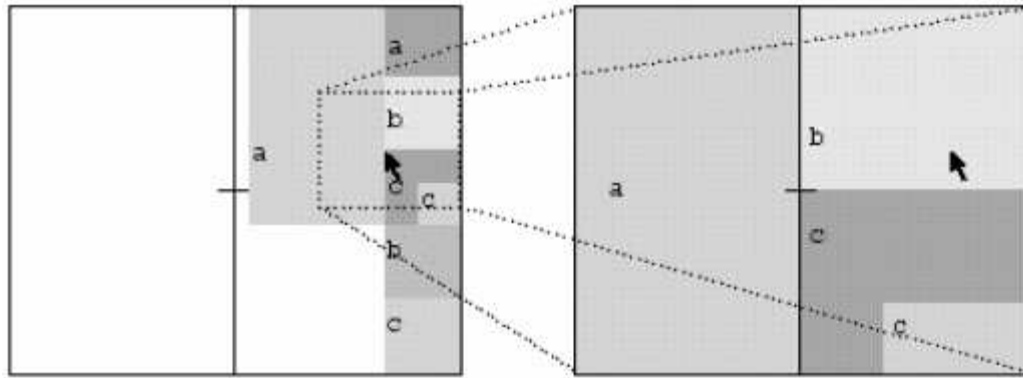
$$P(x_1 = a_i, x_1 = a_j) = P(x_1 = a_i)P(x_2 = a_j|x_1 = a_i). \quad (1.6.4.1.).$$

Kalbos modelis aprašytas 1.6.4.1. parodo tikimybes. Intervalas $[0, 1)$ gali būti suskirstytas į intervalų seką, koresponduojančią $x_1x_2\dots x_n$, tokiu būdu, kad intervalo ilgis yra lygus sekos tikimybei, pateiktai mūsų nagrinėjamam modeliui.

Intervalų seka, koresponduoja su alfabetiškai surikiuota knygų seka Burges „Babel Biblioteka“. Kiekvienas knygos dydis yra proporcingas jos turinio tikimybei, kaip kalbos modeliui. Vartotojas rašo slysdamas į biblioteką ir pasirinkdamas norimą knygą.

1.6.4.1 paveiksle stačiakampiai parodyti kaip kvadratai, tačiau programoje naudojami stačiakampiai su skirtingais sekos aspektais.

1.6.4.2 paveiksle parodyta sąsajos dinamika. Turima omenyje, kad vartotojas laiko kursorių pastovioje ekrano padėtyje. Po to kai S rėmelis atsinaujina, taikinio taškas padidėja taip, kad prieš tai buvęs „po pele“ taškas atsiduria ties ekrano viduryje esančiu kryželiu. S šiuo atveju galima vadinti žingsnio parametru. Taikinio sekimo įrenginys valdo sekamo taško tęstinį didėjimą. Stačiakampis padidėja taip, kad vieta po kursoriu kerta vidurio tašką (kryželį). Po to S rėmelis atsinaujina (1.6.4.2. pav.). Todėl S vadiname žingsnio parametru.



1.6.4.2 pav. Sąsajos dinamika

1.6.4.1. Kalbos modelis

Pasirinkdami [5, 8, 9] kalbos modelį Dasher programai buvo atsižvelgta į pagrindines dvi savybes:

- Kaip gerai kalbos modelis suspaudžia tekstą;
- Laikas, sugaištas skaičiuojant ženklo tikimybę.

Dasher turi skaičiuoti daugybę tikimybių kiekvieną kartą, kai tik vaizdas atsinaujina. Dasher versijose 1.0 – 1.6 kalbos modelis yra grindžiamas populiaraus teksto suspaudimo algoritmu, pavadintu PPM (prognoze pagal dalinį atitikimą). PPM yra kontekstu grindžiamas algoritmas; Jis naudoja ankstesnius ženklus, kad nuspėtų sekantį ženklą. Maksimalus konteksto dydis yra vadinamas modelio seka. Programoje naudojamas algoritmo variantas, pavadintas PPMD5, kuris yra 5 seka ir gali suspausti didžią angliško teksto dalį iki 2 bitų vienam ženklui. Yra geresnių algoritmų, tačiau PPMD5 yra paprastas ir greitas. Turėdami kontekstą galima suskaičiuoti visų alfabeto simbolių tikimybes.

Naudojant Dasher, sunku pasirinkti ženklą, kurio tikimybė yra labai maža. Todėl buvo pridėta maža fiksuota tikimybė δ prie bet kurio ženklo tikimybės ir po to tikimybės yra renormalizuojamos. Dasher 1.6 versijoje δ numatyta vertė yra 0,002. Kada vartotojas įveda tekstą, ženklai gali būti gražinti į PPM algoritmą. Vadinasi pritaikomos būsimos tikimybės vartotojo žodynui.

1.6.4.2. PPM teksto suspaudimo algoritmas

Penkiolika [5, 8] metų buvo kuriamas ir tobulinamas PPM teksto suspaudimo algoritmas. Pirmą kartą originalus logaritmas buvo publikuotas Cleary and Witten. Keli algoritmo patobulinimai buvo aprašyti Moffato, kuris įdiegė PPMC bandomąją versiją. PPMc pasiekia geresnių rezultatų, lyginant su kitais teksto suspaudimo metodais ir siekia juos ir toliau tobulinti ir gerinti.

Baigtinio konteksto modelis naudoja kelių simbolių spėjimą, kad įvertintų sekančio simbolio tikimybę. Yra pageidautina turėti ilgą kontekstą, kad būtų numatyta kiek įmanoma daugiau tinkamos informacijos, atliekant spėjimus. Tačiau naudojant ilgą kontekstą susiduriama su tuo, kad jie dažnai sutinkami novel data set (pradiniame įvedimo tekste), o toliau treniruojamame tekste jie nebesutinkami.

PPM yra baigtinio konteksto statistinio modeliavimo metodas, kuris sujungia kartu kelis fiksuotų sekų modelius, kad atspėtų sekantį simbolį. Konteksto ilgis paprastai yra fiksuotas, nors pastarasis PPM modelis Bloom naudoja adaptyvaus ilgio kontekstus.

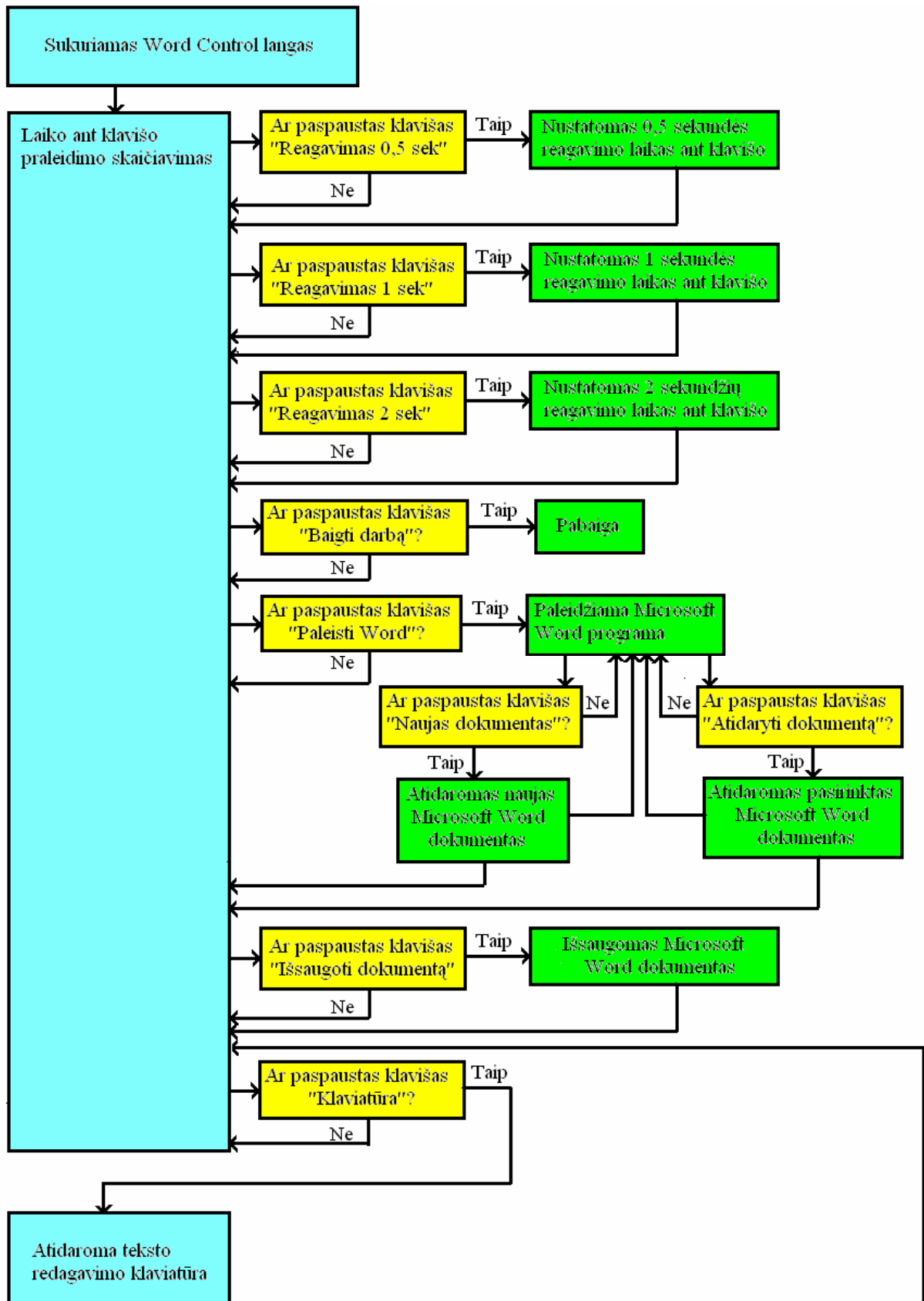
1.6.5. Dasher koncepcija

Dasher [5, 7] koncepcija praktiškai dirba su visomis kalbomis. Šiuo metu Dasher palaiko kai kurias europiečių kalbas bei japonų kalbą. Tam kad panaudoti Dasher su skirtingomis kalbomis, sistemą reikia apmokyti tų kalbų įkeliant aplanką **source** arba **source.txt** į bylą **input/** su mums reikalingos kalbos komponentais. Tik reikia įsitikinti, kad „Word“ pasirinkimas yra negalimas, arba paprasčiausiai aplanką **input/dict** pakeisti į kitą, su mums reikalingos kalbos žodynu.

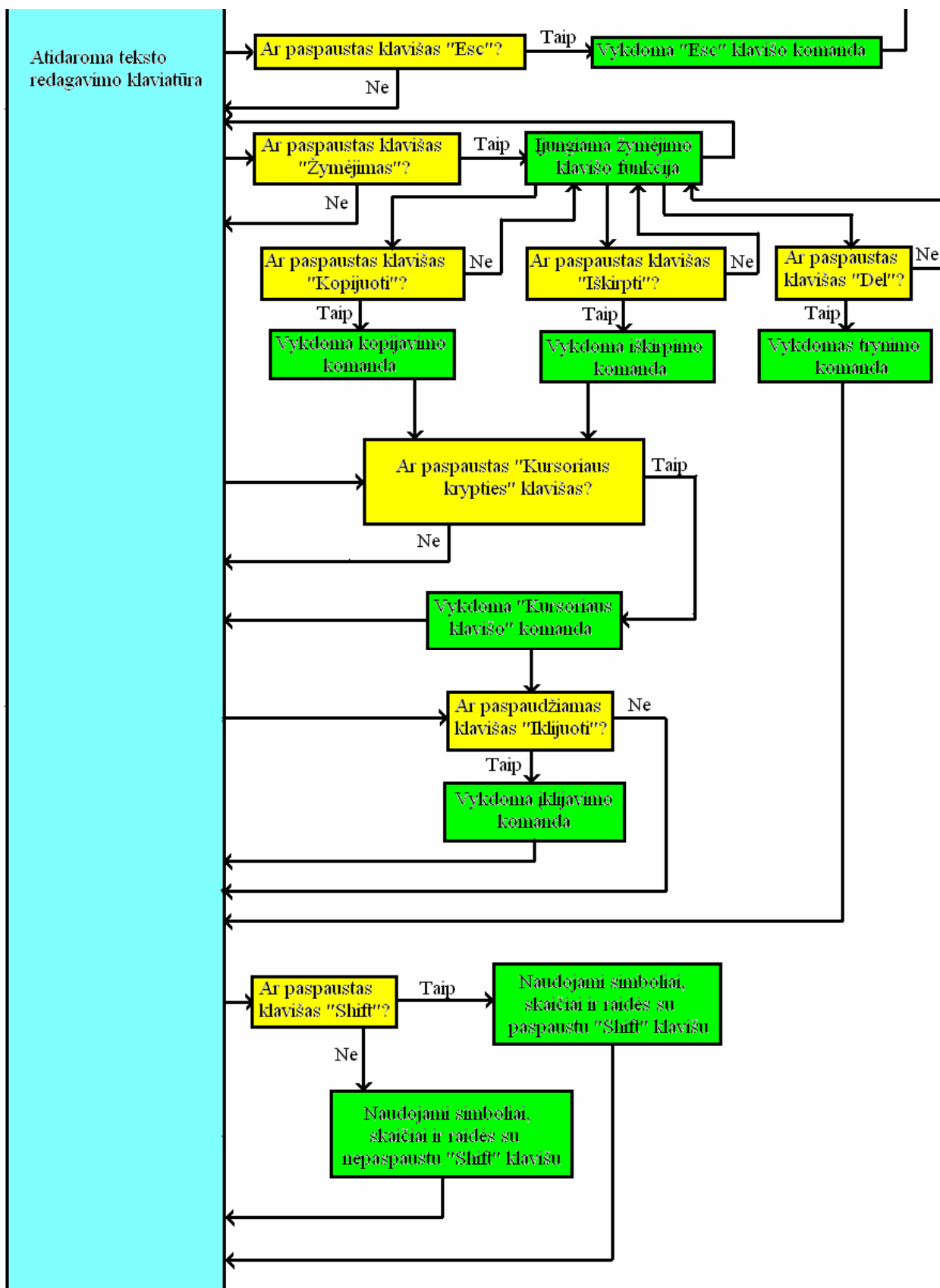
Su trečiąja versija yra padidinamas kalbų kiekis, kurias palaiko Dasher. Tai įgalino padaryti pasitelkus Open Source community. Trečiojoje kaip ir 1.6 versijoje kiekvienos kalbos palaikymui yra reikalingas specialus failas, kuriame yra reikalingos kalbos komponentai (tai apie 300 kilobaitų).

Dasher japoniška versija vadinasi DAISHOYA . Lietuviškai tai reikštu „raštininkas“. Pirmus žingsnius kuriant pilnavertę japonišką versiją, kuri palaiko ir Kana ir Kanji žengė David Ward. Jis sukūrė versiją, kuri palaiko Hiragana versijoje Dasher 1.6.3. Vėlesnės Dasher versijos nei 1.6.8 dirbant su Windows nepalaiko Hiragana. Tai yra todėl, kad atsiranda problemos su šriftais. Bet 1.6.8 Dasher versija pritaikyta darbui su Linux palaiko Hiragana.

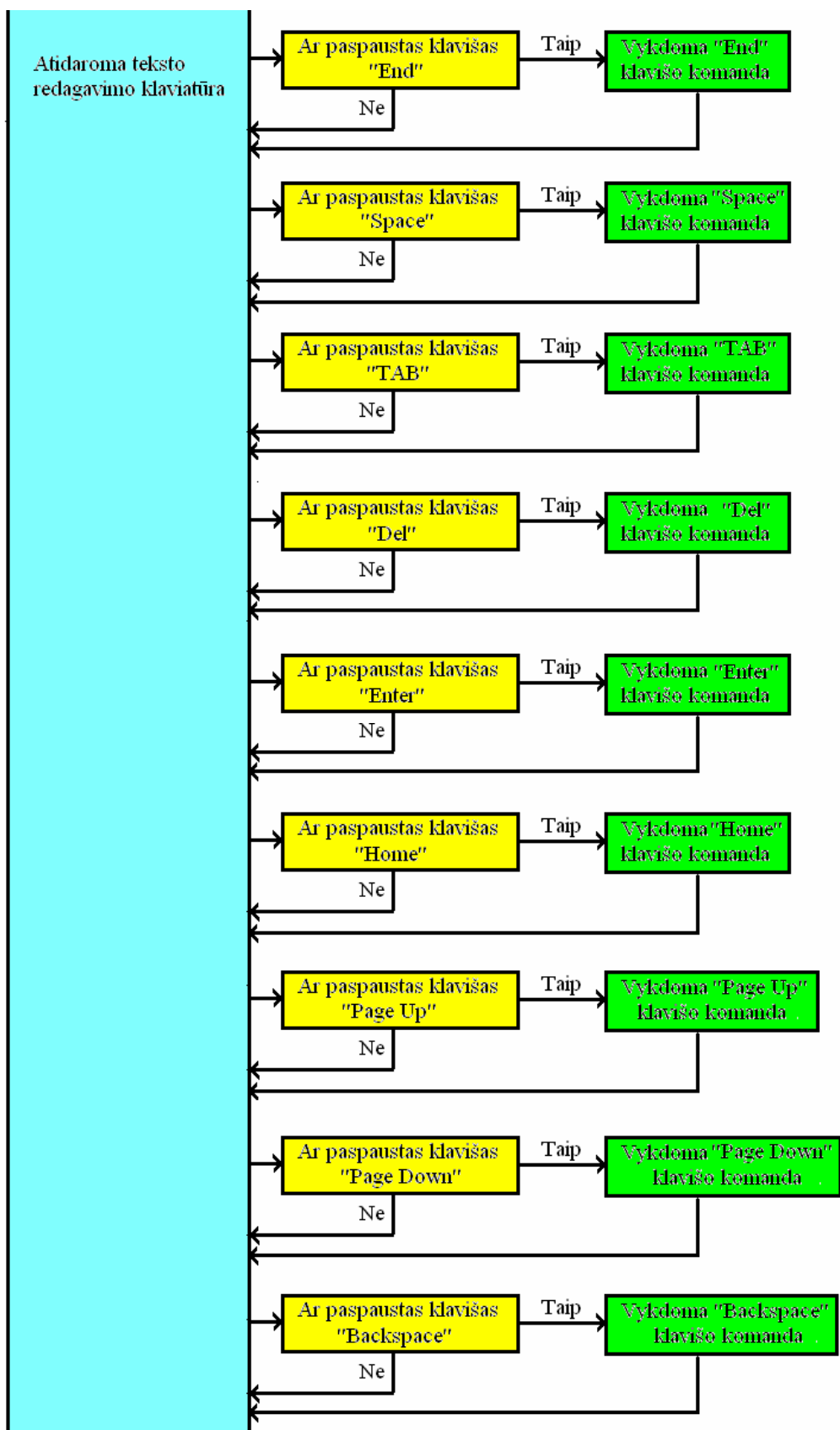
2. SUKURTOS TEKSTO REDAGAVIMO PROGRAMOS ALGORITMAS



2.1 pav. Laiko ant klavišo praleidimo skaičiavimo algoritmas



2.2 pav. Teksto redagavimo klaviatūros algoritmas



2.3 pav. Teksto redagavimo klaviatūros algoritmas (tęsinys)

Programa WordControl.exe sudaryta iš 3 formų – pagrindinė forma yra „frmMain“, dokumentų atidarymui pasirinkimo forma - „frmOpenDoc“ ir teksto redagavimo klaviatūros forma - „frmKeyboard“.

Paleidus programą, sukuriamas „WordControl“ langas (frmMain forma), kuriame išdėstyti su Microsoft Word programa bendraujantys klavišai. Pagrindinė forma gali atlikti šiuos veiksmus:

- Naujo Microsoft Word programos lango atidarymas;
- Naujo Word tipo dokumento sukūrimas;
- Iškviešti dokumentų atidarymo formą ir atidaryti pasirinktą dokumentą, esantį „c:\docs“ kataloge;
- Keisti programos reagavimo laiką į pasirinkto klavišo paspaudimo imitavimą;
- Iškviešti teksto redagavimo klaviatūros formą ir leisti vartotojui valdyti Microsoft Word programą tos klaviatūros pagalba;
- Išsaugoti atidarytą dokumentą Microsoft Word 2000-2003 dokumento formatu;
- Baigti darbą su programa.

Forma „frmMain“ turi TTimer komponentą „CheckTimer“, kuris pagal nustatytą laiko tarpą vykdo veiksmus, aprašytus TTimer metode „OnTimer“. Vartotojas, judindamas kursorių, atsistoja ant formos klavišo. Tuomet suveikia klavišo įvykis „OnMouseEnter“ ir programa perduoda to klavišo vardą kintamajam „btnName“ bei yra paleidžiamas „CheckTimer“ laikmatis, jei po atitinkamo intervalo, kurį gali nusistatyti vartotojas klavišais „Reagavimas 0,5 sek“, „Reagavimas 1 sek“ bei „Reagavimas 2 sek“, įvykdomos komandos aprašytos metodo procedūroje „OnTimer“ ir imituojamas atitinkamo klavišo paspaudimas. Jei vartotojas patraukia kursoriaus žymeklį nuo klavišo, yra sustabdomas „CheckTimer“ veikimas ir klavišo paspaudimas nebus imituojamas. Naudojame šią funkciją C++ programoje:

```
void __fastcall TfrmMain::CheckTimerTimer(TObject *Sender)
{
if (btnName != "") {
TButton* MyBtn;
MyBtn = (TButton*)(FindComponent(btnName));
MyBtn->Click();
}
}
```


Norint pradėti dirbti su Microsoft Word programa sukurtame „Word Control“ lange, užiname ant klavišo „Paleisti Word“. Vartotojas, judindamas kursorių, atsistoja ant formos klavišo „Paleisti Word“. Tuomet suveikia klavišo įvykis „OnMouseEnter“, vykdoma procedūra „btnStartWordMouseEnter“. Tada programa perduoda to klavišo vardą „btnNewDoc“ kintamajam „btnName“ ir paleidžiamas „CheckTimer“ laikmatis. „CheckTimer“ laikmačio intervalai: 500ms, 1000ms ir 2000ms. Laikmatis įjungtas, tada vykdoma procedūra „CheckTimerTimer“ ir, jei nurodytas klavišo vardas `MyBtn = (TButton*)(FindComponent(btnName))`, susikuria naujas klavišas ir įgauna klavišo savybes. Tada „MyBtn->Click()“ imituojamas tų abiejų (sukurto virtualioje erdvėje ir matomas klaviatūroje) klavišų paspaudimas. Tik vartotojui patraukus kursorių nuo klavišo „Paleisti Word“, suveikia klavišo įvykis „OnMouseLeave“, o vykdoma procedūra „btnStartWordMouseLeave“ išjungia laikmatį „CheckTimer“ ir reikšmė „btnName“ prilyginama tuščiai „btnName = ""“.

```
void __fastcall TfrmMain::Button1Click(TObject *Sender)
{
WordApp->Connect();
WordApp->Resize(800, 400) ;
WordApp->set_Visible(1) ;
frmMain->Left=300;
frmMain->Top=700;
frmKeyboard->Left = 50;
frmKeyboard->Top = 620;
}

void __fastcall TfrmMain::btnStartWordMouseEnter(TObject *Sender)
{
btnName = "btnStartWord";
CheckTimer->Enabled = true;
}

void __fastcall TfrmMain::btnStartWordMouseLeave(TObject *Sender)
{
btnName = "";
CheckTimer->Enabled = false;
}
```

Paleidus programą, galima sukurtame „Word Control“ lange užėti ant pasirinkto „Naujas dokumentas“ arba „Atidaryti dokumentą“ klavišo. Pasirinkus klavišą „Naujas dokumentas“, atidaromas naujas Microsoft Word dokumentas, o pasirinkus klavišą „Atidaryti dokumentą“ -

atidaroma „frmOpenDoc“ forma. Jos atidarymo metu yra išvalomas Microsoft Word dokumentų sąrašas, esantis „DocList“, ir užpildomas nauju sąrašu, kuris sugeneruojamas, suradus visus, „c:\docs\“ kataloge esančius „*.doc“, plėtinio failus. Vartotojui užėjus ant „DocList“ dokumentų sąrašo, paleidžiamas „ListTimer“ laikmatis, kuris, kas parinktą laiko tarpą (pagrindinėje formoje jis nustatomas mygtukais „Reagavimas 0,5 sek“, „Reagavimas 1 sek“ bei „Reagavimas 2 sek“), keičia sąrašo eilutės žymėjimą iš viršaus į apačią, tokiu būdu leisdamas vartotojui pasirinkti reikiamą dokumentą. Vartotojui patraukus kursoriaus žymeklį nuo „DocList“ sąrašo, laikmatis „ListTimer“ sustabdomas ir lieka pažymėtas dokumentas, kurį vartotojas norėtų atidaryti. Užėjus ant klavišo „Atidaryti“ arba „Atšaukti“ laikmatį „CheckTimer“, perduodamas atitinkamo mygtuko vardas ir, jei vartotojas nepatraukia kursoriaus žymeklio nuo klavišo, sužadinamas „CheckTimer“ įvykis po atitinkamo laiko intervalo ir imituojamas klavišo paspaudimas. Jei pasirenkamas klavišas „Atidaryti“, uždaroma „frmOpenDoc“ forma ir atidaromas pasirinktas Microsoft Word dokumentas. Atidaro „frmOpenDoc“ formą ir užpildo Microsoft Word dokumentų sąrašą esantį c:\docs\ direktorijoje:

```

void __fastcall TfrmOpenDoc::FormShow(TObject *Sender)
{
HANDLE hSearch;
WIN32_FIND_DATA FileData;
DocList->Items->Clear();
hSearch = FindFirstFile("c:\\docs\\*.doc", &FileData);
TListItem *li = DocList->Items->Add();
    li->ImageIndex = 1;
    li->Caption = FileData.cFileName;
if (hSearch) {
    li = DocList->Items->Add();
    li->ImageIndex = 1;
    li->Caption = FileData.cFileName;
    }
while (FindNextFile(hSearch, &FileData))
    {
    li = DocList->Items->Add();
    li->ImageIndex = 1;
    li->Caption = FileData.cFileName;
    }
}

```

Pagal nustatytą laiko intervalą dokumentų sąrašė juda žymeklis:

```
void __fastcall TfrmOpenDoc::ListTimerTimer(TObject *Sender)
{
    DocList->ItemIndex = listpos;
    listpos ++;
if (listpos == DocList->Items->Count) {
    listpos = 0;
    }
}
```

Atidaromas iš sąrašo pasirinktas Microsoft Word dokumentas:

```
void __fastcall TfrmOpenDoc::Button1Click(TObject *Sender)
{
    OleVariant docname;
if (DocList->ItemIndex>=0) {
    docname= "c:\\docs\\"+ DocList->Items->Item[DocList->ItemIndex]->Caption;
        frmMain->Doc->ConnectTo(frmMain->WordApp->Documents->Open(docname,
EmptyParam, EmptyParam, EmptyParam,
                EmptyParam, EmptyParam, EmptyParam, EmptyParam,
                EmptyParam, EmptyParam));
    frmMain->filename = docname;
    }
    frmOpenDoc->Close();
}
```

Kai kursorius užaina ant formos kraštų, nebejuda pasirenkamų dokumentų žymeklis:

```
void __fastcall TfrmOpenDoc::DocListMouseLeave(TObject *Sender)
{
    ListTimer->Enabled = false;
}
```

Atidarius naują arba pasirinktą Microsoft Word dokumentą, klavišo „Klaviatūra“ pagalba galima įsijungti redagavimo klaviatūrą, kuri bus aptarta vėliau. Redagavus tekstą, galima išsaugoti Microsoft Word dokumentą - tereikia užėti ant klavišo „Išsaugoti dokumentą“ pasirinktą laiko intervalą ir automatiškai suveikia „Save as“ komanda. Norint išsaugoti Microsoft Word dokumentą, reikia naudoti šią funkciją C++ programoje:

```
void __fastcall TfrmMain::btnSaveDocClick(TObject *Sender)
{
```

```
OleVariant par1, par2;
par1 = true;
par2 = wdWordDocument;
Doc->SaveAs((OleVariant)filename);
}
```

„OleVariant“ kintamojo tipas per kurį galime perduoti įvairių tipų reikšmes programai Microsoft Word.

„frmKeyboard“ forma savo išvaizda primena standartinę asmeninio kompiuterio klaviatūrą. Joje išdėstyti pagrindiniai klavišai, leidžiantys įvesti tekstą į Microsoft Word programą, žymėti teksto dalį pasirinkus klavišą „Žymėjimas“, kopijuoti pažymėtą tekstą į laikinąją atmintį („Clipboard“) pasirinkus klavišą „Kopijuoti“, įklijuoti nukopijuotą arba iškirptą tekstą, pasirinkus klavišą „Įklijuoti“, valdyti kursoriaus poziciją Microsoft Word programos dokumento lange „Kursoriaus krypties“ klavišų pagalba. Vartotojui užvedus kursoriaus žymeklį ant pasirinkto klavišo, kintamajam „btnName“ yra perduodamas to klavišo vardas ir įjungiamas laikmačio „CheckTimer“ veikimas. Jei po atitinkamo laiko tarpo (kuris nustatomas pagrindinėje „frmMain“ formoje klavišais „Reagavimas 0,5 sek“, „Reagavimas 1 sek“ bei „Reagavimas 2 sek“) vartotojas nepatraukia kursoriaus žymeklio nuo to klavišo, yra sužadinamas „CheckTimer“ įvykis „OnTimer“ ir imituojamas klavišo paspaudimas. Jei vartotojas patraukia kursoriaus žymeklį nuo to klavišo, „CheckTimer“ laikmatis yra sustabdomas. Jei vartotojas įvykdė „Shift“ klavišo paspaudimą, tada visų simbolių mygtukų reikšmės keičiasi, t.y. vietoje „a“ bus įterpiama „A“, o vietoje „[“ bus - „{“ ir t.t.

Atliekama kopijavimo komanda:

```
void __fastcall TfrmKeyboard::btnCopyClick(TObject *Sender)
{
frmMain->Doc->Range(rangestart, rangeend)->Copy();
}
```

Atliekama iškirpimo komanda:

```
void __fastcall TfrmKeyboard::btnCutClick(TObject *Sender)
{
frmMain->Doc->Range(rangestart, rangeend)->Cut();
}
```

Atliekama įklijavimo komanda:

```
void __fastcall TfrmKeyboard::btnPasteClick(TObject *Sender)
{
frmMain->WordApp->Selection->Paste();
}
```

}

Paste – tai funkcija, kuri perduoda Microsoft Word programai, kad ji įklijuotų laikinoje atmintyje (clipboard) esantį tekstą į Microsoft Word dokumentą, toje vietoje, kurioje yra žymeklis.

TWordApp – tai TWordApplication komponentas, kuris yra TOleServer plėtinys ir leidžia programuotojui palengvinti darbą su Microsoft Word programa. Pagrindiniai komponento parametrai:

- AutoConnect. Reikšmės gali būti **true** arba **false**. Nurodo, ar paleidus programą būtų automatiškai prisijungiama prie Word programos;
- ConnectKind. Nurodoma, kaip daryti sujungimą tarp programos ir Microsoft Word programos. Galimi pasirinkimai:
 - ckRunningOrNew – prijungti prie jau esančios atidarytos Microsoft Word programos arba jei tokios nėra, atidaryti naują;
 - ckNewInstance – visada atidaryti naują Microsoft Word programos langą ir surišti su juo;
 - ckRunningInstance – visada prijungti jau prie paleistos Microsoft Word programos;
 - ckRemote – prijungti prie nuotolinės programos, t.y. esančios paleistos kitame kompiuteryje.
- AutoQuit – nurodoma su **true** arba **false**, kad automatiškai būtų uždaryta Microsoft Word programa, kai bus baigiamas darbas su programa.

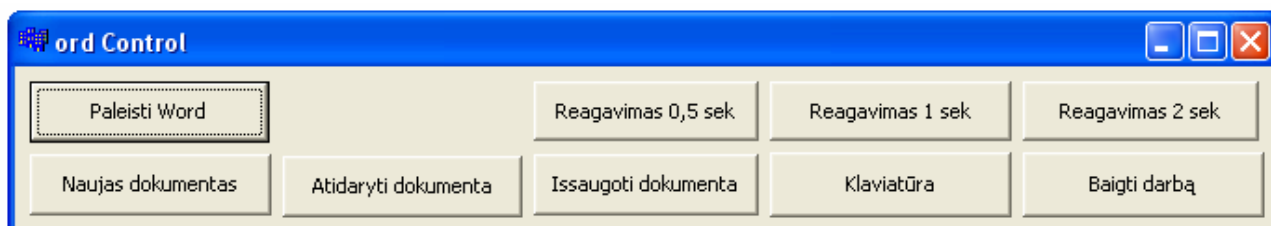
Taip pat redagavimo klaviatūroje yra naudingas klavišas „Space“. Be šio klavišo vargu ar įsivaizduojame teksto redagavimą. „Space“ klavišo funkcija C++ programoje:

```
void __fastcall TfrmKeyboard::btnSpaceClick(TObject *Sender)
{
if ((CAPSState)||(ShiftState)){
frmMain->WordApp->Selection->TypeText((Variant)WideString(" "));
}else
frmMain->WordApp->Selection->TypeText((Variant)WideString(" "));
}
```

Redaguojant tekstą naudojami tokie klavišai, esantys sukurtoje teksto redagavimo klaviatūroje: „End“, „TAB“, „Del“, „Enter“, „Home“, „Page Up“, „Page Down“ ir „Backspace“. Pasirinkus klavišą „Esc“, teksto redagavimo klaviatūra užsidaro ir lieka „Word Control“ langas. Šiame lange pasirinkus klavišą „Baigti darbą“ - programa užsidaro.

3. SUKURTOS TEKSTO REDAGAVIMO PROGRAMOS VALDYMAS

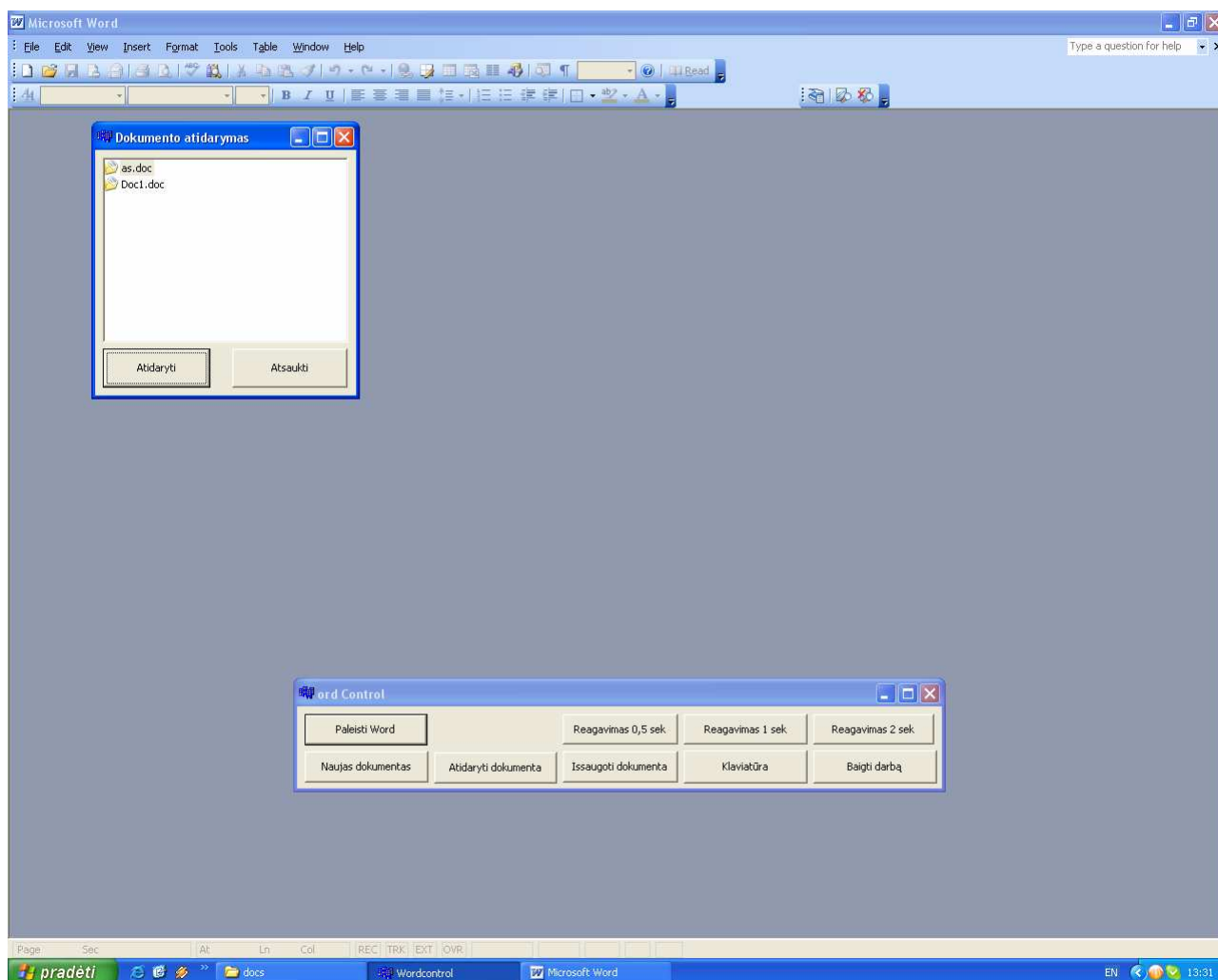
Progarmos „WordControl“ paleidžiamasis exe failas yra C:\docs kataloge. Paspaudus ant WordControl.exe du kartus kairiuoju pelės klavišu, paleidžiama programa. Paleidus programą, nustatomas pradinis 1 sekundės reakcijos laikas, kurį vėliau bus galima keisti. Pagrindinė valdymo klaviatūra pavaizduota 3.1 paveiksle.



3.1 pav. Pagrindinė programos „WordContol“ valdymo klaviatūra

Pasirinkus klavišą „Paleisti Word“, paleidžiama Microsoft Word programa. Galima pasirinkti norimą kursoriaus praledimo laiko intervalą ant klavišo. Galime pasirinkti 0,5 sekundės, 1sekundę ar 2 sekundes - tereikia atitinkamai pasirinkti „Reagavimas 0,5 sek“, „Reagavimas 1 sek“ arba „Reagavimas 2 sek“ klavišą. Tada galimi du pasirinkimo variantai: atidaromas naujas Microsoft Word dokumentas arba atidaromas pasirinktas Microsoft Word dokumentas, esantis C:\docs kataloge. Norint atidaryti naują Microsoft Word dokumentą, reikia pasirinkti klavišą „Naujas dokumentas“, esantį pagrindinėje valdymo klaviatūroje. Pasirinkus klavišą „Atidaryti dokumentą“, esantį pagrindinėje valdymo klaviatūroje, atsidaro Microsoft Word atidarymo klaviatūra. Norint pasirinkti norimą dokumentą, reikia su kursoriaus atsistoti baltame fone atidarymo klaviatūroje. Tada žymeklis juda žemyn, žymėdamas sąraše esamus Microsoft Word dokumentus po vieną. Kai žymeklis pažymi norimą dokumentą, kursorių atitraukiame nuo balto fono ir žymeklis sustoja. Tada pasirenkame klavišą „Atidaryti“ atidarymo klaviatūroje. Jeigu nenorime atidaryti pasirinkto Microsoft Word dokumento, pasirenkame klavišą „Atšaukti“ atidarymo klaviatūroje. Paleistos Microsoft Word programos langas, kuriame parodyta, kaip reikia pasirinkti norimą dokumentą ir, kokioje vietoje yra pagrindinė valdymo klaviatūra, pavaizduota 3.2 paveiksle. Atidarytas pasirinktas Microsoft Word dokumentas su pagrindine valdymo klaviatūra pavaizduotas 3.3 paveiksle. Norėdami redaguoti pasirinktą Microsoft Word dokumentą, reikia pasirinkti klavišą „Klaviatūra“, esantį pagrindinėje valdymo klaviatūroje. Atidaryta teksto redagavimo klaviatūra yra patogioje vietoje ir uždengia pagrindinę valdymo klaviatūrą. Teksto redagavimo klaviatūra pasirinktame Microsoft Word dokumente pavaizduota 3.4 paveiksle. Jei norima pakeisti kursoriaus praledimo

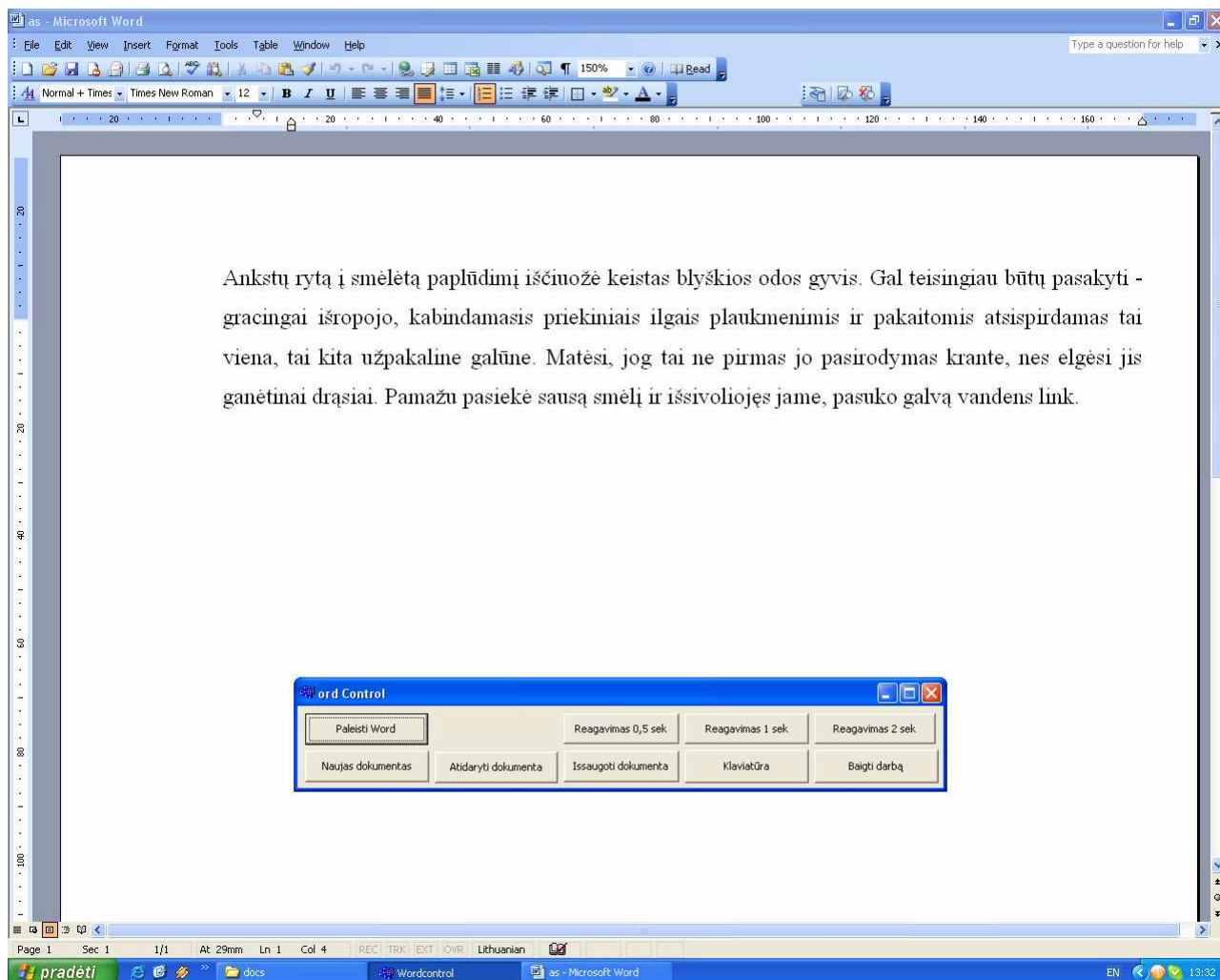
laiko intervalą ant klavišo, reikia pasirinkti teksto redagavimo klaviatūroje klavišą „Esc. Tada teksto redagavimo klaviatūra išsijungia ir lieka tik pagrindinė valdymo klaviatūra. Pasirenkame norimą reagavimo laiką. Tada pasirinkus klavišą „Klaviatūra“ galime toliau redaguoti tekstą.



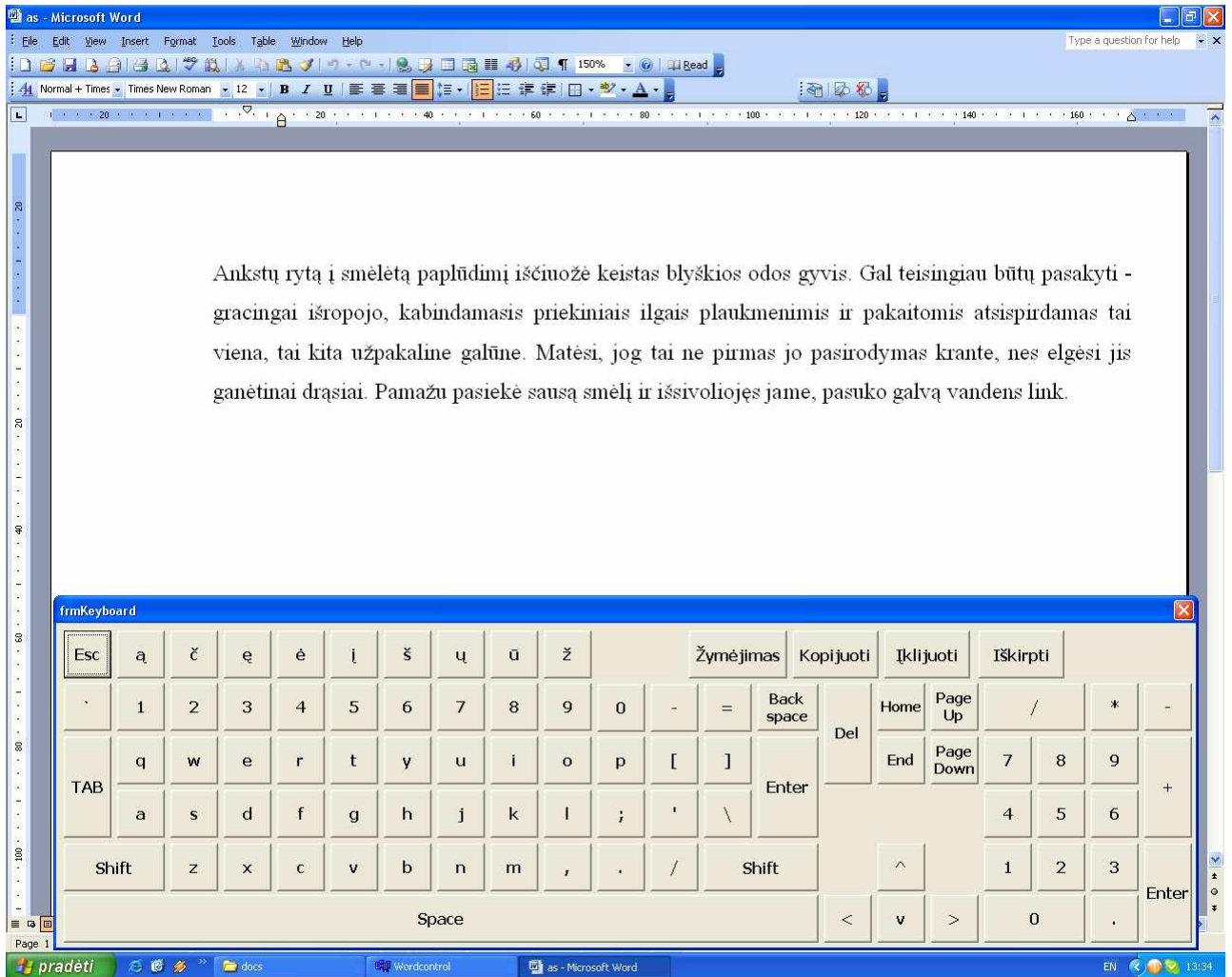
3.2 pav. Atidarytas Microsoft Word programos langas su atidarymo klaviatūra ir pagrindine valdymo klaviatūra

Redaguojant tekstą gali prireikti tam tikrų simbolių arba didžiųjų raidžių, tada reikia pasirinkti „Shift“ klavišą teksto redagavimo klaviatūroje. Teksto redagavimo klaviatūra su įjungtu „Shift“ klavišu pavaizduota 3.5 paveiksle. Visos reikalingos teksto redagavimo operacijos - kaip žymėti, kopijuoti, iškirpti, įklijuoti, trinti ir kt. - yra teksto redagavimo klaviatūroje, kuri pavaizduota 3.4 paveiksle. Norint kopijuoti arba iškirpti tekstą reikia pirmiausia pasirinkti klavišą „Žymėjimas“, tada klavišu „Kursoriaus kryptis“ pažymėti norimą tekstą. Tada pasirenkame klavišą „Kopijuoti“ arba „Iškirpti“, po to, pasirenkamas klavišas „Žymėjimas“, kad išjungti žymėjimo komandą. „Kursoriaus krypties“ klavišų pagalba, atsirandame norimoje teksto vietoje ir, norėdami įklijuoti nukopijuotą arba iškirptą tekstą, pasirenkame klavišą „Įklijuoti“. Norint baigti darbą, reikia

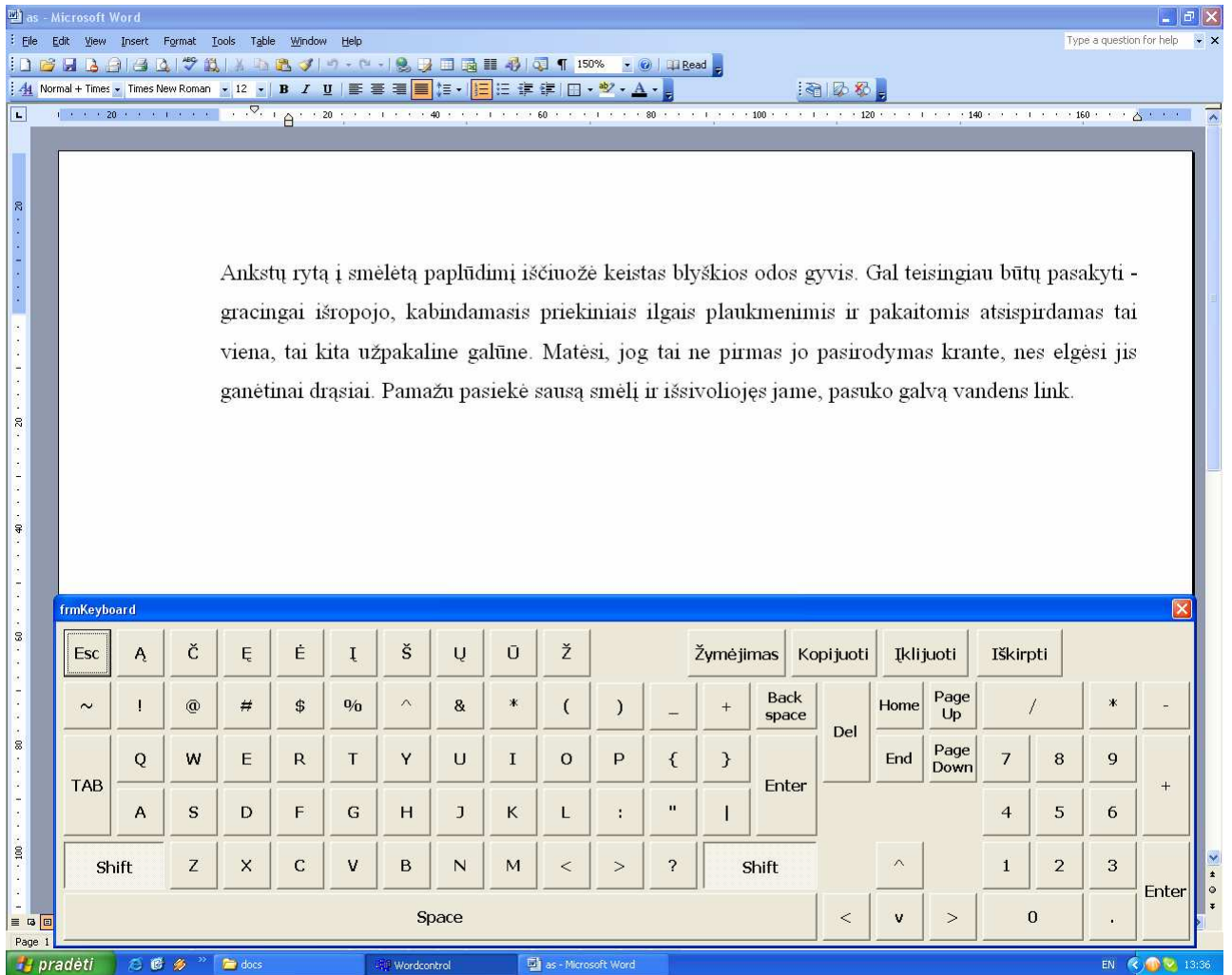
pasirinkti klavišą „Esc“, esantį redagavimo klaviatūroje, ir tada uždaroma klaviatūra. Pagrindinėje valdymo klaviatūroje pasirenkame klavišą „Išsaugoti“. Tada išsaugomas redaguotas tekstas. Pasirenkame klavišą „Baigti darbą“, programa išsijungia ir uždaro Microsoft Word programą.



3.3 pav. Atidarytas pasirinktas Microsoft Word dokumentas su pagrindine valdymo klaviatūra



3.4 pav. Teksto redagavimo klaviatūra pasirinktame Microsoft Word dokumente



3.5 pav. Teksto redagavimo klaviatūra su įjungtu „Shift“ klavišu pasirinktame Microsoft Word dokumente

4. EKSPERIMENTINIS TYRIMAS

4.1. Tiriamųjų pasirinkimas

Planuojant tyrimą, buvo svarbu nustatyti reikalingą minimalų tiriamųjų skaičių, kad būtų galima padaryti statistiškai reikšmingas išvadas, tačiau šiuo atveju tiriamieji bus tik šeši. Tiriamųjų amžius - nuo 20 iki 50 metų. Tiriamieji turėjo redaguoti tekstą standartine klaviatūra, pelės ir žvilgsnio pagalba sukurta teksto redagavimo programa WordControl. Eksperimentinis tyrimas buvo atliekamas su „LC Technologies Eyegaze System“ kompiuteriu ir sukurta programa WordControl. Tiriamojo darbo vieta pavaizduota 4.1.1 paveiksle.



4.1.1 pav. Tiriamojo darbo vieta

Tyrimui naudojamas „LC Technologies Eyegaze System“ kompiuteris, jo sudėtis ir dalys:

Procesorius Intel Pentium CPU 2.80GHz

Monitorius ProLiteE1700S, 17 colių

Atmintis 880 MB DDR 400

Pelė Logitech

Klaviatūra Logitech

Kaupiklis 320GB RPM7200

Įrenginys DVD AD-7173A

4.2. Tyrimo eiga

Tyrimas susideda iš trijų dalių: teksto rinkimas ir redagavimas standartine klaviatūra, teksto rinkimas ir redagavimas pele valdoma WordControl programa, teksto rinkimas ir redagavimas žvilgsniu valdoma WordControl programa.

Tyrimo pradžioje tiriamajam pateikiamas tekstas sudarytas iš 386 simbolių:

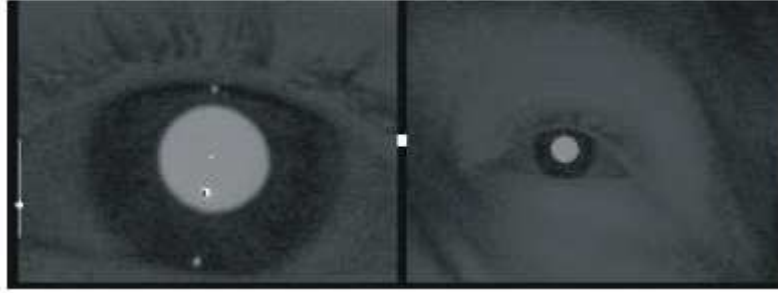
„Ankstų rytą į smėlėtą paplūdimį iščiuožė keistas blyškios odos gyvis. Gal teisingiau būtų pasakyti - gracingai išropojo, kabindamasis priekiniais ilgais plaukmenimis ir pakaitomis atsispirdamas tai viena, tai kita užpakaline galūne. Matėsi, jog tai ne pirmas jo pasirodymas krante, nes elgėsi jis ganėtinai drąsiai. Pamažu pasiekė sausą smėlį ir išsivoliojęs jame, pasuko galvą vandens link.“

Kai tiriamasis susipažįsta su pateikto teksto turiniu, pradėdamas atlikti tyrimas.

1. Teksto rinkimas ir redagavimas standartine klaviatūra. Tiriamasis atsisėdęs prie kompiuterio ir paleidęs programą Microsoft Word, pradeda standartine klaviatūra rinkti tekstą, o padaręs klaidų – jas ištaiso (redaguoja). Pradėjus rinkti tekstą, pradėdamas skaičiuoti laikas ir daromos klaidos. Baigus rinkti ir redaguoti tekstą standartine klaviatūra, gauti rezultatai įrašomi į 4.2.1. lentelę.

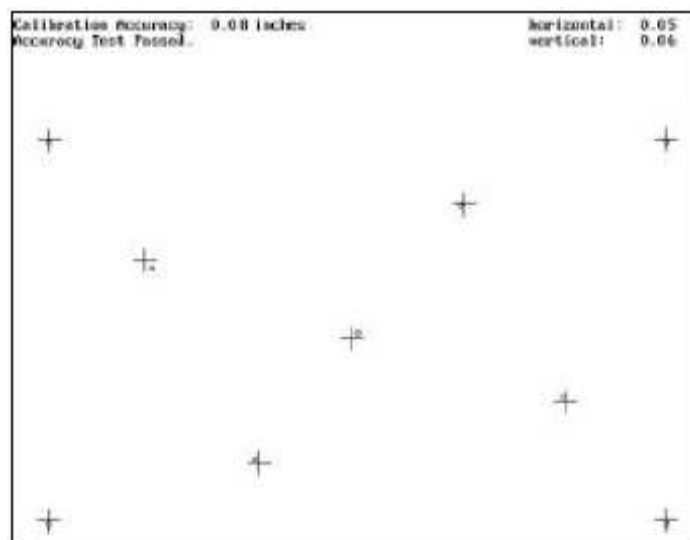
2. Teksto rinkimas ir redagavimas pele valdoma WordControl programa. Tiriamasis paleidžia WordControl programą ir ją valdo pelės pagalba. Pasirinkęs klavišą „Paleisti Word“, tiriamasis paleidžia Microsoft Word programą. Pasirinkus klavišą „Naujas dokumentas“ - atidaromas Microsoft Word naujas dokumentas, o pasirinkus klavišą „Klaviatūra“ - atidaroma teksto redagavimo klaviatūra. Tada tiriamasis pradeda rinkti ir redaguoti tą patį tekstą, kurį rinko standartine klaviatūra. Pradėjus rinkti ir redaguoti tekstą, pradėdamas skaičiuoti laikas ir daromos klaidos. Baigus rinkti ir redaguoti tekstą pele valdoma WordControl programa, gauti rezultatai įrašomi į 4.2.1. lentelę.

3. Teksto rinkimas ir redagavimas žvilgsniu valdoma WordControl programa. Tiriamajam patogiai įsitaisius kėdėje, individualiai nustatomas monitoriaus aukštis, kad būtų optimaliai sukalibruotas akių žvilgsnis. Tada paleidžiama GazeDemo.exe programa ir kameros lęšių pagalba gaunamas ryškus akies atvaizdas monitoriaus kampuose, pavaizduotas 4.2.1 paveiksle. Tiriamasis pasižiūri į kameras ir, jei gerai sureguliuotas akių ryškumas, programa automatiškai pradeda kalibravimą.



4.2.1 pav. Priartintas akies vaizdas ir visas kameros vaizdas

Akių judesiais tiriamasis seka judantį žymeklį, pavaizduotą 4.2.2 paveiksle. Pasibaigus kalibravimui, parodomi žymeklio ir žvilgsniu valdomo kursoriaus buvimo vieta. Jei paklaidos yra nedidelės, pradeda veikti žvilgsniu valdomas kursorius. Tada tiriamasis žvilgsnį nukreipia ant WordControl.exe ikonos ir, paspausdamas klavišą „Enter“, paleidžia WordControl programą. Pasirinkus klavišą „Naujas dokumentas“ - atidaromas Microsoft Word naujas dokumentas, o pasirenkus klavišą „Klaviatūra“ - atidaroma teksto redagavimo klaviatūra. Pradėjus rinkti ir redaguoti tekstą žvilgsniu, pradedamas skaičiuoti laikas ir daromos klaidos. Baigus rinkti ir redaguoti tekstą žvilgsniu valdoma WordControl programa, gauti rezultatai įrašomi į 4.2.1. lentelę.



4.2.2 pav. Kalibravimas

5. REZULTATAI

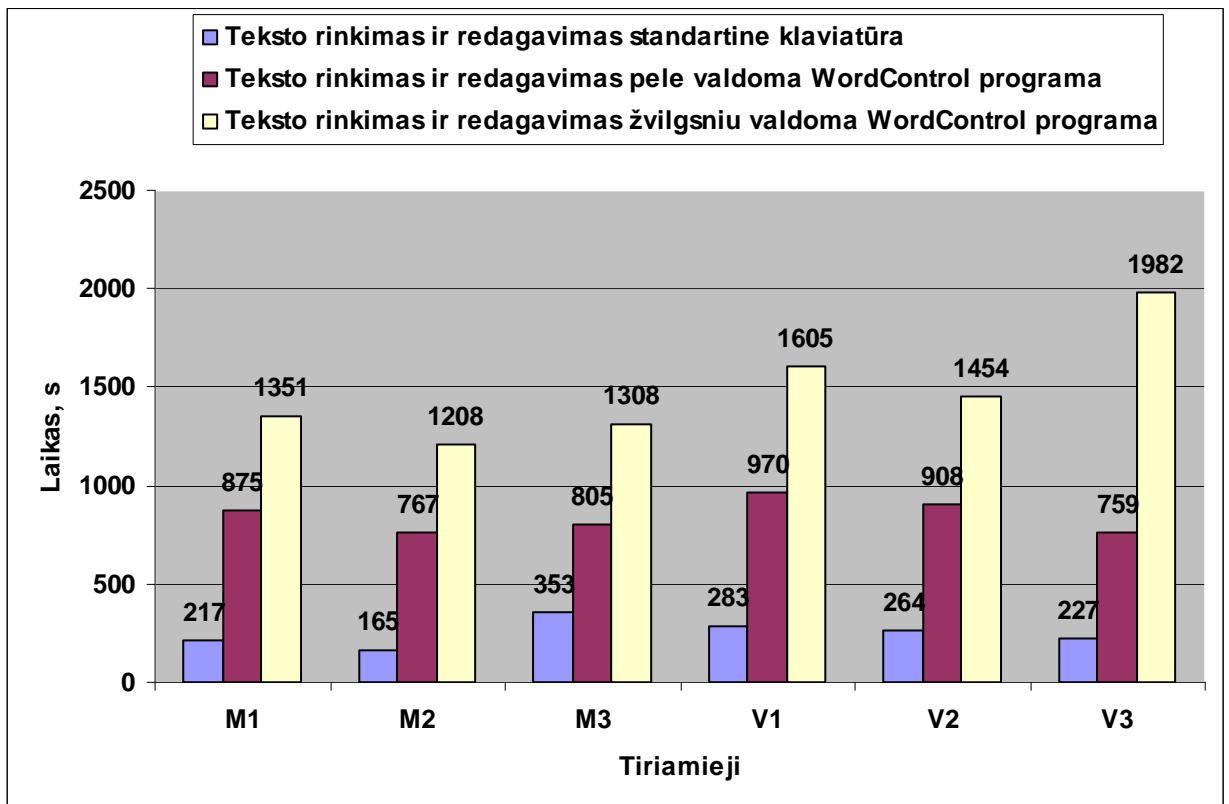
Tyrimo metu buvo užpildyta 5.1. lentelė, kurios rezultatai pateikiami 5.1, 5.2 ir 5.3 paveiksluose.

5.1. lentelė

Tyrimo metu užfiksuotų rezultatų lentelė

| Tiriamasis | Teksto redagavimas rankų pagalba | | Teksto redagavimas pelės pagalba | | Teksto redagavimas žvilgsnio pagalba | |
|------------|----------------------------------|--------------------------|----------------------------------|--------------------------|--------------------------------------|--------------------------|
| | Laikas, s | Padarytų klaidų skaičius | Laikas, s | Padarytų klaidų skaičius | Laikas, s | Padarytų klaidų skaičius |
| M1 | 217 | 1 | 875 | 3 | 1351 | 9 |
| M2 | 165 | 2 | 767 | 5 | 1208 | 8 |
| M3 | 353 | 11 | 805 | 5 | 1308 | 11 |
| V1 | 283 | 5 | 970 | 9 | 1605 | 20 |
| V2 | 264 | 4 | 908 | 6 | 1454 | 12 |
| V3 | 227 | 2 | 759 | 4 | 1982 | 17 |

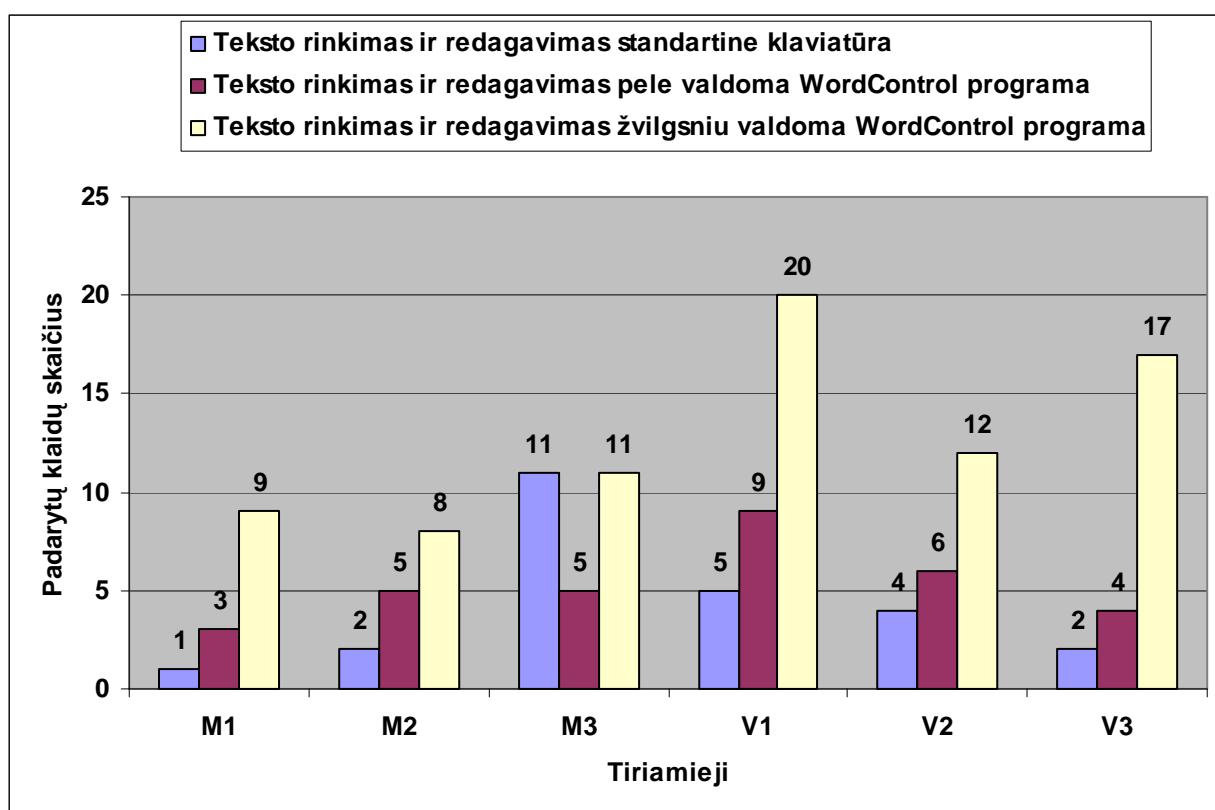
5.1 lentelės paaiškinimas: M – tiriamasis moteris, V – tiriamasis vyras.



5.1 pav. Tiriamiesiems pateikto teksto rinkimo ir redagavimo standartine klaviatūra, pele ir žvilgsniu valdoma WordControl programa, laiko rodikliai

Iš 5.1 paveiksle pateiktų duomenų matyti, kad trumpiausias teksto rinkimo ir redagavimo standartine klaviatūra laikas (165s) yra tiriamojo M2, o ilgiausias laikas (353s) – tiriamojo M3. Trumpiausias teksto rinkimo ir redagavimo pele valdoma WordControl programa laikas (759s) yra tiriamojo V3, o ilgiausias laikas (970s) – tiriamojo V1. Trumpiausias teksto rinkimo ir redagavimo žvilgsniu valdoma WordControl programa laikas (1208s) yra tiriamojo M2, o ilgiausias laikas (1982s) – tiriamojo V3.

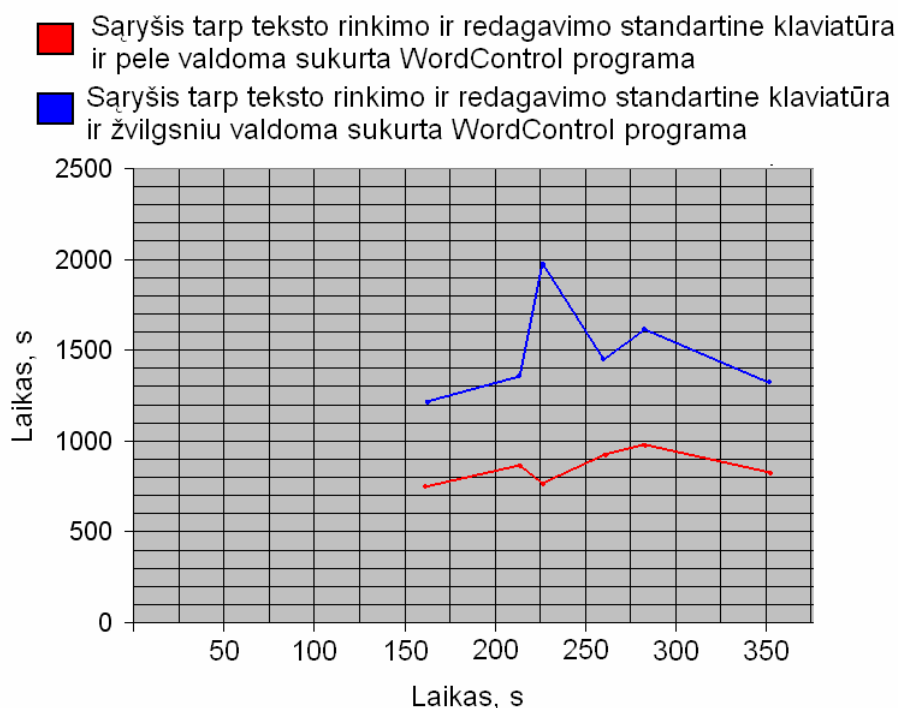
Galima teigti, kad tekstas greičiausiai renkamas ir redaguojamas standartine klaviatūra, o ilgiausiai – žvilgsniu valdoma WordControl programa.



5.2 pav. Tiriamiesiems pateikto teksto rinkimo ir redagavimo standartine klaviatūra, pele ir žvilgsniu valdoma WordControl programa, padarytų klaidų skaičiaus rodikliai

Iš 5.2 paveiksle pateiktų duomenų matyti, kad mažiausiai teksto rinkimo ir redagavimo standartine klaviatūra, klaidų (1) padarė tiriamasis M1, o daugiausiai klaidų (11) – tiriamasis M3. Mažiausiai teksto rinkimo ir redagavimo pele valdoma WordControl programa klaidų (3) padarė tiriamasis M1, o daugiausiai klaidų (9) – tiriamasis V1. Mažiausiai teksto rinkimo ir redagavimo žvilgsniu valdoma WordControl programa, klaidų (8) padarė tiriamasis M2, o daugiausiai klaidų (20) – tiriamasis V1.

Galima teigti, kad mažiausiai klaidų buvo padaryta renkant ir redaguojant tekstą standartine klaviatūra, o daugiausiai – žvilgsniu valdoma WordControl programa.



5.3 pav. Laiko sąryšis tarp teksto rinkimo ir redagavimo metodų

Kai tiriamieji tekstą rinko ir redagavo pele arba žvilgsniu valdoma WordControl programa, programos reagavimo laikas į pasirinkto klavišo paspaudimo imitavimą buvo 1 sekundė, nes tiriamieji pirmą kartą naudojo sukurta programa WordControl ir neturėjo įgūdžių. Tačiau, jei nors vienas tiriamasis būtų ilgiau valdęs WordControl programą, jo reakcijos laikas būtų trumpesnis ir savaime būtų greičiau surenkamas bei redaguojamas tekstas.

Įsivaizduokime, kad tiriamasis, žvilgsniu valdoma WordControl programa naudoja ilgesnį laikotarpį, jo reakcija žymiai pagerėjusi, jis gerai išsimena klaviatūroje išdėstytus klavišus ir visiškai nebedaro klaidų. Tarkim, kad yra labai tiksliai sukalibruotas akių žvilgsnis ir tiriamasis visiškai nejudina galvos, puikiai žino tą patį renkamą tekstą. Tada atliekamas tyrimas, kurio metu renkamame ir redaguojamame tekste yra 386 simboliai, vadinasi, mažiausiai laiko visiems simboliams įvesti reikia:

1. Programos reagavimo laikas į pasirinkto klavišo paspaudimo imitavimą nustatomas 0,5 sekundės reakcijos laikas: $386 \text{ simboliai} * 0,5s \text{ reakcijos laikas} = 193 \text{ sekundės}$.
 2. Tarkime tiriamasis sugaišta 0,5 sekundės, kol nuveda žymeklį nuo klavišo, kuris jau imitavo paspaudimą iki kito klavišo: $386 \text{ klavišai} * 0,5s \text{ reakcijos laikas} = 193 \text{ sekundės}$.
 3. Tiriamasis pasitikrindamas ar teisingai įvedė simbolį dar sugaišta 1 sekundę: $386 \text{ klavišai} * 1s \text{ reakcijos laikas} = 386 \text{ sekundės}$.
 4. Sudedami visi sugaišto laiko rezultatai ir gaunama: $193s + 193s + 386s = 772s$.
- Vadinasi, tiriamasis rinkdamas tekstą be klaidų sugaišo apie 772 sekundes (12min 52s).

IŠVADOS IR SIŪLYMAI

Galima teigti, kad tyrimo rezultatai priklausė nuo tokių sąlygų:

- Kokia tiriamojo patirtis dirbant kompiuteriu, renkant ir koreguojant tekstą;
- Koku tikslumu buvo sukalibruotas žvilgsnis;
- Kaip tiksliai buvo išlaikoma stabili galvos padėtis;
- Kiek įtakojo pašaliniai šviesos spinduliai.

Kadangi tiriamieji tekstą rinko ir koregavo pirmą kartą, pele ir žvilgsniu valdoma WordControl programa, tai programos reagavimo laikas į pasirinkto klavišo paspaudimo imitavimą buvo 1 sekundė. Iš tyrimo rezultatų galima matyti, koks yra trumpiausias ir ilgiausias teksto rinkimo ir redagavimo metodo atlikimo laikas sekundėmis.

Remiantis tyrimo rezultatais galima teigti, kad tekstas greičiausiai renkamas ir redaguojamas standartine klaviatūra, o ilgiausiai – žvilgsniu valdoma WordControl programa. Mažiausiai klaidų buvo padaryta renkant ir redaguojant tekstą standartine klaviatūra, o daugiausiai – žvilgsniu valdoma WordControl programa.

Tyrimas parodė kad, jei tiriamasis tą patį tekstą rinktų ir redaguotų žvilgsniu valdoma WordControl programa ilgesnį laiką ir visiškai nedarytų klaidų, jis galėtų sugaišti apie 772 sekundes.

Remiantis tiriamųjų, dalyvavusių tyrime, nuomone nustatyta, kad pele valdant WordControl programą klaviatūros klavišai yra tinkamo dydžio. WordControl programą valdant žvilgsniu, klavišų dydis galėtų būti truputį didesnis.

Remiantis tiriamųjų, dalyvavusių tyrime, nuomone nustatyta, kad WordControl programą valdant žvilgsniu, klavišų dydis galėtų būti truputį didesnis. Tai galima būtų padaryti panaikinus dešiniąją klaviatūros dalį, kurioje yra skaičiai ir matematiniams veiksams reikalingi simboliai, nes jie kartojasi likusioje klaviatūros dalyje. Panaikinus dešiniąją klaviatūros dalį, galima būtų padidinti likusių klavišų dydį.

LITERATŪRA

1. Hallahan DP, Kauffman JM. Ypatingieji mokiniai, Vilnius, „Alma littera“ 2003.
2. Švenčionių krašto internetinė svetainė. [žiūrėta 2007 05 20] Prieiga per internetą:
<http://svencioniu.krastas.w3.lt/Archyvas/2006-11-18/docs/doc06.htm>
3. [žiūrėta 2007 05 20] Prieiga per internetą:
http://www.mip.lt/temp/w200402232007_Kompiuteriu%20tipai...doc
4. Communication by Gaze Interaction: D3.1 User requirements report with observations of difficulties users are experiencing COGAIN – 511598 (NoE) – Annex I, July 7, 2004 p. 5 – 66.
5. A. Steponavičius, Teksto rinkimo metodai, magistro darbas, Šiauliai, 2006 p. 23 – 44.
6. Eligijus Kubilinskas, Unifikuota kontrolės ir komunikacijų sistema neįgaliesiems, magistro darbas, Šiauliai, 2001 p. 8 – 24.
7. Dasher programos internetinė svetainė. [žiūrėta 2007-05-12] Prieiga per internetą:
<http://www.inference.phy.cam.ac.uk/dasher/>
8. Vydas Čekanavičius, Gediminas Murauskas, Statistika ir jos taikymai I dalis, TEV, Vilnius, 2000 m.
9. Ona Monkevičienė, Specialiojo ugdymo pagrindai, Šiaulių Universitetas, Specialiojo ugdymo mokslinis centras, VŠĮ Šiaulių universiteto leidykla, 2003 m. psl. 426 – 446.
10. Vincas Laurutis, Gintautas Daunys, Akių judesių tyrimai: Taikomasis aspektas, Lietuvos Katalikų mokslo akademijos Suvažiavimo darbai, T.19. Vilnius, 2005 p. 2 – 6.

PRIEDAI

Sukurta WordControl programa ir jos kūrimui skirti failai yra kompaktiniame diske.

Sukurtos programos WordControl aprašymas

Aprašomos naudojamoms pagrindinėms bibliotekoms.

```
#include <vcl.h>
```

```
#pragma hdrstop
```

Sukuriamos visos formos (pagrindinė valdymo klaviatūra, Microsoft Word dokumento atidarymas, teksto redagavimo klaviatūra), ir programa pradeda dirbti;

```
USEFORM("Unit1.cpp", frmMain);
```

```
USEFORM("Unit2.cpp", frmOpenDoc);
```

```
USEFORM("Unit3.cpp", frmKeyboard);
```

Pradeda dirbti programa ir sukurtos formos.

```
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
```

```
{
    try
    {
        Application->Initialize();
        Application->CreateForm(__classid(TfrmMain), &frmMain);
        Application->CreateForm(__classid(TfrmOpenDoc), &frmOpenDoc);
        Application->CreateForm(__classid(TfrmKeyboard), &frmKeyboard);
        Application->Run();
    }
    catch (Exception &exception)
    {
        Application->ShowException(&exception);
    }
    catch (...)
    {
        try
        {
            throw Exception("");
        }
    }
}
```

```
    }  
    catch (Exception &exception)  
    {  
        Application->ShowException(&exception);  
    }  
}  
return 0;  
}
```

Aprašomos naudojami pagrindinės bibliotekos.

```
#include <vcl.h>
#include <wchar.h>
#include <stdlib.h>
#pragma hdrstop
#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
#pragma package(smart_init)
#pragma link "Word_XP_srvr"
#pragma resource "*.dfm"
```

Sukuriami pagrindinės valdymo klaviatūros forma.

```
__fastcall TfrmMain::TfrmMain(TComponent* Owner)
    : TForm(Owner)
{
}
```

Paleidžiama Microsoft Word programa, nustatomas jos lango dydis. Nustatomas pagrindinės ir teksto redagavimo klaviatūrų koordinatės.

```
void __fastcall TfrmMain::Button1Click(TObject *Sender)
{
WordApp->Connect();
WordApp->Resize(800, 400) ;
WordApp->set_Visible(1) ;
frmMain->Left=300;
frmMain->Top=700;
frmKeyboard->Left = 50;
frmKeyboard->Top = 620;
}
```

Atidaro Microsoft Word atidarymo dokumento formą.

```
void __fastcall TfrmMain::btnOpenDocClick(TObject *Sender)
{
frmOpenDoc->ShowModal();
}
```

Atidaro teksto redagavimo klaviatūrą.

```
void __fastcall TfrmMain::Button5Click(TObject *Sender)
```

```
{
frmKeyboard->Show();
}
```

Atidaro naują Microsoft Word dokumentą.

```
void __fastcall TfrmMain::Button3Click(TObject *Sender)
{
  WordApp->Documents->Add(EmptyParam, EmptyParam, EmptyParam, EmptyParam);
}
```

Kai kursorius užvestas ant šio klavišo nustatyta laiko intervalą tada atidaroma Microsoft Word 2007 programa.

```
void __fastcall TfrmMain::btnStartWordMouseEnter(TObject *Sender)
{
  btnName = "btnStartWord";
  CheckTimer->Enabled = true;
}
```

Aprašomas naujo Microsoft Word dokumento atidarymas, kai kursoriaus užvestas ant šio klavišo nustatyta laiko intervalą.

```
void __fastcall TfrmMain::btnNewDocMouseEnter(TObject *Sender)
{
  btnName = "btnNewDoc";
  CheckTimer->Enabled = true;
}
```

Aprašomas Microsoft Word dokumento atidarymas, kai kursoriaus užvestas ant šio klavišo nustatyta laiko intervalą.

```
void __fastcall TfrmMain::btnOpenDocMouseEnter(TObject *Sender)
{
  btnName = "btnOpenDoc";
  CheckTimer->Enabled = true;
}
```

Programa išsaugo Microsoft Word dokumentą.

```
void __fastcall TfrmMain::btnSaveDocMouseEnter(TObject *Sender)
{
  btnName = "btnSaveDoc";
  CheckTimer->Enabled = true;
}
```

Aprašoma funkcija „mygtuko paspaudimas“.

```
void __fastcall TfrmMain::CheckTimerTimer(TObject *Sender)
{
if (btnName != "") {
TButton* MyBtn;
MyBtn = (TButton*)(FindComponent(btnName));
MyBtn->Click();
}
}
```

Programa nieko nedaro jei kursoriaus nėra nustatyta laiko intervalą ant klavišo „Paleisti Word“.

```
void __fastcall TfrmMain::btnStartWordMouseLeave(TObject *Sender)
{
btnName = "";
CheckTimer->Enabled = false;
}
```

Programa nieko nedaro jei kursoriaus nėra nustatyta laiko intervalą ant klavišo „Naujas dokumentas“.

```
void __fastcall TfrmMain::btnNewDocMouseLeave(TObject *Sender)
{
btnName = "";
CheckTimer->Enabled = false;
}
```

Programa nieko nedaro jei kursoriaus nėra nustatyta laiko intervalą ant klavišo „Atidaryti dokumenta“.

```
void __fastcall TfrmMain::btnOpenDocMouseLeave(TObject *Sender)
{
btnName = "";
CheckTimer->Enabled = false;
}
```

Programa nieko nedaro jei kursoriaus nėra nustatyta laiko intervalą ant klavišo „Issaugoti dokumenta“.

```
void __fastcall TfrmMain::btnSaveDocMouseLeave(TObject *Sender)
{
btnName = "";
CheckTimer->Enabled = false;
```



```
}
```

Programa nieko nedaro jei kursoriaus nėra nustatyta laiko intervalą ant klavišo „Klaviatūra“.

```
void __fastcall TfrmMain::btnKeyboardMouseLeave(TObject *Sender)
```

```
{
```

```
btnName = "";
```

```
CheckTimer->Enabled = false;
```

```
}
```

Atidaroma teksto redagavimo klaviatūra, jei kursoriaus užvestas ant šio klavišo nustatyta laiko intervalą.

```
void __fastcall TfrmMain::btnKeyboardMouseEnter(TObject *Sender)
```

```
{
```

```
btnName = "btnKeyboard";
```

```
CheckTimer->Enabled = true;
```

```
}
```

Microsoft word dokumento saugojimas.

```
void __fastcall TfrmMain::btnSaveDocClick(TObject *Sender)
```

```
{
```

```
OleVariant par1, par2;
```

```
par1 = true;
```

```
par2 = wdWordDocument;
```

```
Doc->SaveAs((OleVariant)filename);
```

```
}
```

Uždaroma pagrindinė valdymo klaviatūra.

```
void __fastcall TfrmMain::btnCloseClick(TObject *Sender)
```

```
{
```

```
frmMain->Close();
```

```
}
```

Programa nieko nedaro jei kursoriaus nėra nustatyta laiko intervalą ant klavišo „Baigti darbą“

```
void __fastcall TfrmMain::btnCloseMouseLeave(TObject *Sender)
```

```
{
```

```
btnName = "";
```

```
CheckTimer->Enabled = false;
```

```
}
```

Programa uždaroma, jei kursorius užvestas ant šio klavišo nustatyta laiko intervalą.

```
void __fastcall TfrmMain::btnCloseMouseEnter(TObject *Sender)
```

```
{
btnName = "btnClose";
CheckTimer->Enabled = true;
}
```

Nustatomas 0,5 ssekundės laiko intervalas, kai kursorius yra ant klavišo.

```
void __fastcall TfrmMain::btnTime05MouseEnter(TObject *Sender)
{
btnName = "btnTime05";
CheckTimer->Enabled = true;
}
```

Nustatomas 1 sekundės laiko intervalas, kai kursorius yra ant klavišo.

```
void __fastcall TfrmMain::btnTime10MouseEnter(TObject *Sender)
{
btnName = "btnTime10";
CheckTimer->Enabled = true;
}
```

Nustatomas 2 sekundžių laiko intervalas, kai kursorius yra ant klavišo.

```
void __fastcall TfrmMain::btnTime20MouseEnter(TObject *Sender)
{
btnName = "btnTime20";
CheckTimer->Enabled = true;
}
```

Programa nieko nedaro jei kursoriaus nėra nustatyta laiko intervalą ant klavišo „Reagavimas 0,5 sek“.

```
void __fastcall TfrmMain::btnTime05MouseLeave(TObject *Sender)
{
btnName = "";
CheckTimer->Enabled = false;
}
```

Programa nieko nedaro jei kursoriaus nėra nustatyta laiko intervalą ant klavišo „Reagavimas 1 sek“.

```
void __fastcall TfrmMain::btnTime10MouseLeave(TObject *Sender)
{
btnName = "";
CheckTimer->Enabled = false;
}
```

Programa nieko nedaro jei kursoriaus nėra nustatyta laiko intervalą ant klavišo „Reagavimas 2 sek“.

```
void __fastcall TfrmMain::btnTime20MouseLeave(TObject *Sender)
```

```
{
  btnName = "";
  CheckTimer->Enabled = false;
}
```

Programa įjungia 0,5 sekundės taimerį.

```
void __fastcall TfrmMain::btnTime05Click(TObject *Sender)
```

```
{
  frmMain->CheckTimer->Interval = 500;
  frmKeyboard->CheckTimer->Interval = 500;
  frmOpenDoc->CheckTimer->Interval = 500;
}
```

Programa įjungia 1 sekundės taimerį.

```
void __fastcall TfrmMain::btnTime10Click(TObject *Sender)
```

```
{
  frmMain->CheckTimer->Interval = 1000;
  frmKeyboard->CheckTimer->Interval = 1000;
  frmOpenDoc->CheckTimer->Interval = 1000;
}
```

Programa įjungia 2 sekundžių taimerį.

```
void __fastcall TfrmMain::btnTime20Click(TObject *Sender)
```

```
{
  frmMain->CheckTimer->Interval = 2000;
  frmKeyboard->CheckTimer->Interval = 2000;
  frmOpenDoc->CheckTimer->Interval = 2000;
}
```

Aprašomos naudojamoms pagrindinėms bibliotekoms.

```
#include <vcl.h>
#pragma hdrstop
#include "Unit2.h"
#include "Unit1.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
```

Aprašomi naudojami kintamieji.

```
TfrmOpenDoc *frmOpenDoc;
```

```
int listpos=0;
```

```
AnsiString btnName;
```

Sukuriamas Microsoft Word dokumento atidarymo forma.

```
__fastcall TfrmOpenDoc::TfrmOpenDoc(TComponent* Owner)
    : TForm(Owner)
{
}
```

Atidaromas pasirinktas Microsoft Word dokumentas esantis c:\docs\ direktorijoje.

```
void __fastcall TfrmOpenDoc::FormCreate(TObject *Sender)
{
}
```

Atidaro dokumento forma ir užpildo rastų dokumentų sąrašą.

```
void __fastcall TfrmOpenDoc::FormShow(TObject *Sender)
{
HANDLE hSearch;
WIN32_FIND_DATA FileData;
DocList->Items->Clear();
hSearch = FindFirstFile("c:\\docs\\*.doc", &FileData);
TListItem *li = DocList->Items->Add();
    li->ImageIndex = 1;
    li->Caption = FileData.cFileName;
if (hSearch) {
    li = DocList->Items->Add();
    li->ImageIndex = 1;
    li->Caption = FileData.cFileName;
}
```

```

while (FindNextFile(hSearch, &FileData))
{
    li = DocList->Items->Add();
    li->ImageIndex = 1;
    li->Caption = FileData.cFileName;
}
}

```

Atidaromas iš sąrašo pasirinktas Microsoft Word dokumentas.

```

void __fastcall TfrmOpenDoc::Button1Click(TObject *Sender)
{
    OleVariant docname;
    if (DocList->ItemIndex>=0) {
        docname= "c:\\docs\\"+ DocList->Items->Item[DocList->ItemIndex]->Caption;
        frmMain->Doc->ConnectTo(frmMain->WordApp->Documents->Open(docname,
EmptyParam, EmptyParam, EmptyParam,
EmptyParam, EmptyParam, EmptyParam, EmptyParam,
EmptyParam, EmptyParam));
        frmMain->filename = docname;
    }
    frmOpenDoc->Close();
}

```

Kai kursorius užaina ant formos kraštų nebeįveda pasirinktų dokumentų žymeklis.

```

void __fastcall TfrmOpenDoc::DocListMouseLeave(TObject *Sender)
{
    ListTimer->Enabled = false;
}

```

Programa nieko nedaro jei kursoriaus nėra nustatyta laiko intervalą ant klavišo „Paleisti Word“.

```

void __fastcall TfrmOpenDoc::btnOpenDocMouseLeave(TObject *Sender)
{
    btnName = "";
    CheckTimer->Enabled = false;
}

```

Programa nieko nedaro jei kursoriaus nėra nustatyta laiko intervalą ant klavišo „Cancel“.

```

void __fastcall TfrmOpenDoc::btnCancelMouseLeave(TObject *Sender)
{

```

```
btnName = "";
CheckTimer->Enabled = false;
}
```

Kai kursorius užvestas ant formos (kur yra dokumentų sąrašas) nustatyta laiko intervalą, tada juda žymeklis dokumentų sąraše.

```
void __fastcall TfrmOpenDoc::DocListMouseEnter(TObject *Sender)
{
ListTimer->Enabled = true;
}
```

Kai kursorius užvestas ant šio klavišo nustatyta laiko intervalą tada atidaromas pasirinktas Microsoft Word dokumentas.

```
void __fastcall TfrmOpenDoc::btnOpenDocMouseEnter(TObject *Sender)
{
btnName = "btnOpenDoc";
CheckTimer->Enabled = true;
}
```

Kai kursorius užvestas ant šio klavišo nustatyta laiko intervalą tada uždaroma sąrašo forma.

```
void __fastcall TfrmOpenDoc::btnCancelMouseEnter(TObject *Sender)
{
btnName = "btnCancel";
CheckTimer->Enabled = true;
}
```

Paleidžiama taimerio funkcija.

```
void __fastcall TfrmOpenDoc::CheckTimerTimer(TObject *Sender)
{
if (btnName != "") {
TButton* MyBtn;
MyBtn = (TButton*)(FindComponent(btnName));
MyBtn->Click();
}
}
```

Pagal nustatyta laiko intervalą dokumentų sąraše juda žymeklis.

```
void __fastcall TfrmOpenDoc::ListTimerTimer(TObject *Sender)
{
DocList->ItemIndex = listpos;
```

```
listpos ++;
```

```
if (listpos == DocList->Items->Count) {
```

```
    listpos = 0;
```

```
}
```

```
}
```

Uždaroma dokumentų sąrašo forma.

```
void __fastcall TfrmOpenDoc::btnCancelClick(TObject *Sender)
```

```
{
```

```
    frmOpenDoc->Close();
```

```
}
```

Aprašomos naudojamoms pagrindinėms bibliotekoms.

```
#include <vcl.h>
```

```
#pragma hdrstop
```

```
#include "Unit3.h"
```

```
#include "Unit1.h"
```

```
#pragma package(smart_init)
```

```
#pragma resource "*.dfm"
```

Aprašomi naudojami kintamieji.

```
TfrmKeyboard *frmKeyboard;
```

```
bool ShiftState;
```

```
bool CAPSState;
```

```
bool SelectState;
```

```
int SelPos;
```

```
int MouseMove;
```

```
AnsiString btnName;
```

```
OleVariant moveobject, movepos, range, rangestart, rangeend;
```

Atidaroma redagavimo klaviatūros forma.

```
__fastcall TfrmKeyboard::TfrmKeyboard(TComponent* Owner)
```

```
    : TForm(Owner)
```

```
{
```

```
}
```

Užsidaro teksto redagavimo klaviatūra.

```
void __fastcall TfrmKeyboard::btnESCClick(TObject *Sender)
```

```
{
```

```
    Close();
```

```
}
```

Aprašomas „Shift“ klavišo paspaudimas.

```
void __fastcall TfrmKeyboard::btnShiftClick(TObject *Sender)
```

```
{
```

```
    if (btnShift->Down) {
```

```
        btnShift->Down = true;
```

```
        btnShift2->Down = true;
```

```
        ShiftState = true;
```

```
        DoShift();
```



```

}else
{
    btnShift->Down = false;
    btnShift2->Down = false;
    ShiftState = false;
    DoUnShift();
}
}

```

Programa reaguoja paspaustas arba nepaspaustas „Shift“ klavišas.

```

void __fastcall TfrmKeyboard::btnShift2Click(TObject *Sender)

```

```

{
if (btnShift2->Down) {
    btnShift->Down = true;
    btnShift2->Down = true;
    ShiftState = true;
    DoShift();
}
else
{
    btnShift->Down = false;
    btnShift2->Down = false;
    ShiftState = false;
    DoUnShift();
}
}

```

```

void TfrmKeyboard::DoShift()

```

```

{
    btnA1->Caption = "A";
    btnC1->Caption = "Č";
    btnE1->Caption = "Ė";
    btnE12->Caption = "Ē";
    btnI1->Caption = "Į";
    btnS1->Caption = "Š";
    btnU1->Caption = "U";
    btnU12->Caption = "Ū";
    btnZ1->Caption = "Ž";
}

```

btnTilde->Caption = "~";
btn1->Caption = "!";
btn2->Caption = "@";
btn3->Caption = "#";
btn4->Caption = "\$";
btn5->Caption = "%";
btn6->Caption = "^";
btn7->Caption = "&&";
btn8->Caption = "*";
btn9->Caption = "(";
btn0->Caption = ")";
btnDash->Caption = "_";
btnEqual->Caption = "+";
btnQ->Caption = "Q";
btnW->Caption = "W";
btnE->Caption = "E";
btnR->Caption = "R";
btnT->Caption = "T";
btnY->Caption = "Y";
btnU->Caption = "U";
btnI->Caption = "I";
btnO->Caption = "O";
btnP->Caption = "P";
btnA->Caption = "A";
btnS->Caption = "S";
btnD->Caption = "D";
btnF->Caption = "F";
btnG->Caption = "G";
btnH->Caption = "H";
btnJ->Caption = "J";
btnK->Caption = "K";
btnL->Caption = "L";
btnZ->Caption = "Z";
btnX->Caption = "X";
btnC->Caption = "C";

```
    btnV->Caption = "V";
    btnB->Caption = "B";
    btnN->Caption = "N";
    btnM->Caption = "M";
    btnParag1->Caption = "{";
    btnParag2->Caption = "}";
    btnP2->Caption = ":";
    btnE2->Caption = "\"";
    btnBSlash->Caption = "|";
    btnLess1->Caption = "<";
    btnLess2->Caption = ">";
    btnSlash->Caption = "?";
}
void TfrmKeyboard::DoUnShift()
{
    btnA1->Caption = "a";
    btnC1->Caption = "č";
    btnE1->Caption = "e";
    btnE12->Caption = "ë";
    btnI1->Caption = "i";
    btnS1->Caption = "š";
    btnU1->Caption = "u";
    btnU12->Caption = "ü";
    btnZ1->Caption = "ž";
    btnTilde->Caption = "`";
    btn1->Caption = "1";
    btn2->Caption = "2";
    btn3->Caption = "3";
    btn4->Caption = "4";
    btn5->Caption = "5";
    btn6->Caption = "6";
    btn7->Caption = "7";
    btn8->Caption = "8";
    btn9->Caption = "9";
    btn0->Caption = "0";
```

btnDash->Caption = "-";
btnEqual->Caption = "=";
btnQ->Caption = "q";
btnW->Caption = "w";
btnE->Caption = "e";
btnR->Caption = "r";
btnT->Caption = "t";
btnY->Caption = "y";
btnU->Caption = "u";
btnI->Caption = "i";
btnO->Caption = "o";
btnP->Caption = "p";
btnA->Caption = "a";
btnS->Caption = "s";
btnD->Caption = "d";
btnF->Caption = "f";
btnG->Caption = "g";
btnH->Caption = "h";
btnJ->Caption = "j";
btnK->Caption = "k";
btnL->Caption = "l";
btnZ->Caption = "z";
btnX->Caption = "x";
btnC->Caption = "c";
btnV->Caption = "v";
btnB->Caption = "b";
btnN->Caption = "n";
btnM->Caption = "m";
btnParag1->Caption = "[";
btnParag2->Caption = "];
btnP2->Caption = ";";
btnE2->Caption = """;
btnBSlash->Caption = "\\";
btnLess1->Caption = "<";
btnLess2->Caption = "<.";

```

    btnSlash->Caption = "/";
}

```

Čia paspaudus klavišą „A“ arba „a“ atspasdinama „A“ arba „a“ raidė Microsoft Word dokumente. Tokiu pat metodu atspasdinami likę klaviatūros simboliai ir raidės

```

void __fastcall TfrmKeyboard::btnA1Click(TObject *Sender)
{
    if ((CAPSSState)||((ShiftState)){
        frmMain->WordApp->Selection->TypeText((Variant)WideString("A"));
    } else
        frmMain->WordApp->Selection->TypeText((Variant)WideString("a"));
}

```

Aprašomas „Space“ klavišo paspaudimas.

```

void __fastcall TfrmKeyboard::btnSpaceClick(TObject *Sender)
{
    if ((CAPSSState)||((ShiftState)){
        frmMain->WordApp->Selection->TypeText((Variant)WideString(" "));
    } else
        frmMain->WordApp->Selection->TypeText((Variant)WideString(" "));
}

```

Kursoriaus paslinkimas kairėn per 1 žingsneli Microsoft Word redaguojamame dokumente.

```

void __fastcall TfrmKeyboard::btnLeftClick(TObject *Sender)
{
    moveobject = wdCharacter;
    movepos = -1;
    frmMain->WordApp->Selection->Move(moveobject,movepos);
}

```

Kursoriaus paslinkimas dešinėn per 1 žingsneli Microsoft Word redaguojamame dokumente.

```

void __fastcall TfrmKeyboard::btnRightClick(TObject *Sender)
{
    moveobject = wdCharacter;
    movepos = 1;
    frmMain->WordApp->Selection->Move(moveobject,movepos);
    if (SelectState) {
        frmMain->WordApp->Selection->Move(moveobject,movepos);
        rangeend = frmMain->WordApp->Selection->End;
    }
}

```

```

        DoSelect();
    }
}

```

Kursoriaus paslinkimas aukštyn per 1 žingsneli Microsoft Word redaguojamame dokumente.

```

void __fastcall TfrmKeyboard::btnUpClick(TObject *Sender)
{
    moveobject = wdLine;
    movepos = -1;
    frmMain->WordApp->Selection->MoveDown(moveobject,movepos);
if (SelectState) {
        rangeend = frmMain->WordApp->Selection->End;
        DoSelect();
    }
}

```

Kursoriaus paslinkimas žemyn per 1 žingsneli Microsoft Word redaguojamame dokumente.

```

void __fastcall TfrmKeyboard::brnDownClick(TObject *Sender)
{
    moveobject = wdLine;
    movepos = 1;
    frmMain->WordApp->Selection->MoveDown(moveobject,movepos);
if (SelectState) {
        rangeend = frmMain->WordApp->Selection->End;
        DoSelect();
    }
}

```

Kursorius perkeliamas redaguojamame Microsoft Word dokumente lapu į viršų.

```

void __fastcall TfrmKeyboard::btnPageUpClick(TObject *Sender)
{
    OleVariant par1, par2;
    par1 = 0;
    par2 = 1;
    frmMain->WordApp->ActiveWindow->PageScroll(par1, par2);
}

```

Kursorius perkeliamas į Microsoft Word redaguojamo dokumento eilutės pradžią.

```

void __fastcall TfrmKeyboard::btnHomeClick(TObject *Sender)

```

```

{
OleVariant par1, par2;
par1 = wdLine;
par2 = wdMove;
frmMain->WordApp->Selection->HomeKey(par1, par2);
if (SelectState) {
rangeend = frmMain->WordApp->Selection->End;
    DoSelect();
}
}

```

Kursorius perkeliamas į Microsoft Word redaguojamo dokumento eilutės pabaigą.

```

void __fastcall TfrmKeyboard::btnEndClick(TObject *Sender)

```

```

{
OleVariant par1, par2;
par1 = wdLine;
par2 = wdMove;
frmMain->WordApp->Selection->EndKey(par1, par2);
if (SelectState) {
rangeend = frmMain->WordApp->Selection->End;
    DoSelect();
}
}

```

Microsoft Word redaguojamame dokumente suaktyvinamas „Del“ klavišas, kuris vykdo trynimo komandą.

```

void __fastcall TfrmKeyboard::btnDelClick(TObject *Sender)

```

```

{
OleVariant par1, par2;
par1 = wdCharacter;
par2 = 1;
frmMain->WordApp->Selection->Delete(par1, par2);
if (SelectState) {
    DoSelect();
}
}

```

Microsoft Word redaguojamame dokumente suaktyvinamas „Backspace“ klavišas, kuris vykdo trynimo komandą.

```
void __fastcall TfrmKeyboard::btnBackSpaceClick(TObject *Sender)
{
    OleVariant par1, par2;
    par1 = wdCharacter;
    par2 = -1;
    frmMain->WordApp->Selection->Delete(par1, par2);
    if (SelectState) {
        DoSelect();
    }
}
```

Microsoft Word redaguojamame dokumente suaktyvinamas „Enter“ klavišas, kuris vykdo perėjimo į žemesnę eilutę komandą.

```
void __fastcall TfrmKeyboard::btnEnterClick(TObject *Sender)
{
    frmMain->WordApp->Selection->TypeParagraph();
}
```

Microsoft Word redaguojamame dokumente suaktyvinamas „Enter“ klavišas, kuris vykdo perėjimo į žemesnę eilutę komandą.

```
void __fastcall TfrmKeyboard::btnEnter2Click(TObject *Sender)
{
    frmMain->WordApp->Selection->TypeParagraph();
}
```

Kursorius perkeliamas redaguojamame Microsoft Word dokumente lapu į apačią..

```
void __fastcall TfrmKeyboard::btnPageDownClick(TObject *Sender)
{
    OleVariant par1, par2;
    par1 = 1;
    par2 = 0;
    frmMain->WordApp->ActiveWindow->PageScroll(par1, par2);
}
```

Microsoft Word redaguojamame dokumente suaktyvinamas „Iškirpti“ klavišas, kuris vykdo iškirpimo komandą.

```
void __fastcall TfrmKeyboard::btnCutClick(TObject *Sender)
```



```
{
frmMain->Doc->Range(rangestart, rangeend)->Cut();
}
```

Microsoft Word redaguojamame dokumente suaktyvinamas „Įklijuoti“ klavišas, kuris vykdo įklijavimo komandą.

```
void __fastcall TfrmKeyboard::btnPasteClick(TObject *Sender)
{
frmMain->WordApp->Selection->Paste();
}
```

Microsoft Word redaguojamame dokumente suaktyvinamas „Kopijuoti“ klavišas, kuris vykdo kopijavimo komandą.

```
void __fastcall TfrmKeyboard::btnCopyClick(TObject *Sender)
{
frmMain->Doc->Range(rangestart, rangeend)->Copy();
}
```

Microsoft Word redaguojamame dokumente suaktyvinamas „Žymėjimas“ klavišas, kuris vykdo žymėjimo komandą.

```
void __fastcall TfrmKeyboard::btnSelectClick(TObject *Sender)
{
if (btnSelect->Down) {
    btnSelect->Down = true;
    SelectState = true;
    rangestart = frmMain->WordApp->Selection->End;
    SelPos = 1;
else
    {
        btnSelect->Down = false;
        SelectState = false;
        rangeend = frmMain->WordApp->Selection->End;
        range = frmMain->Doc->Range(rangestart, rangeend);
    }
}
```

```
void TfrmKeyboard::DoSelect()
{
    rangeend = frmMain->WordApp->Selection->End;
```

```

range = frmMain->Doc->Range(rangestart, rangeend);
frmMain->Doc->Range(rangestart, rangeend)->Select();
}

```

Paspaudžiamas klavišas „A“, jei kursoriaus užvestas ant šio klavišo nustatytą laiko intervalą. Taip aprašomi likę symboliai ir raidės.

```

void __fastcall TfrmKeyboard::btnAMouseEnter(TObject *Sender)
{
    btnName = "btnA";
    CheckTimer->Enabled = true;
}

```

Programa nieko nedaro jei kursoriaus nėra nustatytą laiko intervalą ant šio klavišo. Taip aprašomi likę symboliai ir raidės.

```

void __fastcall TfrmKeyboard::btnAMouseLeave(TObject *Sender)
{
    btnName = "";
    CheckTimer->Enabled = false;
}

```

Imituojamas klavišo „Shift“ įjungimas – išjungimas, kai kursoriaus užvestas ant šio klavišo pagal nustatytą laiko intervalą.

```

void __fastcall TfrmKeyboard::CheckTimerTimer(TObject *Sender)
{
if (btnName != "") {
    TButton* MyBtn;
    MyBtn = (TButton*)(FindComponent(btnName));
    MyBtn->Click();
}
if (btnName == "btnShift") {
    if (!btnShift->Down) {
        btnShift->Down = true;
        btnShift2->Down = true;
        ShiftState = true;
        DoShift();
    }
else
    {
        btnShift->Down = false;

```

```

        btnShift2->Down = false;
        ShiftState = false;
        DoUnShift();
    }
}
if (btnName == "btnShift2") {
    if (!btnShift->Down) {
        btnShift->Down = true;
        btnShift2->Down = true;
        ShiftState = true;
        DoShift();
    } else
    {
        btnShift->Down = false;
        btnShift2->Down = false;
        ShiftState = false;
        DoUnShift();
    }
}
if (btnName == "btnSelect") {
    if (!btnSelect->Down) {
        btnSelect->Down = true;
        SelectState = true;
        rangestart = frmMain->WordApp->Selection->End;
        SelPos = 1;
    } else
    {
        btnSelect->Down = false;
        SelectState = false;
        rangeend = frmMain->WordApp->Selection->End;
        range = frmMain->Doc->Range(rangestart, rangeend);
    }
}
}
}

```

Programa nieko nedaro jei kursoriaus nėra nustatyta laiko intervalą ant šio klavišo. Tokiu metodu aprašomi likę simboliai ir raidės.

```
void __fastcall TfrmKeyboard::btnESCMouseLeave(TObject *Sender)
{
    btnName = "";
    CheckTimer->Enabled = false;
}
```

Kai kursorius užvestas ant šio klavišo nustatyta laiko intervalą tada imituojamas šio klavišo paspaudimas. Tokiu metodu aprašomi likę simboliai ir raidės.

```
void __fastcall TfrmKeyboard::btnESCMouseEnter(TObject *Sender)
{
    btnName = "btnESC";
    CheckTimer->Enabled = true;
}
```