

Uniwersytet Mikołaja Kopernika w Toruniu

Wydział Filozofii i Nauk Społecznych

Katedra Kognitywistyki

Paweł Drojecki

nr. albumu 295460

kognitywistyka

Praca licencjacka

Porównanie metod aktywacji przycisków za pomocą wzroku

Opiekun pracy dyplomowej

dr hab. Jacek Matulewski

Katedra Informatyki Stosowanej

Wydział Fizyki, Astronomii

i Informatyki Stosowanej

Toruń, 2020

Spis treści

Wstęp (str 2)

Rozdział 1 - Interakcja wzrokowa (str 3 - 22)

Rozdział 2 - Problem badawczy i hipotezy (str 23 - 26)

Rozdział 3 - Procedura eksperymentu (str 27 - 35)

3.1. Osoby badane i rekrutacja (str 27 - 31)

3.2 Opis działania metod aktywacji przycisków i układów klawiszy (str 28 - 35)

Rozdział 4 - Analiza i wyniki (str 36 - 45)

Rozdział 5 - Dyskusja (str 46 - 47)

Rozdział 6 - Pomysły na rozwój oprogramowania (str 48 - 50)

Bibliografia (str 51 - 54)

Załączniki (str 55 - 57)

Wstęp

Wzrok to narząd umożliwiający nam widzenie, a więc odbiór sygnałów. Okulografy pozwalają go jednak zmienić w narzędzie pozwalające na kontrolę, w pierwszym rzędzie kontrolę komputera, a za jego pośrednictwem także naszego otoczenia. To szczególnie ważna możliwość, jeżeli inne sposoby wywierania wpływu na świat są niedostępne. Okulograf, lub z angielskiego eyetracker, umożliwia wyznaczenie miejsca spojrzenia. Miejsce to może być świadomie kontrolowane przez użytkownika, co umożliwia interakcję wzrokową między człowiekiem a komputerem. Wzrok pełni tu rolę podobną do myszy komputerowej - wskazuje miejsca na ekranie i pozwala na ich "kliknięcie".

W pracy opisany jest eksperyment i jego wyniki, w którym wykorzystany został program, pozwalający na wprowadzanie czterocyfrowych liczb (dalej będziemy je nazywali pinami) za pomocą trzech metod wykorzystujących eyetracker. Wśród nich była nowatorska metoda wykorzystująca gesty wzrokowe.

Celem badania było zweryfikowanie tego, jak osoby badane radzą sobie z każdą z tych metod oraz jak można te metody zoptymalizować. W badaniach wzięły udział osoby zdrowe, ale należy traktować je jako pierwszy etap służący do poprawienia narzędzi eksperymentalnych przed przystąpieniem do badań z udziałem osób niepełnosprawnych ruchowo.

Rozdział 1

Interakcja wzrokowa

Interakcja wzrokowa (ang. *gaze interaction*) to część szerszego działu zajmującego się interakcją człowiek-komputer (ang. *human-computer interaction*), w której okulograf (ang. *eyetracker*) używany jest jako kontroler oprogramowania. To oznacza, że powstaje pętla zwrotna pozwalająca na kontrolę zawartości ekranu za pomocą kierunku spojrzenia i jego zmiany tzn. wykonywanych przez użytkownika ruchów oczu. Prace naukowe dotyczące tego działu powstają od lat osiemdziesiątych; a obecnie coraz więcej ośrodków naukowych zajmuje się tym tematem. Nie będę tu przedstawiał podstawowych zagadnień dotyczących interakcji wzrokowej, przejdę od razu do zagadnień szczegółowych, związanych ze wzrokowym wprowadzaniem tekstu (ang. *gaze text entry*) i przeprowadzonym przeze mnie eksperymentem. Po szersze wprowadzenie do dziedziny odsyłam do podręczników A. Duchowski(2003) oraz K. Holmqvist (2017).

Ważnym zagadnieniem dotyczącym wzrokowego wprowadzania tekstu jest optymalizacja polegająca na zwiększaniu szybkości wprowadzania znaków, przy jednoczesnym zachowaniu niewielkiej liczby pomyłek. Miarą prędkości wprowadzania tekstu jest zwykle liczba wprowadzanych słów na minutę (ang. *words per minute*, wpm). Alternatywną, również często stosowaną jednostką jest liczba znaków na minutę. Należy jednak uściślić, czy chodzi o wszystkie znaki wprowadzone przez użytkownika, czy tylko o poprawne znaki tj. czy prędkością jest iloraz ostatecznej długości wprowadzonego tekstu do czasu, jaki zajęło jego wprowadzenie, czy iloraz wszystkich wprowadzeń, włączając w to błędnie wpisane znaki i ich usuwanie. Druga miara szybkości jest zawsze większa od pierwszej i pokazuje techniczną łatwość korzystania z klawiatury wzrokowej, ale to pierwsza lepiej oddaje praktyczną użyteczność klawiatury. Tej miary będziemy używać w tej pracy, chyba że zaznaczone zostanie inaczej.

Szybkość wpisywania można optymalizować zarówno ułatwiając sam akt wprowadzania znaku, jak i przez zmniejszenie liczby popełnianych błędów. Przy ustalonej długości tekstu do wpisania przez badanego, optymalizacje te będą wpływać na czas wprowadzania tekstu, który jest kluczową, obok liczby popełnionych błędów, obiektywną miarą użyteczności projektowanego systemu. Jest to miara, która odzwierciedla wiele czynników, które wpływają na wpisywanie wzrokowe, ale również, czego nie wolno pomijać, na komfort użytkownika. Te czynniki to między innymi: przypadkowe aktywacje elementów interfejsu, ergonomia układu klawiatury, czas akceptacji wprowadzania przycisku, a także bardzo ważne: rodzaj informacji zwrotnej. Poniżej postaram się omówić te czynniki bardziej szczegółowo.

Czas aktywacji (ang. *dwell-time*)

Interfejsy sterowane wzrokiem wymagają takich ruchów oka, które są łatwe do odróżnienia od ruchów naturalnych, które mają miejsce np. podczas eksploracji zawartości ekranu. To ma zapobiec niezamierzonym aktywacjom elementów interfejsu. Jednym z często stosowanych sposobów pozwalających na potwierdzenie intencji użytkownika jest wymóg dłuższego przytrzymania spojrzenia na wyróżnionym obszarze interfejsu graficznego w celu uruchomienia jakiejś akcji. Czas aktywacji (ang. *dwell-time*) w tym przypadku powinien być wyraźnie dłuższy, niż czas typowej fiksacji, ale warto podkreślić, że ten ostatni silnie zależy od wykonywanego zadania i może wynosić od ułamka sekundy, aż do kilku sekund. Sterowanie oparte na czasie aktywacji (ang. *dwell-time based control*) jest podstawową i najczęściej stosowaną formą kontroli w interakcji wzrokowej ze względu na poczucie pewności, jakie daje użytkownikowi. Niska liczba błędów jest jednak opłacona mniejszą wydajnością. W przypadku wzrokowego wprowadzania tekstu jest to szczególnie dotkliwe, ponieważ minimalny czas na wprowadzenie pewnej liczby liter równy jest iloczynowi tej liczby i czasu akceptacji. To ograniczenie można oczywiście próbować obejść np.

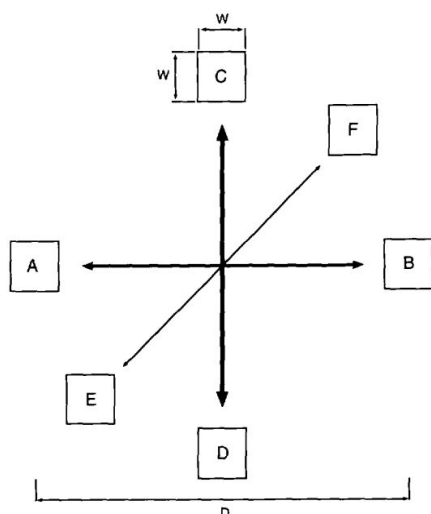
przez wykorzystanie mechanizmu sugestii, ale nawet wówczas sterowanie oparte na czasie akceptacji zwykle jest wolniejsze od innych metod.

Czas aktywacji w interakcji wzrokowej zwykle mieści się w granicy od 400 do 900 ms (Majaranta i inni, 2006). Z badań wynika, że im dłuższy czas aktywacji, tym większa dokładność wpisywania, czyli mniejsza liczba błędów popełnianych przez użytkownika, choć uzyskana kosztem mniejszego komfortu użytkowania (Majaranta i inni, 2003). Optymalny czas aktywacji pozwala na uzyskanie tzw. *flow*, w którym użytkownik mocno skupia się na wykonywanej czynności i popełnia mniej błędów. Prawdziwa jest również zależność odwrotna: im mniejszy czas aktywacji, tym większe jest prawdopodobieństwo wystąpienia pomyłek (Majaranta i inni, 2003). To szczególny przykład tzw. kompromisu szybkość-dokładność (ang. *speed-accuracy trade off*) w interakcji wzrokowej.

Tą zależność sprawdzono na przykład w badaniu, w którym uczestnicy mieli za zadanie patrzeć w ciągu dwóch minut naprzemiennie na dwa obiekty, zmieniając obserwowany obiekt tak szybko, jak to możliwe (Chi & Lin, 1997). Bódźce były zróżnicowane pod względem:

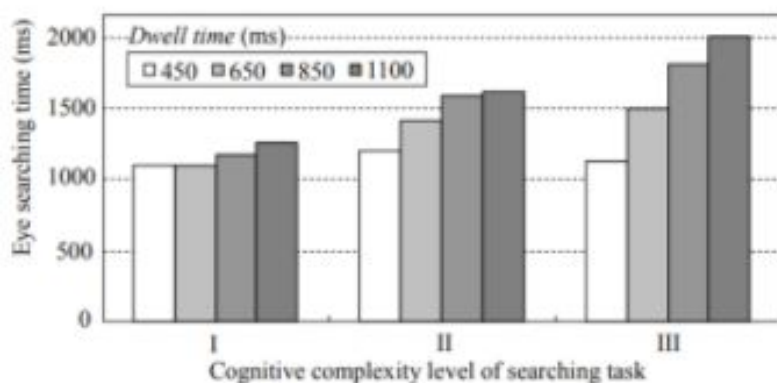
- wielkości - kwadraty o wielkości 1 cm i 2 cm,
- odległości między obiektami - 4 cm i 16 cm.
- kierunku ruchów oczu - poziomo, pionowo, po przekątnej (rys. 1.1).

Wyniki pokazują, że odległości między celami ($F = 4,95$, $p < 0,05$) i początkowy kierunek ruchu oczu ($F = 5,68$, $p < 0,05$) miały statystycznie istotny wpływ na czas ruchu oczu. Opierając się na prawie Fittsa (Laubheimer P., 2019), które mówi o tym, że czas potrzebny do osiągnięcia celu, także w przypadku wzroku, jest zależny od rozmiarów i odległości obiektu, który jest celem, można przypuszczać, że większa odległość między celami musi zwiększać czas ruchu oczu. Natomiast w tym doświadczeniu większa szerokość celu powodowała dłuższy czas ruchu gałek ocznych, co jest wyraźnie niezgodne z prawem Fittsa.



Rys. 1.1. Trzy zestawy symetrycznego celu (Chi & Lin, 1997; Zhang i inni, 2011)

W kolejnym badaniu Zhang i inni (2011) sprawdzili czy wydłużenie czasu aktywacji (*dwell-time*) ma wpływ na dokładność wykonywania zadań, które wykorzystują ten sposób interakcji wzrokowej. Za pomocą kilku zadań badano trzy następujące aspekty: obciążenie poznawcze, czas rzeczywistego przebywania spojrzenia w obrębie elementów interaktywnych oraz rodzaj sprzężenia zwrotnego.









Rys. 1.2 Czas szukania w zależności od skomplikowania wykonywanego zadania (Zhang i inni, 2011)

Wyniki wskazują na to, że wraz ze wzrostem obciążenia poznawczego wydłuża się czas wyszukiwania i czas przebywania w obszarze (**rys. 1.2**).

Zmniejszenie czasu aktywacji generuje jeszcze jeden problem, który w literaturze nazywany jest problemem dotyku Midasa (ang. *Midas touch problem*). Ujawnia się podczas przeszukiwania wzrokiem interfejsu w celu znalezienia właściwego przycisku. Temu zagadnieniu poświęcę jednak osobny podrozdział poniżej.

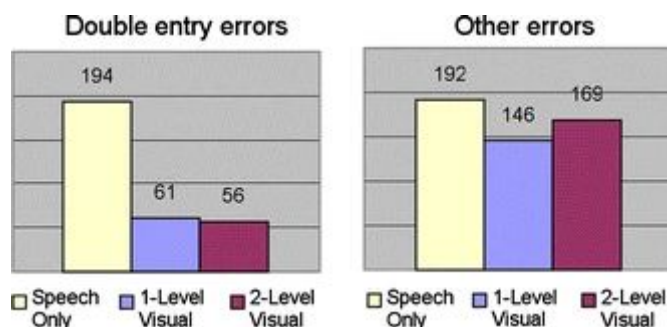
Kolejne badanie pokazuje różnice w liczbie popełnianych błędów w zależności od czasu aktywacji, który był równy 450 i 900 ms (Majaranta i inni, 2006). Przypomnę, że czas aktywacji to czas przez jaki należy utrzymać spojrzenie w obrębie elementu interaktywnego (przycisku wzrokowego), aby nastąpiło “kliknięcie”. W badaniu podzielono próby ze względu na typy informacji zwrotnej (**rys. 1.3**):

- tylko mowa (symbol na klawiszu jest odczytywany przy wyborze),
- wizualizacja jednopoziomowa (tło klawisza zmienia się na kolor czerwony po dokonaniu wyboru),
- wizualizacja dwupoziomowa (klawisz jest podświetlony przy wejściu w obszar elementu (ang. *focused*), a po dokonaniu wyboru (ang. *selected*) tło wprowadzanego przycisku zmienia się na kolor czerwony).

Feedback mode	While focused	When selected
Speech Only	none	letter spoken
1-Level Visual 	none 	red background 
2-Level Visual 	highlight 	red background 

Rys. 1.3. Rodzaje sprzężeń zwrotnych (Majaranta i inni, 2006)

Najwięcej błędów powodowanych niezamierzonymi podwójnymi kliknięciami przy ustawieniu czasu aktywacji na 450 ms występowało dla trybu sprzężenia zwrotnego “tylko mowa” (**rys. 1.4**) (Majaranta i inni, 2006).



Rys. 1.4: Błędy spowodowane niechcianym kliknięciem w zależności od rodzaju sprzężenia zwrotnego. (Majaranta i inni, 2006)

System wzrokowego wprowadzania tekstu badany w opisywanym eksperymencie zawiera zabezpieczenie przed nieumyślnym ponownym aktywowaniem tego samego obszaru. Po aktywacji przycisku dodaje się do podstawowego czasu aktywacji 450 ms dodatkowe 150 ms, jeżeli powtórnie wybrany ma być ten sam przycisk (czas powtórnej aktywacji) (Majaranta i inni, 2006).

Warto również wspomnieć o wzrokowych systemach wprowadzania tekstu, które dopasowują czas aktywacji do liczby błędów popełnianych przez użytkownika. W badaniu Majaranty i innych (2009) użyto takiej metody, co szybkość wprowadzania z 6.9 wpm aż do 19.9 wpm. W efekcie działania systemu adaptacji czas akceptacji spadł średnio z 876 ms do 282 ms, a liczba błędów spadła z 1.28% do 0.36%.

Informacja zwrotna

Informacja zwrotna (ang. *feedback*) informuje użytkownika o aktualnym stanie obsługiwanej przez niego aplikacji i efekcie działań użytkownika. Jej wpływ był widoczny na przykład w przedstawionym wyżej doświadczeniu Majaranty i in. (2003). Informacja zwrotna może m.in. wspomagać skupienie użytkownika na wykonywanej czynności. W interfejsach sterowanych wzrokowo może być użytych wiele rodzajów feedbacku, ale wszystkie one powinny zachowywać tzw.

zasadę akcji i reakcji. A mianowicie, gdy użytkownik wykonuje akcję, musi otrzymać informację o tym, w jakim stanie znajduje się element, z którym podjął interakcję. To pozwala m.in. na zrobienie szybkiej korekty np. w przypadku pomyłkowej aktywacji elementu.

W przypadku “przycisków wzrokowych”, na których skupiam się w tej pracy, dobrym przykładem informacji zwrotnej jest zmiana koloru przycisku, jeżeli spojrzenie znajduje się w obszarze przycisku, tak jak w badaniu Majaranty i in. (2003), jak również po jego “kliknięciu”.

Moim zdaniem można wyróżnić trzy podstawowe etapy informacji zwrotnej wspólne dla różnych metod interakcji wzrokowej (por. **rys. 1.5**), które zostaną tu omówione na przykładzie przycisku wzrokowego:

1. Inicjalizacja

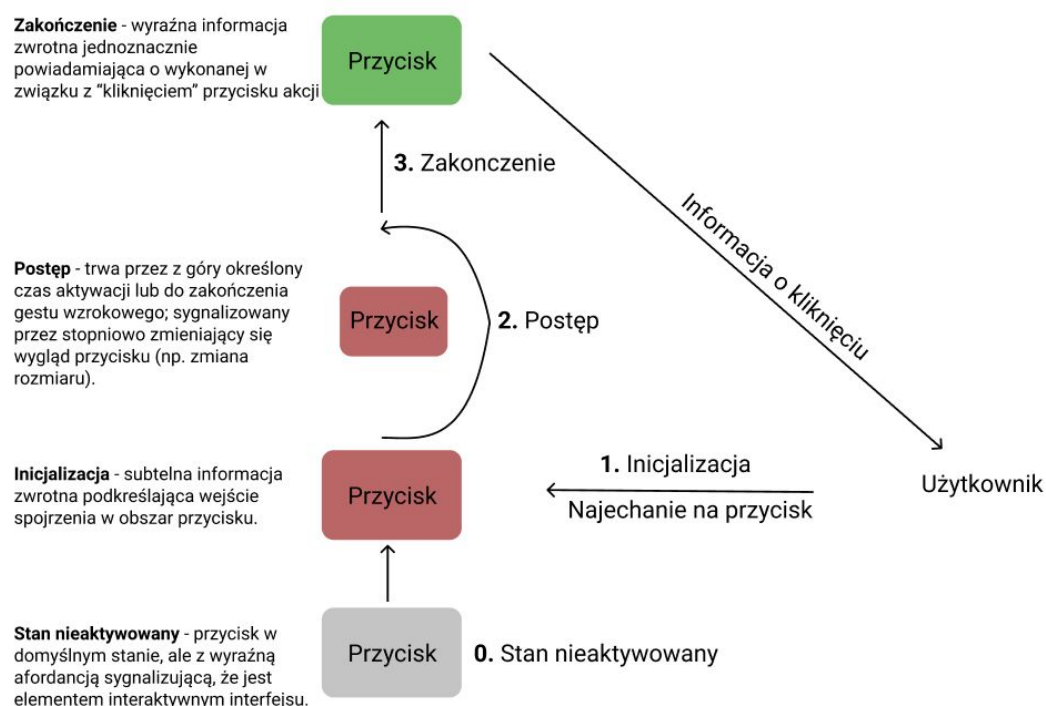
Inicjalizacja następuje w momencie, gdy wzrok wchodzi w obszar przycisku. Fakt ten może być zasygnalizowany delikatną zmianą wyglądu przycisku. Inicjalizacja nie powinna prowadzić do natychmiastowego kliknięcia, gdyż będzie to wiązało się z dużą liczbą niezamierzonych akcji. Powinna natomiast zapoczątkowywać dłuższy proces, podczas którego oczekiwane jest utrzymanie wzroku na przycisku przez czas aktywacji lub wykonanie dodatkowych ruchów oczu przez użytkownika (tzw. gesty wzrokowe).

2. Postęp

Po inicjalizacji przycisku następuje czas aktywacji, po którym przycisk zostanie kliknięty. Przycisk powinien sygnalizować upływ tego czasu stopniowo zmieniając kolor, rozmiar, prezentując pasek postępu lub w inny ciągły sposób. To z jednej strony pozwoli na wycofanie się w porę z niechcianej interakcji, a z drugiej pomaga dodatkowo skupić wzrok użytkownika ułatwiając utrzymanie spojrzenia przez czas aktywacji.

3. Zakończenie

Zakończenie całego procesu kliknięcia wiąże się z wykonaniem akcji związanej z przyciskiem np. wprowadzenie litery lub cyfry. Potwierdzeniem tego może być sygnał dźwiękowy (np. odczytanie litery) albo zmiana koloru przycisku na pewien czas przed przywróceniem koloru odpowiedniego dla przycisku w stanie wyjściowym.



Rys. 1.5. Schemat działania feedbacku.

W interfejsach wzrokowych wykorzystuje się przede wszystkim wizualne i dźwiękowe informacje zwrotne. Oba typy można stosować wspólnie w zależności od projektu i indywidualnych potrzeb użytkowników. Majaranta i inni (2006) sprawdzili właśnie te dwa rodzaje informacji zwrotnej w czterech wariantach:

- tylko wizualny (przycisk jest podświetlany, gdy skupia się na nim wzrok, a jego symbol kurczy się wraz z upływem czasu oczekiwania; kurczenie przyciąga uwagę, pomagając użytkownikowi skoncentrować się na środku przycisku; po wybraniu litera zmienia się do stanu początkowego),
- tylko dźwiękowy (podczas kliknięcia litery, jest ona wypowiedzana, co oznacza, że *feedback* pojawia się w trzecim etapie i dotyczy nie tylko faktu wprowadzenia, ale też wprowadzonego znaku),
- wizualny i dźwiękowy (połączono wizualną informację zwrotną z krótkim dźwiękiem przy akceptacji kliknięcia).
- dźwiękowy z dodatkowym wizualnym (wykorzystano wizualną informację zwrotną, a na końcu kliknięcia była wypowiedzana wprowadzana litera).

Problem dotyku Midasa

“Najsłynniejszy król Midas jest znany w mitologii greckiej ze względu na swoją zdolność do przekształcania wszystkiego, czego dotknął, w złoto. Nazywa się to *złotym dotykiem* lub *dotykiem Midasa*.” (Treker, 2020) W przypadku eyetrackingu problem dotyku Midasa dotyczy jednak trochę czegoś innego (Jacob i inni 1995). Spojrzenie nie powoduje bowiem zmiany monitora albo przycisku w złoto, a oznacza, przypadkowe niezamierzone kliknięcia jakie wykonuje użytkownik podczas obserwacji zawartości ekranu lub podczas przenoszenia spojrzenia do docelowego przycisku, na drodze do którego znajduje się inny przycisk. Wówczas jego niezamierzona aktywacja spowoduje, że trzeba wykonać dodatkową czynność usunięcia niechcianego wprowadzenia.

Problem dotyku Midasa można zmniejszyć manipulując czasem aktywacji oraz wzbogacając informację zwrotną, co było przedmiotem wcześniej wspomnianego badania (Majaranta i inni, 2006). Jeżeli wydłużymy czas kliknięcia, to użytkownik, który będzie eksplorował ekran jest mniej narażony na to, że przytrzyma intencjonalnie wzrok na jednym przycisku dłużej niż równy jest jego czas aktywacji. Jednak wydłużenie czasu aktywacji też nie jest pożądanym efektem, bo zmniejsza efektywność interakcji wzrokowej.

Standardowym rozwiązaniem problemu dotyku Midasa jest system rozpoznawania tego, czy spojrzenia są intencjonalne czy nieintencjonalne, w którym wykorzystuje się czas ich utrzymywania wzroku w pewnym obszarze (ang. *dwell time*). W literaturze przedmiotu ten czas zwykle ustawiany jest na ok. 450-500 ms (B. Velichkovsky i inni, 1997), jednak dla nowicjuszy powinien być dłuższy. Później rozszerzono tę klasyfikację o rozróżnienie fiksacji na dwa rodzaje: ogniskową oraz peryferyjną (B. B. Velichkovsky i inni, 2014). Wykorzystano przy tym wiedzę na temat strumienia brzuszno i grzbietowego w korze mózgowej, a mianowicie to, że praca każdego z tych strumieni odpowiada specyficznej aktywności oczu. Obserwacji otoczenia towarzyszą sakkady o wysokiej amplitudzie i krótkich fiksacjach, przez co człowiek nie jest w stanie dobrze zidentyfikować obiektu, ale jest w stanie objąć spojrzeniem większy obszar. Jest to tzw. widzenie dla działania. Odpowiada za nie strumień grzbietowy. Natomiast długie fiksacje i sakkady o małej częstotliwości wiążą się raczej z tzw. widzeniem dla percepcji, za które odpowiada strumień brzuszny. Autorom pracy udało się wychwycić różnice w tych dwóch rodzajach postrzegania i je zidentyfikować w czasie rzeczywistym. To pozwoliło na uniknięcie wielu błędów spowodowanych niepewnością odczytywania intencji użytkownika. To z kolei doprowadziło do mniejszego zmęczenia wzroku i większej prędkości pracy (B. B. Velichkovsky i inni, 2014).

Układ klawiatury

Układ interfejsu wzrokowego aplikacji (ang. *layout*) można modyfikować w taki sposób, aby poprawiać jego ergonomiczność. Zmianie podlegają: liczba przycisków, odległość między nimi, ich wielkość i rozmieszczenie.

Poważnym problemem jest duża liczba liter w alfabecie, która wymusza dużą liczbę przycisków na klawiaturze fizycznej, jak również na klawiaturze wzrokowej, która ją imituje. Problem ten dobrze opisali Bee i Andre (2008) pisząc: *“Spośród systemów bez probabilistycznego przewidywania liter lub uzupełniania słów, systemy oparte na klawiaturze sterowanej spojrzeniem są najszybsze, ponieważ wprowadzenie litery zajmuje tylko jeden krok. Takie systemy muszą wyświetlać wszystkie litery jednocześnie na ekranie. Alfabet ma 26 liter do których można dodać przyciski poleceń (np. spacja, usuń itp.). Wymagane jest miejsce na co najmniej 30 przycisków. W zależności od dokładności urządzenia do śledzenia wzroku, przyciski wymagają określonego rozmiaru. Ostatecznie interfejs do pisania będzie wymagał dużo miejsca.”*

Jest wiele sposobów na rozmieszczenie przycisków na ekranie. Jedne wykorzystują cały ekran, aby wykorzystać jak największą przestrzeń. Inne wykorzystują różne sposoby rozmieszczenia przycisków jak: kołowe (Huckauf & Urbina, 2007) albo na krawędziach prostokąta (Majaranta i inni, 2006). Zmniejszenie liczby klawiszy i odległości między klawiszami można wykorzystać do zaoszczędzenia miejsca na ekranie (Miniotas i inni, 2003).

Klawiatura nie musi być statyczna, a może się dopasowywać do działań użytkownika (Panwar i inni, 2012; B. B. Velichkovsky i inni, 2014). Stworzono na przykład oparty na spojrzeniu system wprowadzania tekstu, który optymalizuje klawiaturę w odniesieniu do ruchu wzroku, a także dynamicznie dostosowuje czas przebywania zgodnie z poziomem komfortu użytkownika (Panwar i inni, 2012; B. B. Velichkovsky i inni, 2014).

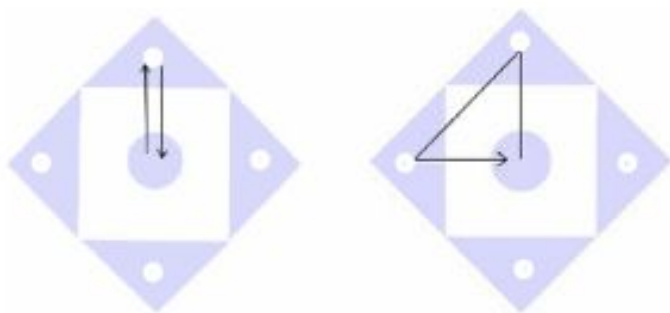
Istnieją również systemy, które projektowane były z myślą o realizacji konkretnego celu. Przykładem jest tradycyjny układ klawiszy *Qwerty*, który został zaprojektowany do maszyn do pisania tak, aby zminimalizować zacinać się klawiszy. Z kolei układ *Dvorak* jest dostosowany do częstotliwości używanych liter. Układ ten próbuje zminimalizować ruchy dłoni i palców, z naciskiem na prawą rękę. Układ *Fitaly* został zaprojektowany w celu optymalizacji ruchów myszy podczas wprowadzania tekstu jednym palcem, rysikiem lub piórem. W układzie tym występują dwa klawisze spacji, zachowując przy tym bliskość najczęściej używanych liter w języku angielskim (E, T, A, H). Z kolei *Opti* to zoptymalizowany układ wirtualnej klawiatury dostosowana do języka angielskiego. Natomiast *Lewis* to układ, w którym przyciski ułożone są w siatce 5 × 6, w której klawisze są umieszczane alfabetycznie (Zhai i inni, 2002).

Gesty

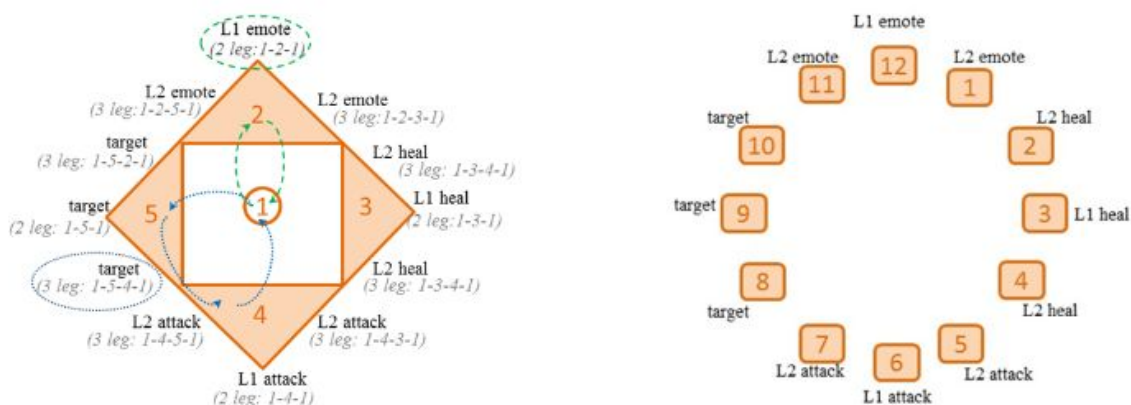
Do interakcji z komputerem wykorzystuje się również gesty. Gesty mogą być związane z ekranem (z umieszczonymi na nim “przewodnikami dla oczu”) lub zupełnie niezwiązane, wykorzystujące ścieżki spojrzeń wymaganych do zakończenia gestów (np. ruch w górę i potem w prawo). W obu przypadkach liczba łatwych do zapamiętania i wykonania gestów jest ograniczona. Nie można wobec tego zanadto zwiększać ich liczby, bo ogranicza to liczbę operacji, jakie można z ich pomocą wykonać. W przypadku gestów wykonywanych w odniesieniu do elementów widocznych na ekranie, problemem może być też konieczność kompromisu między liczbą tych elementów i ich rozmiarem. Zbyt niewielkie rozmiary mogą skutkować przypadkowymi wprowadzeniami (Drewes & SchmiDwell Time, 2007; Zhai i inni, 2002).

W badaniach, w których wykorzystano metodę gestów, polegającą na tym, że gest zawsze zaczyna się od okrągłego obszaru w środku i przechodzi przez jeden albo dwa trójkątne obszary (**rys. 1.6**), a wszystkie gesty kończą się fiksacją z

powrotem w okrągłym obszarze pokazano, że wprowadzenie wzrokowe za pomocą gestów jest dobrą alternatywą dla wprowadzania za pomocą czasu aktywacji (*dwell-time*) (Hyrskykari i inni, 2012; Istance i inni, 2010). Podczas tego badania uczestnicy grali w grę, w której używali gestów albo przycisków *dwell-time* (**rys. 1.7**). Rozpoznawanych było 12 gestów (cztery dwupunktowe i osiem trójpunktowych), co było alternatywą do 12 przycisków *dwell-time*. Różnica między czasem ukończenia zadania przy użyciu gestów dwupunktowych i przycisków *dwell-time* nie była znacząca (test t-Studenta $p = 0,058$). Również różnica w czasie wykonania zadań w przypadku gestów trójpunktowych i przycisków *dwell-time* nie była znacząca (test t-Studenta $p = 0,56$).

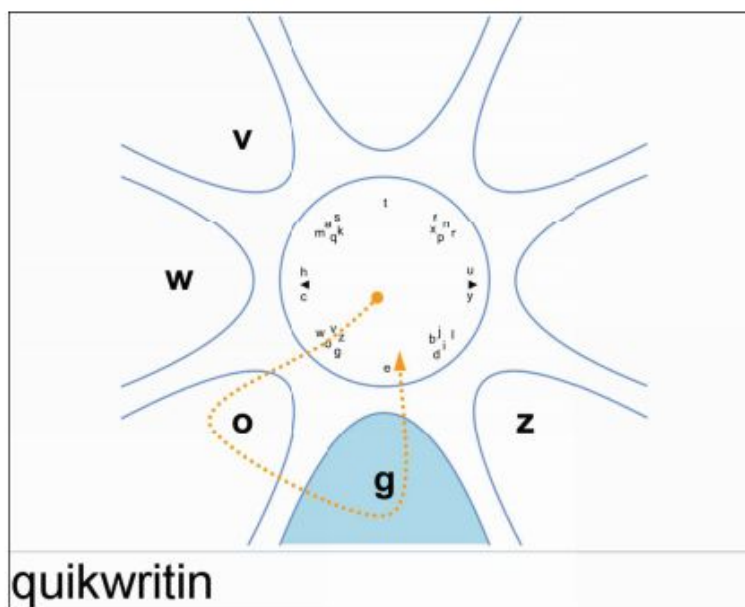


Rys. 1.6. Dwa rodzaje gestów. Po lewej gest dwupunktowy, a po prawej trójpunktowy. "Punkt" oznacza miejsce, jakie musi być odwiedzone przez spojrzenie w trakcie wykonywania gestu (Hyrskykari i inni, 2012; Istance i inni, 2010)



Rys. 1.7. Akcje jakie można wykonać przy pomocy gestów (po lewej) i Dwell Time (po prawej) (Hyrskykari i inni, 2012; Istance i inni, 2010)

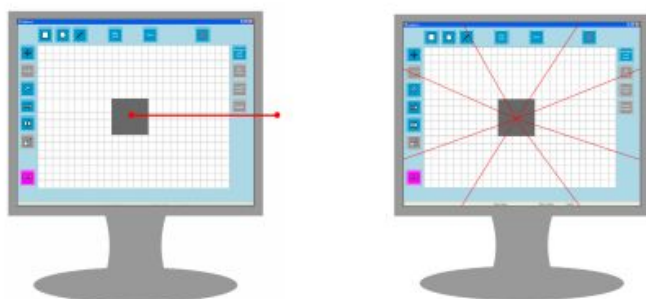
Metoda interakcji wzrokowej używana w badaniu Hyrskykari i innych (2012) bardzo przypomina system *QuickWriting* użyty przez Bee i André (2008), który wcześniej został stworzony przez Perlin (1998). Isokoski (2000) opisał jak używać tego systemu w kontekście wprowadzania wzrokowego. Został on pierwotnie stworzony do pisania za pomocą rysika, ale skutecznie można z niego korzystać także za pomocą wzroku. Po modyfikacji oryginalnego *QuickWriting* (**rys 1.8**) badani wprowadzali litery z szybkością 5 wpm, ale ci sami badani osiągnęli szybkość wpisywania 8 wpm za pomocą układu *Qwerty* (Bee & André, 2008)



Rys 1.8. Przerywana linia oznacza ścieżkę wzrokową, a użytkownik patrzy obecnie na literę “g” co podpowiada mu podświetlenie przycisku. (Bee & André, 2008)

Wprowadzanie tekstu nie jest jedynym sposobem wykorzystania gestów wzrokowych. Projektowane są także systemy, które służą np. do rysowania. W systemie stworzonym przez Heikkilä (2013) wykorzystuje się modyfikowalne i ruchome kształty (**rys. 1.9**). Dostępne są narzędzia takie jak *Move*, *Nudge* i *Resize*, które są inicjowane i kontrolowane właśnie za pomocą gestów. Narzędzie *Nudge* przesuwa kształt o jeden kwadrat siatki widocznej na ekranie w kierunku

wykonanego gestu. Za pomocą narzędzia Move można przesunąć kształt w kierunku wskazanym przez gest. W tym przypadku ruch zakończy się jednak dopiero, gdy użytkownik zamknie oczy lub gdy kształt zetknie się z krawędzią obszaru rysowania. Z kolei narzędzie Resize powoduje zmianę rozmiaru kształtu: po wybraniu kształtu (zaznaczeniu go), po jego bokach pojawiają się uchwyty. Uchwyty jest wybierany metodą *dwell-time* o czasie aktywacji 100 ms, a następnie w zależności od kierunku gestu (do wewnątrz lub na zewnątrz) kształt kurczy się lub rośnie (Heikkilä, 2013).



Rys. 1.9. Po lewej – Gest wzrokowy zaczyna się od narysowanego kształtu i kończy poza ekranem. Po prawej – trzeba wzrokiem odwiedzić co najmniej trzy punkty, zanim będzie można wykonać gest (Heikkilä, 2013)

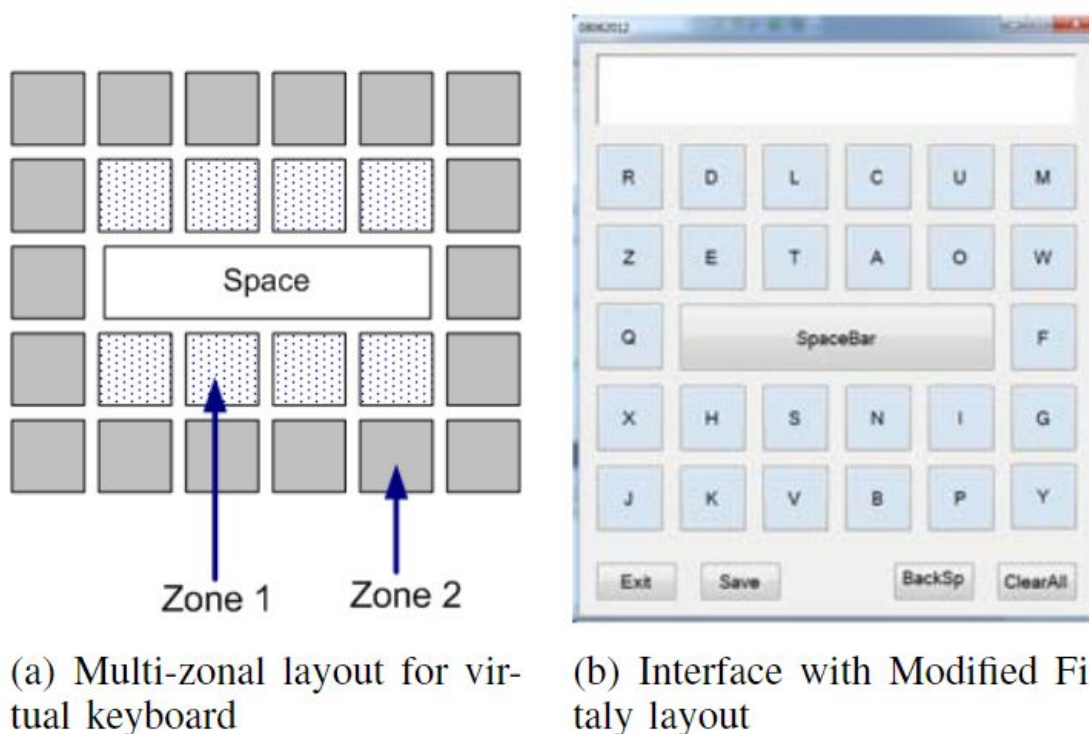
Przykłady interfejsów wzrokowych

Zaproponowano także wiele innych interfejsów wzrokowych, szczególnie do wzrokowego wprowadzania tekstu. Różnią się one omówionymi wyżej własnościami. Kilka wybranych, które nie były wspomniane we wcześniejszych podrozdziałach, pokrótce przedstawię poniżej.

Eyeboard

Jest to wirtualna klawiatura, której zadaniem jest optymalizowanie długości ruchu wzroku. Dynamicznie dostosowuje czas wprowadzania zgodnie z

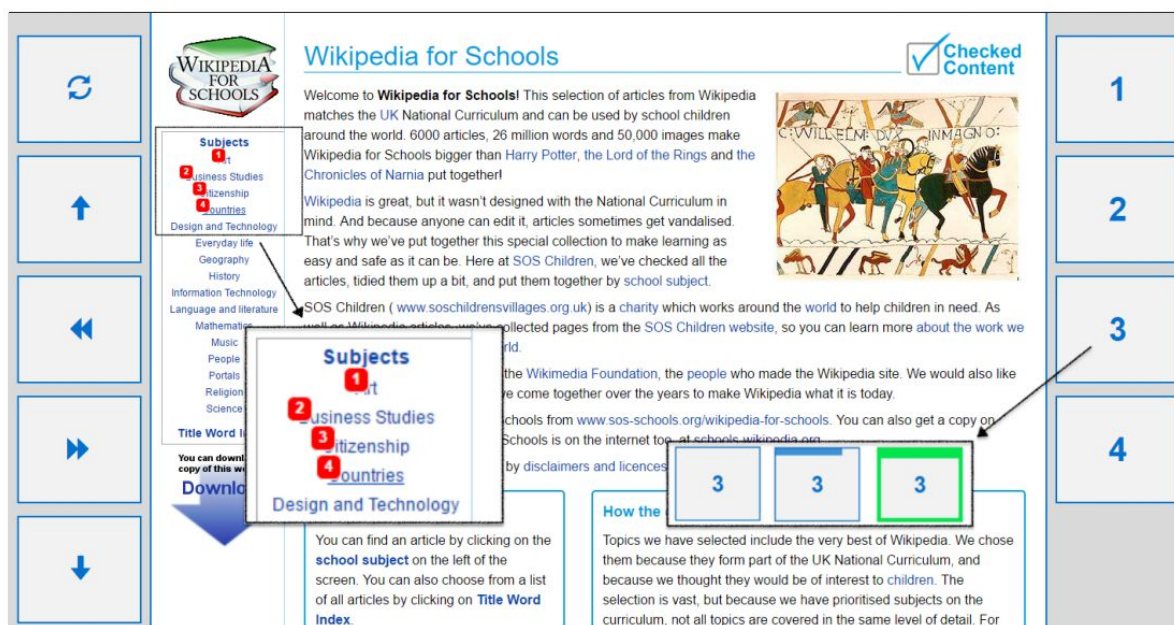
ustalonym w czasie rzeczywistym poziomem komfortu użytkownika. Optymalizacja długości ruchu oczu następuje poprzez zmianę wielkości klawiszy, odstępu między nimi i powiększania układu klawiatury (Panwar i inni, 2012).



Rys. 1.10. Schematyczne przedstawienie układu klawiaturowego Fitaly (Panwar i inni, 2012).

EyeLinks

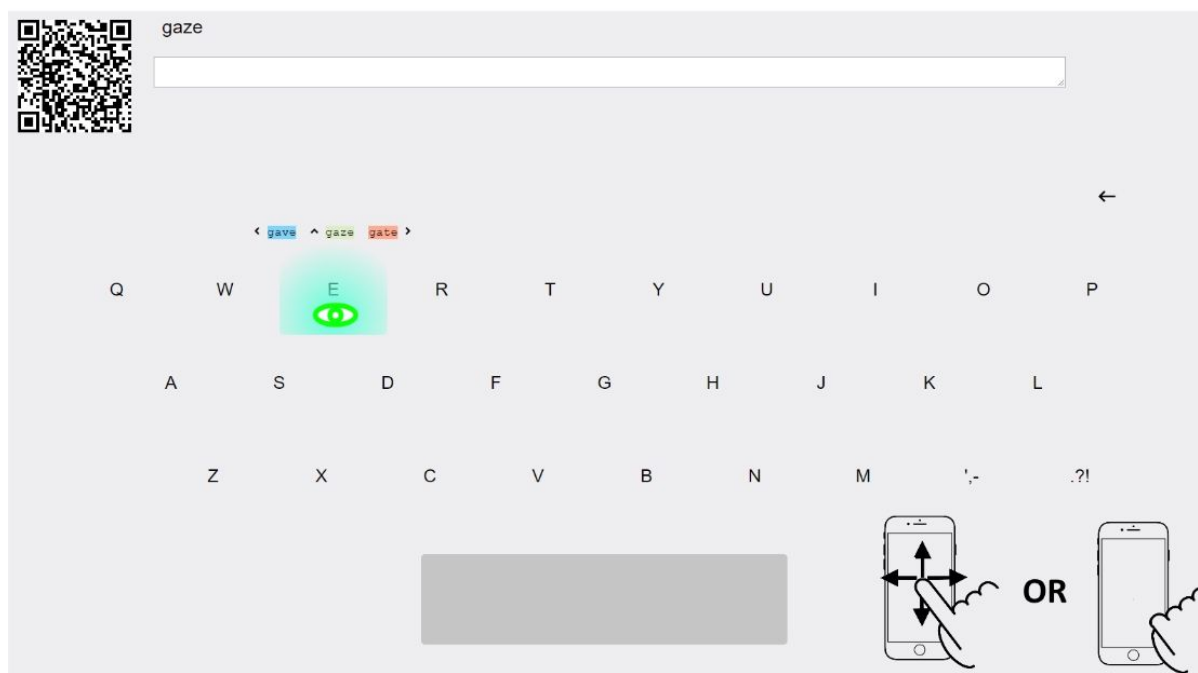
Aplikacja, ułatwia przeglądanie stron internetowych. Przypisuje identyfikatory numeryczne do elementów “klikalnych” w obszarze, na który aktualnie patrzy użytkownik. Te linki mają swoje odpowiedniki w postaci przycisków na pasku bocznym (**rys. 1.10**). Jeżeli użytkownik chce kliknąć link, robi to korzystając przycisku bocznego. To rozwiązanie umożliwia użytkownikom swobodne przeglądanie stron internetowych, unikając problemu dotyku Midasa i zwiększając dokładność (Figueiredo & Fonseca, 2018).



Rys 1.11 Użytkownik aktualnie spogląda na linki którym, zostały przypisane identyfikatory od jedynki do czwórki. Panel boczny (po prawej) zawiera przyciski odnoszące się do każdego z tych identyfikatorów.

Tag Swipe

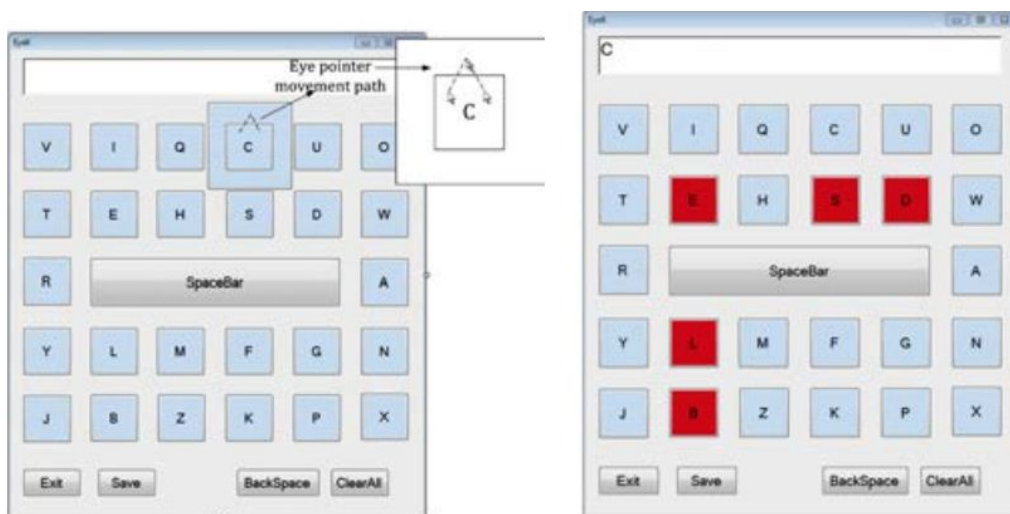
Ten system łączy interakcję wzrokową z wykorzystaniem dotyku palcem podczas wpisywania słowa. Za pomocą palca można zdecydować czy ruch ma być ruchem ciągłym czy też nie. Dzięki temu jest możliwe wprowadzanie tekstu bez przerw. Działa to podobnie jak byśmy na ekranie telefonu trzymali palec i jednym pociągnięciem wpisali całe słowo. Jeżeli wpisywane słowo znajdzie się na liście sugerowanych słów, gest dotykowy (przeciągnięcie w lewo lub w prawo) pozwala użytkownikowi wybrać alternatywne słowo. System rejestruje i porównuje ścieżkę spojrzenia użytkownika z idealną ścieżką słów, aby wygenerować najlepsze słowo, które jest wprowadzane automatycznie (Kumar i inni, 2020).



Rys. 1.11. System TagSwipe. Wybieranie przewidzianego słowa. (Kumar i inni, 2020).

EyeK

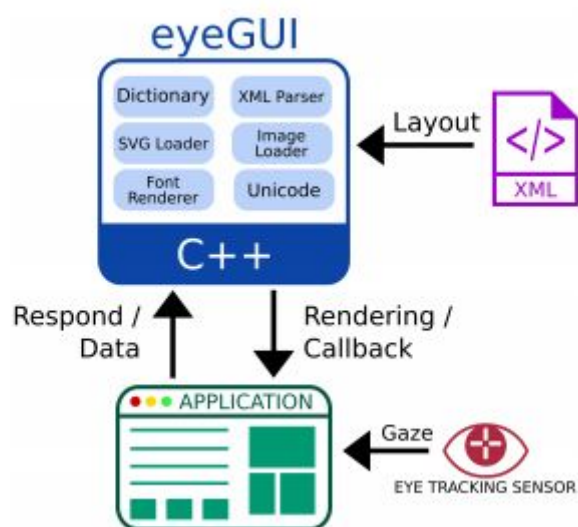
W tym interfejsie znajduje się 27 znaków rozmieszczonych w tabeli 5x6. Wiersz główny (trzeci wiersz) zawiera przycisk spacji, który jest cztery razy szerszy od zwykłego przycisku. Inne przyciski są ułożone zgodnie z zasadą, że częściej używane znaki są bliżej spacji. Podczas wpisywania jakiegoś słowa następuje podświetlenie znaków, których prawdopodobieństwo wybrania jest najwyższe (**rys. 1.12**). Na przykład po wpisaniu dwóch liter „KO” system EyeK wybiera kolejne prawdopodobne znaki jako „Ł”, „L”, „O”, „E” i „C” i koloruje je na czerwono (Sarcar i inni, 2013).



Rys. 1.12. Klawiatura aplikacji EyeK (Sarcar i inni, 2013).

eyeGUI

Strony i aplikacje internetowe nie zostają w tyle i również coraz częściej oferują możliwość obsługi za pomocą sterowania wzrokiem. W przypadku interfejsu eyeGUI (rys. 1.10) (Menges i inni, 2016) generowane są na stronie dodatkowe przyciski z boków, umożliwiające łatwiejsze użytkowanie za pomocą wzroku. Instalacja tego systemu na komputerze nie jest problemem, bo wystarczy zainstalować rozszerzenie do przeglądarki Chrome. Dzięki temu można używać tych samych elementów strony internetowej, co w zwykłej przeglądarce, ale za pomocą wzroku i kamery internetowej (xlabs, 2014).



Rys. 1.13. Schemat działania aplikacji eyeGUI (Menges i inni, 2016)

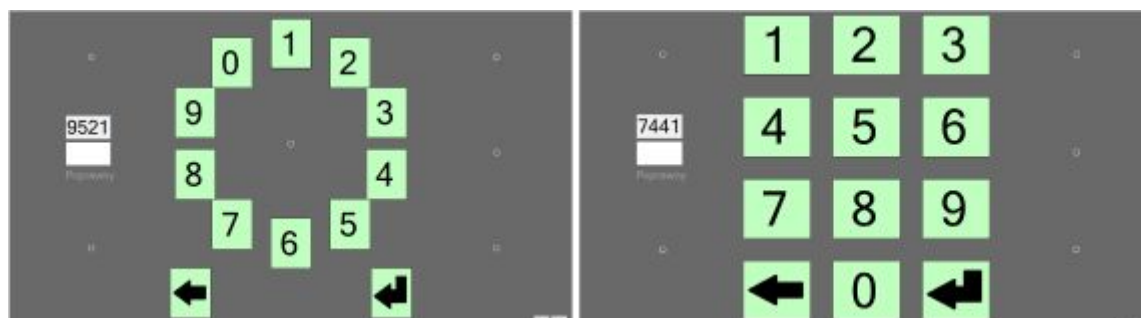
Rozdział 2

Problem badawczy i hipotezy

Celem tej pracy jest zbadanie wybranych metod wzrokowej aktywacji przycisków widocznych na ekranie. Aktywacja przycisku (w celu uruchomienia jakiejś czynności) to jedna z “atomowych” akcji, jakich potrzeba do projektowania interfejsów, także interfejsów wzrokowych. Pozwala użytkownikowi na zachowanie kontroli nad rozpoczęciem wskazanej przez niego czynności. Na potrzeby testów przygotowane zostało oprogramowanie, które wymaga od użytkownika kilkukrotnego wprowadzenia różnych czterocyfrowych numerów pin korzystając z zestawu przycisków wzrokowych. Przyciski z cyframi mogą być ułożone w dwojaki sposób: tak, jak w typowych klawiaturach cyfrowych (układ dalej nazywany *Telephon Pad*) lub na okręgu (układ *Circle*) (**rys. 2.1**). Program umożliwia także ustawienia wielkości przycisków i odległości między nimi, ale ta możliwość nie była wykorzystywana w opisywanym eksperymencie. Dodatkowo możliwe są trzy metody zatwierdzania:

- 1) oparte na czasie akceptacji (metoda nazywana będzie dalej *Dwell Time*),
- 2) bazujące na wykryciu fiksacji (*Fixation*),
- 3) oparte na geście wzrokowym “wejdz i wyjdź” (*Enter and Leave*).

Metoda *Fixation* wymaga przeprowadzenia analizy zdarzeń okoruchowych w czasie rzeczywistym.



Rys. 2.1. Dwa typy klawiatury numerycznej po lewej *Circle* a po prawej *Telephon Pad*.

Zwróćmy uwagę, że dwie z powyższych metod wychodzą poza najczęściej używany schemat metod z czasem akceptacji. Uwzględniając sposób ułożenia przycisków, otrzymujemy sześć kontrolowanych warunków eksperymentu (patrz **tab 3.1** w następnym rozdziale). Każdy uczestnik badania wykonywał zadania we wszystkich sześciu warunkach.

Naszym zasadniczym zamiarem było zbadanie możliwości przełamania kompromisu szybkość-precyzja (ang. *speed-accuracy trade off*), jednak ze świadomością tego, jak trudno jest nakłonić użytkowników, szczególnie nowicjuszy, do bardziej zaawansowanych metod interakcji wzrokowej. Potencjalne zastosowania badanych metod mogą dotyczyć zarówno osób niepełnosprawnych ruchowo, w przypadku których należy liczyć się z różnego rodzaju ograniczeniami, ale dla których sterowanie wzrokowe może być jedynym sposobem używania komputera lub nawet kontaktu z otoczeniem, jak i zdrowych, w przypadku których wprowadzanie osobistych numerów uwierzytelniania (pin) bez konieczności dotykania klawiatury może być ważne zarówno w kontekście bezpieczeństwa (brak odcisków palców), jak i zdrowia (unikanie miejsc przenoszenia wirusów, co nabrało szczególnego znaczenia w trakcie epidemii COVID-19).

Postanowiliśmy zweryfikować następujące hipotezy:

Hipoteza 1

Metoda *Enter and Leave* skróci czas wprowadzania liczb (mniejszy będzie średni czas wprowadzania cyfry).

Hipoteza 2

Metoda *Fixation* ma mniejszy czas wprowadzania od metody *Dwell Time*.

Hipoteza 3

Metody *Dwell Time* i *Fixation* będą podobne, pod względem liczby błędów.

Hipoteza 4

Liczba sakad w przypadku metody *Enter and Leave* będzie wyższa niż w przypadku innych metod, lecz nie będzie znaczących różnic między metodami *Fixation* i *Dwell Time*.

Hipoteza 5

Użyteczność deklarowana przez użytkowników będzie znacząco niższa w przypadku metody *Enter and Leave* w porównaniu z pozostałymi metodami.

Hipoteza 6

Średni czas fiksacji przy wprowadzaniu liczb metodą *Enter and Leave* będzie niższy niż w przypadku metody *Dwell Time*.

Hipoteza 7

Zmiana średnicy źrenicy będzie największa w przypadku metody *Enter and Leave*.

Hipoteza 8

Entropia będzie większa w przypadku metody *Enter and Leave* niż w pozostałych, natomiast pomiędzy metodami *Dwell Time* i *Fixation* nie będzie istotnej różnicy.

Rozdział 3

Procedura eksperymentu

3.1. Osoby badane i rekrutacja

W Badaniu wzięło udział 37 osób między 19 a 50 rokiem życia. Jeżeli badany musiał używać okularów, mógł to robić w trakcie badania pod warunkiem że walidacja potwierdziła dokładność wyznaczania położenia spojrzenia na ekranie z błędem mniejszym niż pół stopnia w obu kierunkach. Ruchy głowy nie były w żaden sposób ograniczone, ale badany był pouczany, żeby ograniczał je do minimum, podobnie jak ruchy ciała.

Analiza uwzględnia dane zebrane dla 31 badanych. W sześciu przypadkach, na różnych etapach badani zrezygnowali, przede wszystkim przez duże trudności z wprowadzaniem pinów, bądź błędy związane ze zrywaniem połączenia eyetrackera z komputerem i związane z tym zniechęcenie.

W badaniu wykorzystano następujący sprzęt:

- okulograf SMI (model RED 250),
- krzesło z oparciem,
- laptop DELL precision M4800 15.6" o rozdzielczości matrycy 1920×1080,
- aplikacja *GazeButtonsExperiment* przygotowana do eksperymentu przez promotora (wersja programu: 1.0.6995.26983).

Procedura badania

Badanemu została przedstawiona instrukcja korzystania z programu i metod aktywacji przycisków (**zał. 2**) i informacja dla osoby badanej (**zał. 4**). Następnie zbierane były zgody na uczestnictwo w badaniu (**zał. 3**). W instrukcji

badany został poinstruowany, że będzie miał za zadanie aktywować (“klikać”) przyciski za pomocą wzroku i będzie to robił trzema metodami.

Po zapoznaniu się z instrukcją i podpisaniu zgody, badany siadał na krześle naprzeciw laptopa z eyetrackerem w odstępnie 65-70 cm od eyetrackera. Pokój był pozbawiony światła słonecznego. Światło padało z lampy, która była przymocowana na stałe w prawym górnym rogu pola widzenia badanego. Następnie przeprowadzana była kalibracja eyetrackera, a po niej także walidacja. Jeżeli kalibracja przeszła pomyślnie, uruchamiana była wersja demonstracyjna badania, w której testował dwie metody: z czasem aktywacji (ang. *dwell time*) oraz z gestem “wejdź i wyjdź”. W trakcie części demonstracyjnej badany miał możliwość zadawania pytań dotyczących działania metod, potem proszony był o zachowanie ciszy. Po zakończeniu części demonstracyjnej włączany był właściwy eksperyment.

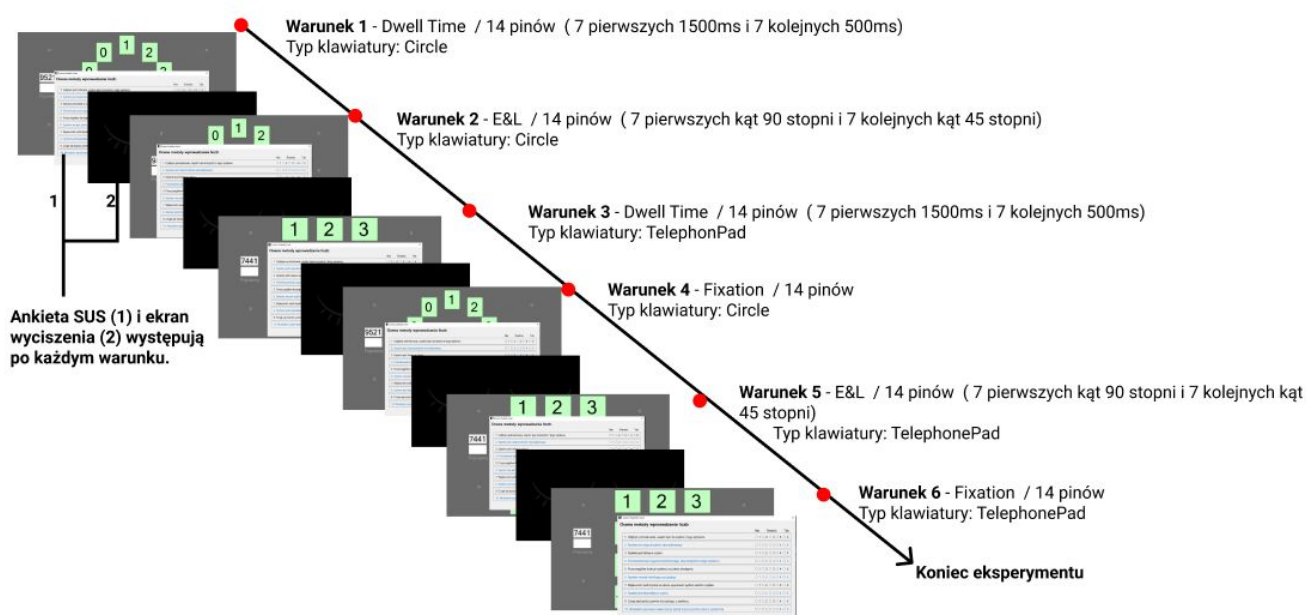
Jak wspomniano wyżej, w eksperymencie wykorzystanych było sześć warunków (**tab 2**) po 14 prób każdy. Każda próba wymagała wpisania czterocyfrowego pinu który był losowany z bazy czternastu liczb: 1793, 1719, 3971, 9713, 1397, 7147, 7417, 7411, 7441, 1295, 9512, 9125, 9512, 9152 i czterech liczb w przypadku wersji demonstracyjnej: 1234, 1972, 2018, 6728. W ciągu całego badania badany miał wobec tego wpisać 336 ($6 \times 14 \times 4$) cyfr. Numery pin były widoczne na ekranie, a poza tym były wypowiedzane przez syntezytor mowy systemu Windows (jeden raz w momencie pojawienia się nowego pinu na ekranie). Dodatkowo sygnalizowany był moment, w którym wprowadzony pin jest prawidłowy (zmiana na zielony koloru napisu “poprawny” widocznego pod polem z podglądem wpisywanych liczb).

l.p	Metoda	Czas aktywacji	Układ klawiatury	Feedback
1.	<i>Dwell Time</i>	500 ms i 1500ms	<i>Circle</i>	przy inicjalizacji i przy kliknięciu

2.	<i>Dwell Time</i>	1500 ms lub 1500ms	<i>Telephon Pad</i>	przy inicjalizacji i przy kliknięciu
3.	<i>Enter and Leave</i>	500ms - 1000ms	<i>Circle</i>	przy kliknięciu
4.	<i>Enter and Leave</i>	500ms - 1000ms	<i>Telephon Pad</i>	przy kliknięciu
5.	<i>Fixation</i>	nie dotyczy	<i>Circle</i>	przy inicjalizacji i przy kliknięciu
6.	<i>Fixation</i>	nie dotyczy	<i>Telephon Pad</i>	przy inicjalizacji i przy kliknięciu

Tab. 3.1. Warunki eksperymentalne. W kolumnie czas aktywacji występują czasy potrzebne do kliknięcia przycisku.

Przykładowy przebieg eksperymentu. Każdemu badaniemu została losowana kolejność warunków.



Rys. 3.1. Schemat przebiegu eksperymentu. Kolejność warunków była losowana dla każdego badanego.

Badanie składało się z sześciu sekwencji, odpowiadających poszczególnym warunkom eksperymentu, składających się z 14 prób. W przypadku metody *Dwell Time* pierwsze 7 prób były ustawione na czas wprowadzania 1500 ms, a drugie - 500 ms. W metodzie *Enter and Leave* pierwsze 7 prób miały ustawione kąt pomiędzy ruchami wejścia i wyjścia 45 stopni, a 7 ostatnich - 90 stopni. Kąt

między ruchami wejścia i wyjścia jest jednym z warunków do spełnienia, aby nastąpiło kliknięcie (zob. podrozdział 3.2. Metoda “wejdź i wyjdź”). Kolejność prób, jak i kolejność pinów do wpisania była losowa. Badany miał możliwość przerwania eksperymentu na każdym etapie, ale również badacz miał możliwość przerwania go, jeżeli wprowadzanie liczb aktualną metodą sprawiało zbyt duże trudności i nie było szansy na dokończenie sekwencji prób.

Po każdej sekwencji prób (tj. przed zmianą metody) badany wypełniał ankietę skali użyteczności systemu (SUS), złożoną z dziesięciu pytań ze skalą Likerta (**rys. 3.2**).

	Nie	Średnio	Tak
1. Gdybym potrzebował, często bym korzystał z tego systemu.	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
2. System jest niepotrzebnie skomplikowany.	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
3. System jest łatwy w użyciu.	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
4. Potrzebowałbym wsparcia technicznego, aby korzystać z tego systemu.	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
5. Poszczególne funkcje systemu są łatwo dostępne.	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
6. System nie jest zbyt logiczny (spójny).	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
7. Większość osób będzie w stanie opanować system bardzo szybko.	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
8. System jest kłopotliwy w użyciu.	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
9. Czuję się bardzo pewnie korzystając z systemu.	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
10. Musiałem opanować wiele rzeczy przed rozpoczęciem pracy z systemem.	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5

Rys. 3.2. Pytania ankiety SUS wykorzystane w badaniu oraz wybór odpowiedzi w skali Likerta

Po odpowiedzeniu na wszystkie pytania z ankiety SUS badany miał 90 sekund na odpoczynek. W tym czasie na ekranie pojawiał się schematyczny rysunek zamkniętych oczu, który stopniowo stawał się coraz ciemniejszy i na którym widoczna była także liczba sekund do końca odpoczynku (w sekundach), która też łagodnie znikła. Celem było zasugerowanie badanemu, że warto zamknąć oczy w celu odpoczynku.

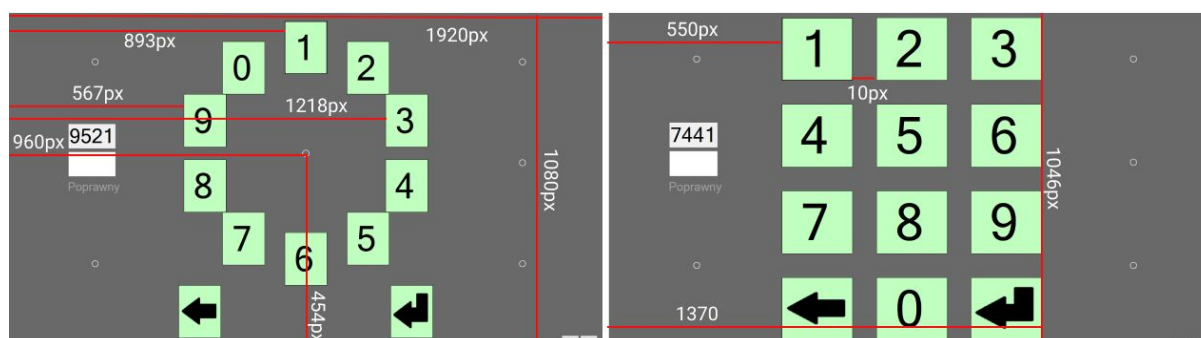
Po przerwie badany przechodził do kolejnej sekwencji 14 prób ze zmienionym warunkiem eksperymentu (zmieniona metoda i/lub ułożenie klawiszy). Po zakończeniu szóstej sekwencji następowała końcowa ankieta SUS, a potem zakończenie badania.

3.2 Opis działania metod aktywacji przycisków i układów klawiszy

Układy klawiszy

Ekran monitora, na którym był przeprowadzony eksperyment miał wymiary 1920×1080. Wszystkie przyciski w obu układach znajdują się na panelu o stałej wielkości zależnej od wielkości okna, który jednak nie był wizualnie wyróżniony z pozostałej części okna. W trakcie eksperymentu okno było powiększone do wielkości ekranu. Górna krawędź panelu znajdowała się o 20 pikseli poniżej górnej krawędzi okna. Szerokość panelu jest automatycznie dopasowywana tak, żeby stosunek szerokości do wysokości pozostawał równy około 0.73. Taki sam margines dzielił dolną część panelu od dolnej krawędzi okna. W ustawieniu *Telephon Pad* dwanaście przycisków były rozłożone równomiernie na całym panelu z marginesem od brzegu i odległościami między przyciskami równym 10 pikseli. To oznacza, że lewy górny róg przycisku z etykietą "1" znajdował się 550 pikseli od lewej krawędzi ekranu i 30 pikseli od górnej krawędzi. Natomiast prawy dolny róg przycisku akceptacji znajduje się 1370 pikseli od lewej krawędzi ekranu i 1046 od górnej. W ustawieniu *Circle* dziesięć przycisków z cyframi było ułożone na okręgu o środku w punkcie (960, 454) o promieniu takim, żeby między najbardziej wysuniętymi przyciskami, a lewą, górną i prawą krawędzią panelu pozostawał margines równy 10 pikseli. W efekcie przycisk z etykietą "1" był odsunięty o 30 pikseli od górnej krawędzi ekranu i jego lewa krawędź była odległa o 893 pikseli od lewej krawędzi. Lewa krawędź przycisków 3 i 4 umieszczona była 1218 pikseli od lewej krawędzi ekranu, a przycisków 8 i 9 o 567 pikseli. Natomiast przycisk usuwania cyfry znajdował się w lewym dolnym rogu 10 pikseli od dolnej i lewej krawędzi panelu, a przycisk akceptacji (nie używany w eksperymencie) znajdował się w prawym dolnym rogu (położenie (1234, 881)). Przyciski miały kolor jasnozielony RGB (192; 255; 192), natomiast ich etykiety były czarne. Tło okna miało kolor ciemno szary RGB (105, 105, 105), co zapobiegało nadmiernemu zmęczeniu oka. Jeżeli

spojrzenie skierowane było na przycisk, jego kolor stawał się nieco ciemniejszy RGB (132; 255; 132), a w momencie aktywacji przycisku (wzrokowego “kliknięcia”) etykieta stawała się ciemnozielona RGB (32; 135; 32). Dodatkowo w przypadku metody opartej na geście “wejdź i wyjdź” na ekranie było umieszczonych osiem dodatkowych kółek, które miały ułatwić zadanie kliknięcia przycisku za pomocą gestu, stanowiąc punkty, do których można było kierować wzrok (przewodnik dla wzroku). Kółka te miały średnicę równą 20 pikseli (**rys 3.4**).



Rys. 3.3. Wymiary układów klawiatury Circle i Telephon Pad.

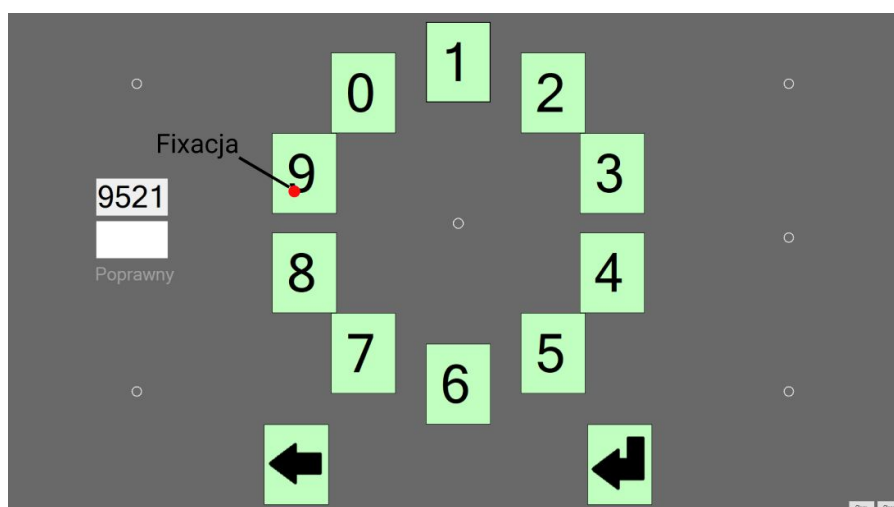
Metoda Dwell Time

Metoda *Dwell Time* oznacza, że użytkownik musi umieścić spojrzenie w obszarze przycisku i pozostawić je tam na dłuższy od założonego czasu aktywacji. W trakcie eksperymentu użyliśmy czasu aktywacji równego 1500 ms (przez pierwsze 7 prób sekwencji), a następnie przez 500 ms. Dłuższy czas aktywacji zwiększa precyzję metody, zapobiegając przypadkowym “kliknięciom”, ale ogranicza prędkość wprowadzania liczb. To typowy przykład opisanego w poprzednim rozdziale kompromisu między precyzją, a szybkością (ang. *speed-accuracy trade off*).

W tej metodzie użyliśmy podwójnej informacji zwrotnej. Przycisk jest podświetlany w sposób ciągły, jeżeli wzrok jest na niego skierowany, a dodatkowo zmienia na chwilę kolor podczas kliknięcia.

Metoda Fixation

Metoda wykorzystująca wykrywanie fiksacji jest podobna do metody z czasem aktywacji, ale warunkiem aktywacji przycisku nie jest czas utrzymywania spojrzenia w obszarze przycisku, ale wykrycie fiksacji. To oznacza, że w trakcie działania programu musi być przeprowadzana analiza danych okoruchowych i wykrywane są fiksacje oraz sakady. Dostępne są dwa algorytmy: prędkościowy i położeniowy. Informacja zwrotna dostarczana użytkownikowi była taka sama, jak w przypadku metody z czasem aktywacji.



Rys. 3.4. Jedna fiksacja, która powoduje kliknięcie przycisku.

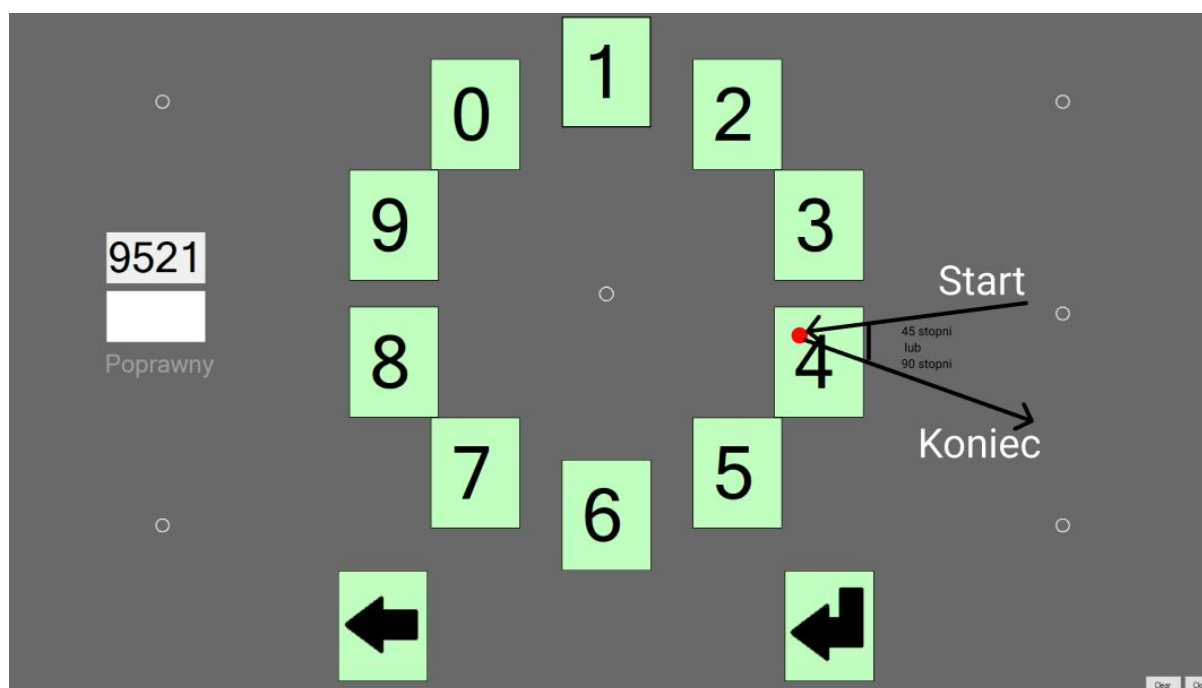
Metoda Enter and Leave

W metodzie *Enter and Leave* użytkownik musi wykonać dwa ruchy okiem, zmieniając położenie spojrzenia na ekranie:

- 1) gest rozpoczyna się z dowolnego miejsca na ekranie, najlepiej niedaleko przycisku, który użytkownik chce “kliknąć”,
- 2) następnie użytkownik najeżdża wzrokiem na przycisk.

- 3) aby zakończyć gest wymagany jest powrót spojrzenia z obszaru przycisku do miejsca w pobliżu rozpoczęcia gestu. To powoduje aktywację przycisku (wzrokowe kliknięcie).

Między kierunkiem ruchu wejścia i wyjścia z obszaru przycisku mierzony jest kąt (por. rys. 15). Można ustawić maksymalną wartość tego kąta, dla którego ruch oka uznawany jest za gest wzrokowy. W trakcie eksperymentu ten kąt progowy wynosi 45 stopni albo 90 stopni (w zależności od fazy sekwencji prób). Dodatkowo wzrok musi znajdować się w obszarze przycisku przez czas od 500 do 1000 ms. Gest wymaga więc chwilowego zatrzymania wzroku na przycisku (akcent), który zapobiega uznaniu za gest swobodnej eksploracji zawartości ekranu.



Rys. 3.5. Pierwsza (górna) strzałka oznacza pierwszy ruch gestu, a druga strzałka (dolna) drugi.

W trakcie wykonywania gestu, jedyną informacją zwrotną była zmiana koloru tła przycisku w momencie najechania na niego wzrokiem i potem zmiana koloru etykiety po “kliknięciu”.

Rozdział 4

Analiza i wyniki

Analiza została przeprowadzona za pomocą języka R w środowisku R Studio (J. J. Allaire, 2009). Do analiz statystycznych wykorzystaliśmy następujące funkcje:

- `pairwise.wilcox.test` - wykonuje test nieparametryczny sumy rang Wilcoxa parami,
- `kruskal.test` - wykonuje test nieparametryczny Kruskala-Wallisa,
- `shapiro.test` - służy do sprawdzenia normalności rozkładu,
- `lm` - oblicza regresję liniową.

W każdej hipotezie z wyjątkiem hipotezy 7 usunęliśmy pomiary odstające i sprawdziliśmy normalność rozkładu za pomocą testu Shapiro-Wilka (SW). Wyniki testu SW umieściliśmy w **tab. 4.1**. Po wykonaniu tych czynności użyliśmy metod statystycznych Kruskala-Wallisa (KW) oraz Wilcoxa (wykonywany parami, pW), ponieważ sam test nieparametryczny KW nie daje informacji o tym, które pary grup są różne od siebie oraz dlatego, że wszystkie zmienne użyte w analizach nie miały rozkładu normalnego. W przypadku hipotezy 7 usunęliśmy tylko pomiary odstające.

hipoteza	wielkość/zmienna	wynik SW (N)	wynik SW (p)
Hipoteza 1	średni czas wpisywania cyfry (w ms)	1991	$p < 2.2e-16$
Hipoteza 2	średni czas wpisywania cyfry (w ms)	1991	$p < 2.2e-16$
Hipoteza 3	liczba usunięć znaków	2012	$p < 2.2e-16$
Hipoteza 4	liczba sakad	1993	$p < 2.2e-16$

Hipoteza 5	użyteczność	173	$p = 2.55e-10$
Hipoteza 6	średni czas fiksacji	1949	$p < 2.2e-16$
Hipoteza 8	rozkład entropii	1943	$p = 2.465e-11$

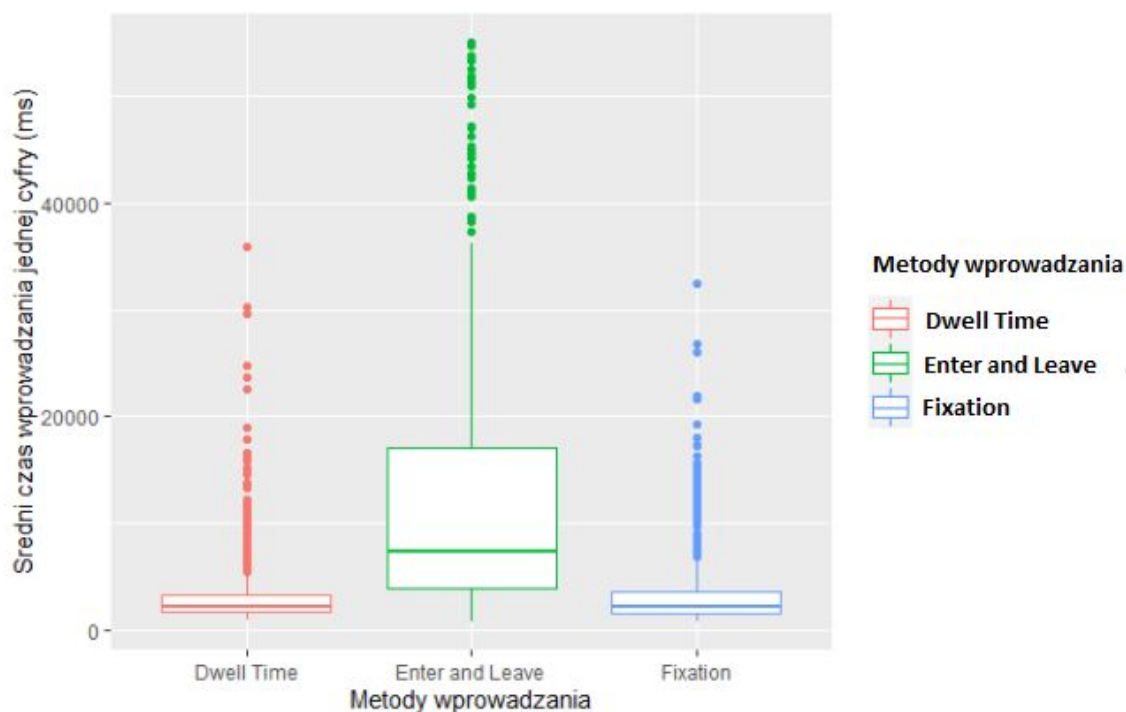
Tab. 4.1. Wyniki testu Shapiro-Wilka na normalność rozkładu.

Hipoteza 1

Różnice między metodą *Enter and Leave*, a metodami *Dwell Time* i *Fixation* są istotne ($pW, p < 2e-16$). Metoda *Enter and Leave* okazuje się metodą, w przypadku której wprowadzenie pojedynczego przycisku trwało najdłużej (patrz **tab 4.1** i **wyk 4.1**), co oznacza, że hipoteza 1 została obalona: metoda *Enter and Leave* nie skraca czasu wprowadzenia przycisku.

metoda	n	średnia	sd	mediana
<i>Dwell Time</i>	798	3167	3360	2155
<i>Enter and Leave</i>	418	12730	12640	7383
<i>Fixation</i>	775	3251	3305	2187

Tab. 4.1. Wyniki summaryczne czasu wprowadzanie każdej z metod w dwóch układach klawiatury. Najdłuższy czas osiągnęła metoda *Enter and Leave*.



Wyk. 4.1. Średni czas wprowadzania cyfry między trzema badanymi metodami w dwóch układach klawiatury.

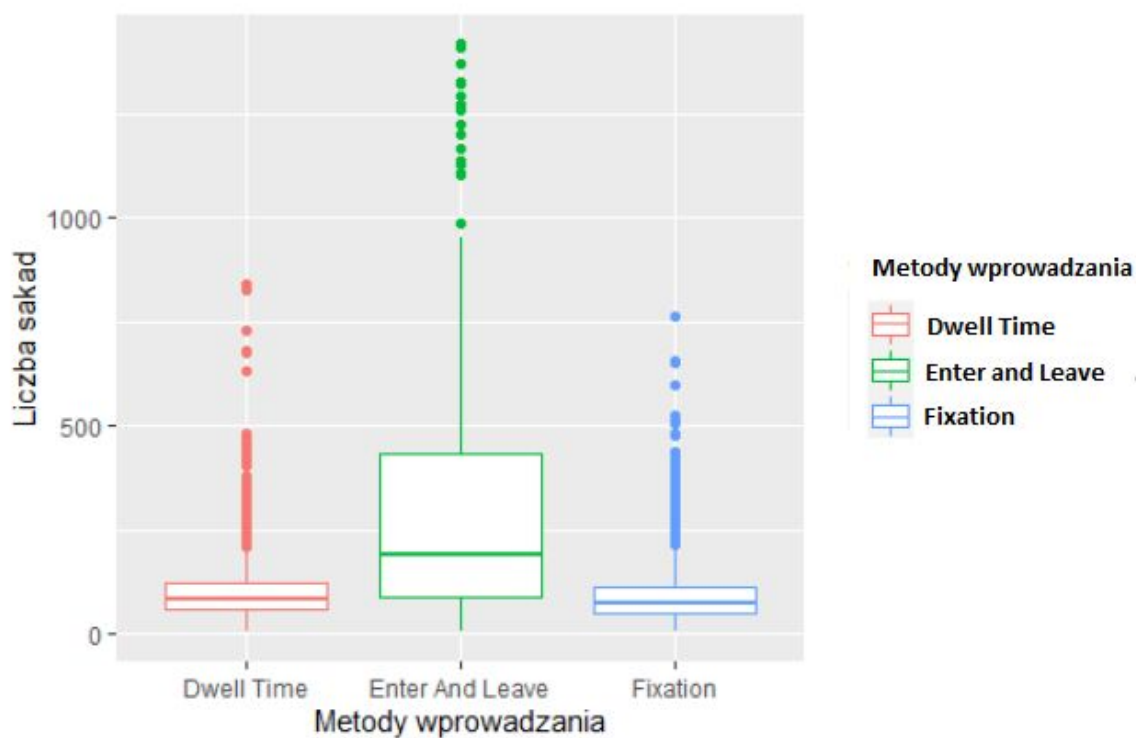
Hipoteza 2

Nie ma istotnej różnicy w średnim czasie wprowadzania cyfry między metodami *Fixation* i *Dwell Time* (pW, $p = 0.45$), więc hipoteza 2 została obalona.

Hipoteza 3

Metody wprowadzania różnicują liczby usuwanych cyfr (poprawianych błędów) (KW, $p < 2.2e-16$). W przypadku układu klawiatury *Circle*, dalej istnieje istotna różnica w liczbie usuwanych cyfr lecz zmniejsza się ona między metodami *Enter and Leave*, a *Fixation* (pW, $p = 0.00052$). Najmniej błędów stwierdzono w przypadku metody *Dwell Time*, kolejną jest metoda *Fixation*, a najwięcej błędów zostało popełnionych w przypadku metody *Enter and Leave* (zob. **tab. 4.2** i **wyk. 4.2**).

To obala naszą hipotezę 3 - metody *Fixation* i *Dwell Time* są istotnie różne od siebie pod względem liczby błędów.



Wyk. 4.2. Ilość usunięć znaków w trzech metodach wprowadzania przycisków w dwóch układach klawiatury.

metoda	n	średnia	sd	mediana
<i>Dwell Time</i>	798	0.2731830	0.8188848	0
<i>Enter and Leave</i>	419	2.3269690	2.8929072	1
<i>Fixation</i>	795	0.9069182	1.3904641	0

Tab. 4.2. Wyniki sumaryczne ilości usunięć znaków każdej z metod w dwóch układach klawiatury.

Hipoteza 4

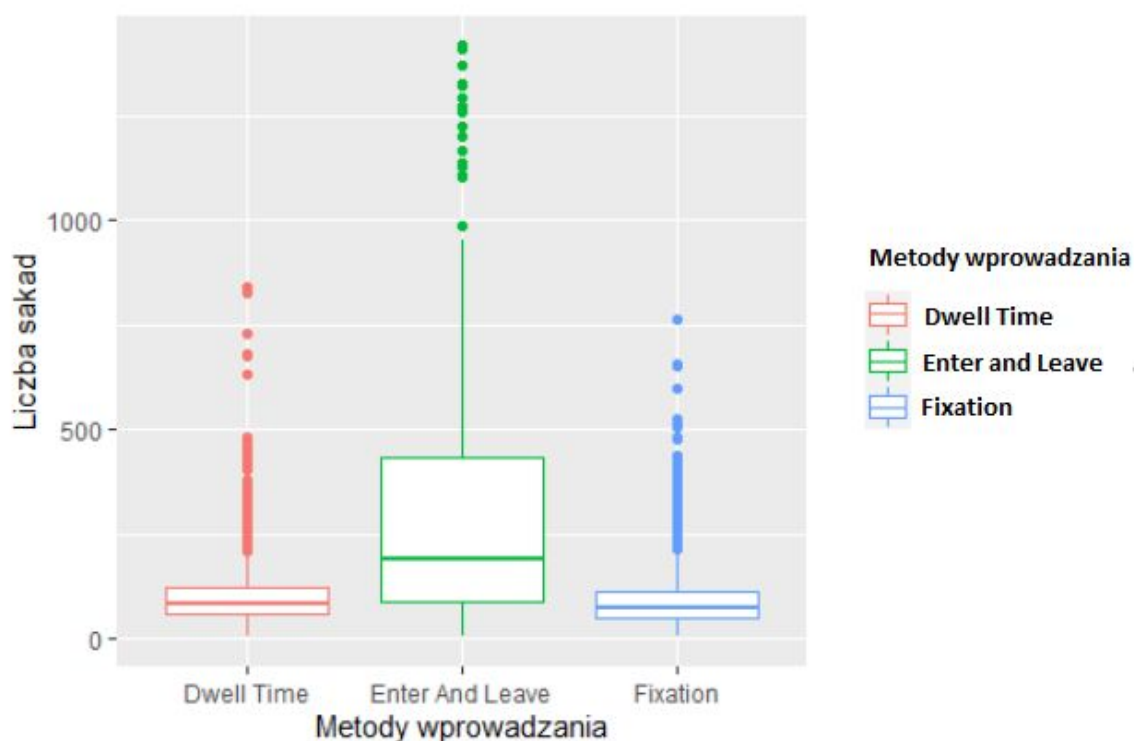
Różnice między metodami są istotne statystycznie (KW, $p < 2.2e-16$). Średnia liczba sakad w trakcie wprowadza liczby w metodzie *Enter and Leave* jest największa i wynosi w przybliżeniu 313 (**tab 4.3**). Daje to istotną różnicę

między resztą metod (pW, $p < 2e-16$). Między metodami *Fixation* i *Dwell Time* nie ma istotnej różnicy (pW, $p = 0.00033$).

Hipoteza 4 nie została potwierdzona. Istotnie najwięcej sakad występuje w metodzie *Enter and Leave*, ale między metodami *Dwell Time* i *Fixation* również zachodzi znacząca różnica.

metoda	n	średnia	sd	mediana
<i>Dwell Time</i>	780	108	97	86
<i>Enter and Leave</i>	418	313	312	191.5
<i>Fixation</i>	795	101	95	75

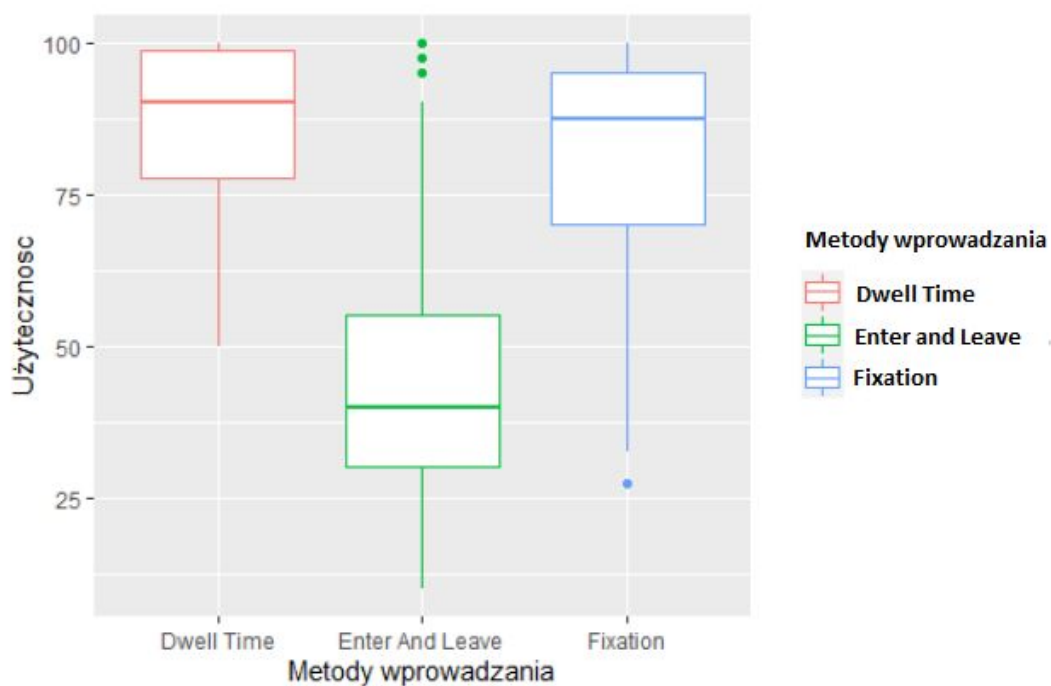
Tab. 4.3. Wyniki sumaryczne ilości sakad każdej z metod w dwóch układach klawiatury.



Wyk. 4.3. Liczba sakad między trzema badanymi metodami w dwóch układach klawiatury.

Hipoteza 5

W ankiecie użyteczności SUS, metoda *Enter and Leave* wypadła najgorzej ze wszystkich metod uzyskując ocenę prawie dwa razy niższą niż dwie pozostałe. Różnica między tymi ocenami jest istotna statystycznie (KW, $p = 1.712e-14$), co potwierdza hipotezę 5.



Wyk. 4.4. Użyteczność między trzema badanymi metodami w dwóch układach klawiatury.

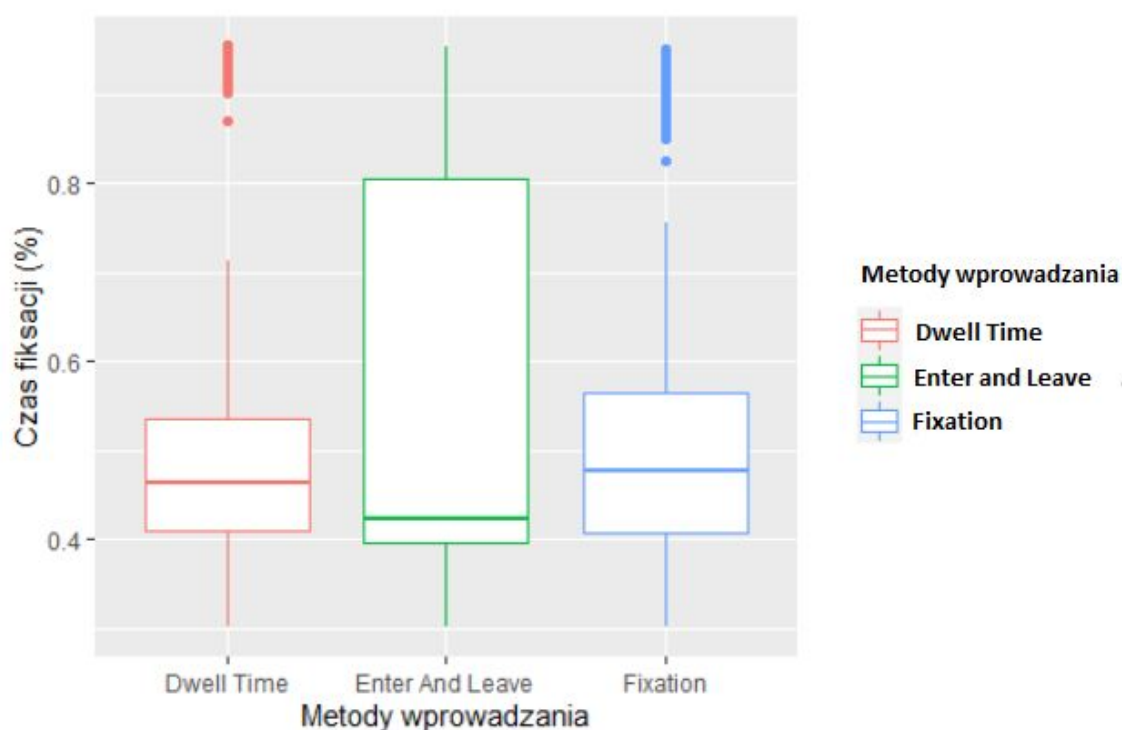
metoda	n	średnia	sd	mediana
<i>Dwell Time</i>	63	86	14	90
<i>Enter and Leave</i>	53	48	24	40
<i>Fixation</i>	57	82	19	87.5

Tab. 4.4. Wyniki sumaryczne użyteczności każdej z metod w dwóch układach klawiatury.

Hipoteza 6

Różnice czasu fiksacji ze względu na użytą metodę wprowadzania między metodami są istotne (KW, $p = 0.0005128$), jednak test pW nie potwierdził istotnej różnicy między metodami *Dwell Time* i *Fixation* (pW, $p = 0.1221$).

Średni czas fiksacji przy wprowadzaniu liczby metodą *Enter and Leave* jest najwyższy i znacząco różni się od metody *Dwell Time*. To oznacza, że hipoteza 6 została obalona.



Wyk. 4.5: Średni czas fiksacji między trzema badanymi metodami w dwóch układach klawiatury.

metoda	n	średnia	sd	medianA
<i>Dwell Time</i>	756	0.5094752	0.1610439	0.4646553
<i>Enter and Leave</i>	433	0.5395167	0.2201821	0.4243215

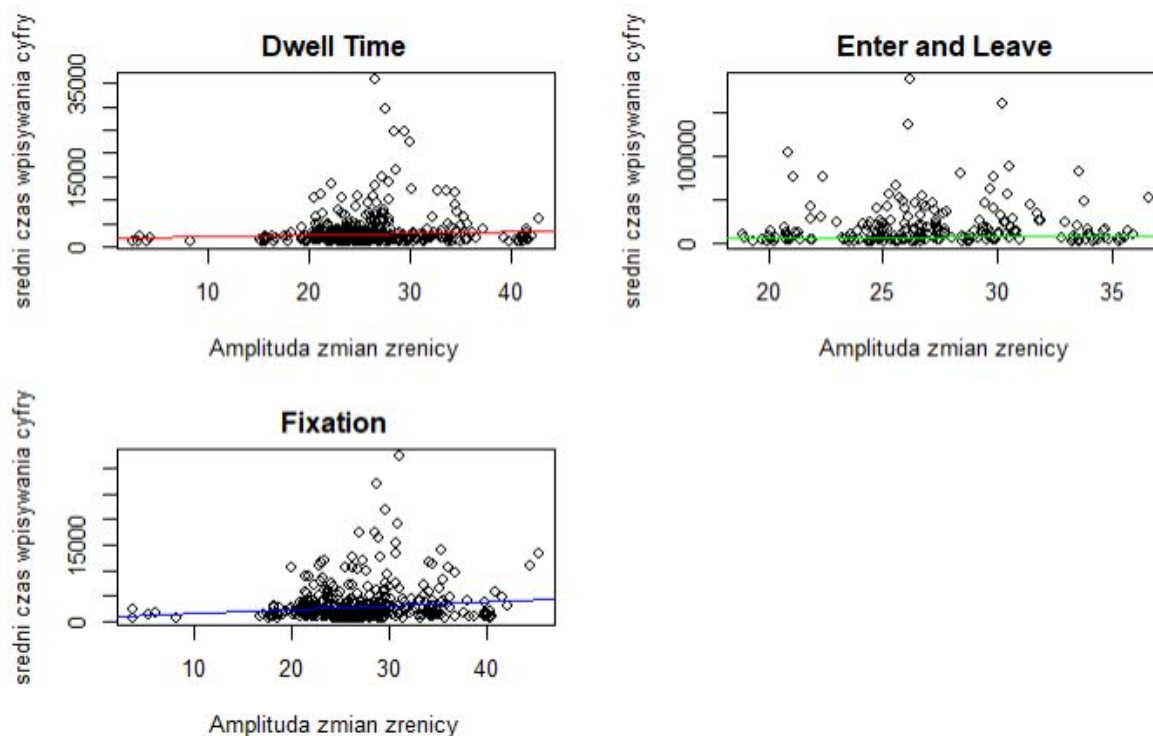
Fixation	760	0.5247286	0.1687348	0.4778359
----------	-----	-----------	-----------	-----------

Tab. 4.5. Wyniki summaryczne czasu fiksacji każdej z metod w dwóch układach klawiatury.

Hipoteza 7

Do dopasowania funkcji liniowej opisującej zależność średniego czasu wpisywania od średnicy źrenicy i amplitudy jej zmiany użyliśmy regresji liniowej. Okazuje się, że najmniejsza zmiana amplitudy w odniesieniu do średniego czasu wpisywania cyfry zachodzi w przypadku metody *Enter and Leave* ($t = 1.073$, $p = 0.285$) (**wyk 4.7**), następna jest w metodzie *Dwell Time* ($t = 2.502$, $p = 0.0126$), a największa istotna zmiana zachodzi w przypadku metody *Fixation* ($t = 2.793$, $p = 0.00539$). Natomiast średnia zmiana średnicy źrenicy w odniesieniu do średniego czasu wpisywania cyfry jest istotna w przypadku wszystkich metod (*Enter and Leave* $t = -2.871$, $p = 0.00465$; *Dwell Time* $t = -2.696$, $p = 0.00721$; *Fixation* $t = -3.194$, $p = 0.00148$).

Hipoteza 7 została więc obalona. Największa zmiana średnicy źrenicy zachodzi w przypadku metody *Fixation*, a nie metody *Enter and Leave*.

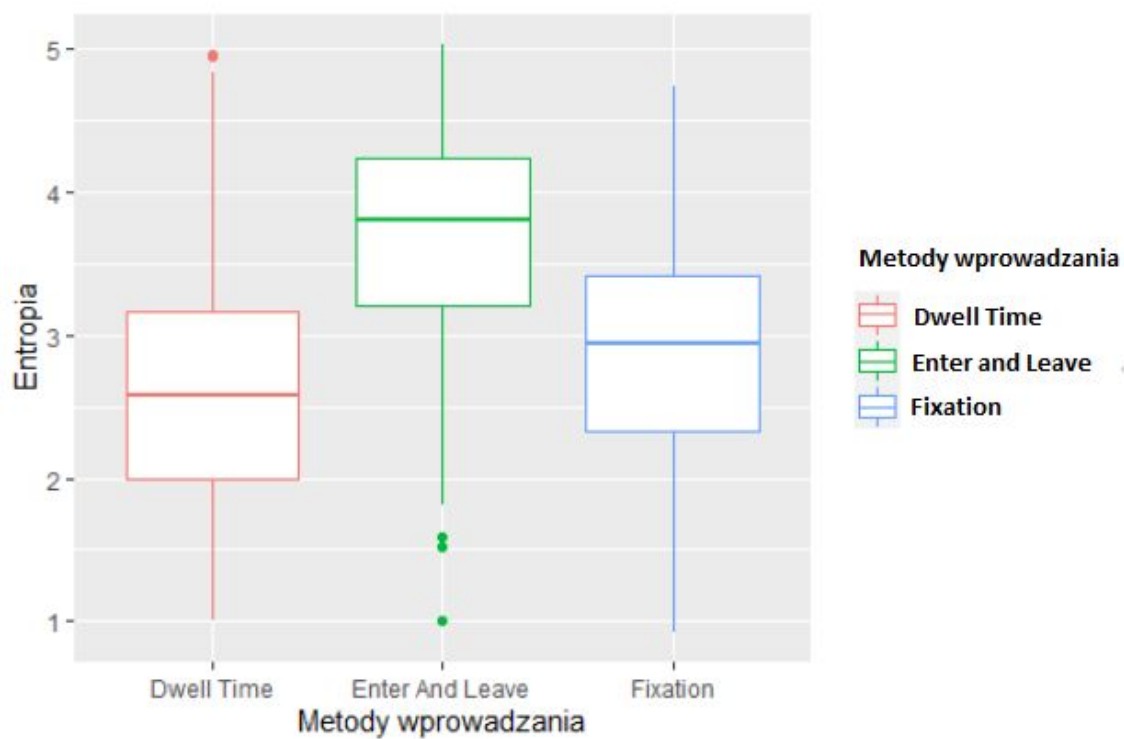


Wyk. 4.7. Linie regresji w zależności od metody wprowadzania.

Hipoteza 8

Różnice rozkładu entropii ze względu na metody wprowadzania są znaczące (KW, $p < 2.2e-16$). W układzie klawiatury *Telephon Pad* i między metodami *Dwell Time* i *Fixation* wartość p jest jednak blisko progu (pW, $p = 0.0039$).

Hipoteza 8 zostało zatem obalona. Rozkład entropii w metodzie *Enter and Leave* jest istotnie większy od rozkładu w przypadku innych metod, ale między metodami *Dwell Time* i *Fixation* również zachodzi istotna różnica.



Wyk. 4.8. Entropia między trzema badanymi metodami w dwóch układach klawiatury.

metoda	n	średnia	sd	mediana
<i>Dwell Time</i>	766	2.641458	0.8018018	2.584963
<i>Enter and Leave</i>	411	3.678843	0.7791516	3.807355
<i>Fixation</i>	766	2.894174	0.7785095	2.947703

Tab. 4.8. Wyniki summaryczne rozkładu entropii dla każdej z metod w dwóch układach klawiatury.

Rozdział 5

Dyskusja

W badaniu porównaliśmy trzy metody wprowadzania wzroku: *Fixation*, *Enter and Leave* oraz *Dwell Time*. We wcześniejszej literaturze nie znalazłem informacji na temat metody *Enter and Leave*, chociaż metoda użyta w badaniach Hyrskykari i inni (2012) oraz Istance i inni (2010) bardzo ją przypomina. Pomimo, że najbardziej innowacyjna metoda *Enter and Leave* nie okazała się najlepsza pod względem użyteczności, zarówno mierzonej obiektywnie, jak i opiniami badanych, to jesteśmy zadowoleni z uzyskanych wyników, które pomogły nam zidentyfikować problemy, na których powinniśmy się skupić, aby ją poprawić (zob. rozdział 6.).

Użyteczność metody *Enter and Leave*

Analiza statystyczna zebranych danych wskazuje, że problemy ze wzrokowym wprowadzaniem tekstu metodą *Enter and Leave* dotyczą przede wszystkim błędów wprowadzania (na co wskazuje potwierdzenie **hipotezy 3**). Pomyłki przy wpisywaniu cyfr podwójnie zwiększają czas wpisywania, bo nie tylko samo błędne wprowadzenie cyfry zajmuje czas, ale i jego poprawienie. Ponadto znalezienie wzrokowego przycisku służącego do usuwania cyfr przerywa skupienie przy ich wprowadzaniu. Moim zdaniem to było główną przyczyną niezadowolenia z metody *Enter and Leave*. Błędy powodowały konieczność wykonania wielu dodatkowych ruchów oczu, co potwierdza większa liczba sakad (potwierdzona **hipoteza 4**). Dodatkowo można zauważyć zwiększoną entropię (**hipoteza 8**), co świadczy o tym, że ruchy oczu były bardziej „rozrzucone” w przypadku metody *Enter and Leave* niż w przypadku innych metod. To może wskazywać na zagubienie użytkownika, który musi eksplorować zawartość ekranu, aby ustalić prawidłowy sposób wprowadzania cyfr, ale może być też spowodowane samą mechaniką działania tej metody, w której gesty wymagają dodatkowych ruchów oka, które mogą być interpretowane jako

sakady. Te czynniki, a więc błędy wpisywania oraz duża liczba ruchów sakkadowych, są prawdopodobnie przyczynami mniejszej użyteczności metody *Enter and Leave* uzyskanej w ankiecie SUS (**hipoteza 5**).

Porównanie metod Dwell Time i Fixation

Metoda *Fixation* wypadła prawie tak dobrze, jak *Dwell Time* pod względem czasu wprowadzania przycisku (**hipoteza 1**), przy czym metoda *Fixation* generuje istotnie mniej sakad (**hipoteza 4**), lecz to nie pomogło zmniejszyć liczby błędnych wprowadzeń (**hipoteza 3**).

Uwagi na temat badania

Dodatkowo po badaniu, uczestnicy dzielili się ze mną swoją subiektywną opinią dotyczącą wrażeń z badania. Dotyczyły one między innymi problemu z “klikalnością” przycisku usuwania, częstym omyłkowym “klikaniem” przycisków sąsiadujących oraz braku *feedbacku* dotyczącego aktualnego stanu programu w trakcie wykonywania gestu w metodzie *Enter and Leave*.

Minusami przeprowadzonego eksperymentu były także błędy działania programu oraz chwilowe braki łączności między eyetrackerem a laptopem podczas eksperymentu. Próby w których taka sytuacja miała miejsce nie zostały uwzględnione w analizach.

Naszym priorytetem w przyszłych badaniach będzie przede wszystkim naprawa wykrytych błędów i rozwiązanie zgłoszonych problemów tak, żeby wszystkie trzy metody wprowadzania cyfr, a w szczególności metoda *Enter and Leave*, były bardziej przyjazne i intuicyjne (zob. rozdział 6).

Rozdział 6

Pomysły na rozwój oprogramowania

Interakcja wzrokowa, w szczególności metody wzrokowego wprowadzania tekstu (i liczb) to nie tylko algorytmy wprowadzania. Z założenia mają służyć ludziom, wymagają wobec tego wieloetapowej optymalizacji. Tę pracę można uznać za jeden z takich etapów.

Poniżej zebrałem wnioski dotyczące badanych metod wprowadzania liczb oparte na wynikach przeprowadzonego przeze mnie eksperymentu. Uwzględniłem nie tylko potwierdzone i odrzucone hipotezy, ale również opinie badanych oraz moje własne obserwacje i przypuszczenia.

Problemy metody *Enter and Leave* i propozycje rozwiązań

Największym problemem metody *Enter and Leave* jest duża liczba błędnych “kliknięć” (hipoteza 3). Dodatkowo badani informowali mnie o tym, że metodzie *Enter and Leave* towarzyszy niewystarczająca informacja zwrotna. To powodowało dużą frustrację wśród uczestników badania.

Badani często mieli również problemy z “kliknięciem” przycisku w układzie klawiatury *Telephon Pad*, szczególnie jeżeli należało “kliknąć” przycisk znajdujący się w środkowej kolumnie i wierszu. Wzrok często “zahaczał” wówczas o inny przycisk, co wywoływało niezamierzone wprowadzenia. Mogło to być związane z niewielkimi odległościami między przyciskami. (por. tzw. *Steering law* Accota i Zhai (1997)). W przypadku układu klawiatury *Telephon Pad* takich ciasnych przejść między przyciskami, które należy ominąć, jest więcej niż w przypadku klawiatury typu *Circle*, w której można swobodnie poruszać wzrokiem po całym wnętrzu koła. Problem ten można rozwiązać zwiększając odległość między przyciskami, ale konsekwencją będzie zmniejszenie ich rozmiaru (taka opcja jest już zaimplementowana, ale nie była wykorzystana w

tym eksperymencie). Drugim rozwiązaniem mogłoby być dodanie informacji zwrotnej przy korzystaniu z metody *Enter and Leave*. Obecnie brakuje tego rodzaju informacji i użytkownik nie wie czy aktywuje gestem właściwy przycisk. Przykładowo można by było użyć kursora wzrokowego, który pokazywałby aktualne miejsce spojrzenia. To mogłoby uchronić przed niekontrolowanymi "kliknięciami", które w tej chwili mogą się zdarzyć przy swobodnej eksploracji ekranu.

Kolejną przyczyną, która wydaje się generować przypadkowe kliknięcia w metodzie *Enter and Leave* jest wyświetlona z lewej strony ekranu liczba, którą badany ma wprowadzić. Jest ona wprowadzanie także wypowiedziana przez syntezy, ale mimo to badani spoglądali na nią w trakcie wprowadzania, aby upewnić się, że nie popełnili błędu. W takim przypadku badany przenosił wzrok na liczbę, a następnie musiał ponownie przeskanować klawiaturę numeryczną, aby znaleźć odpowiedni przycisk wzrokowy. Podczas tego ruchu mogło następować przypadkowe kliknięcie. Rozwiązanie tego problemu można zaczerpnąć z omawianych już badań Hyrskykari i innych (2012) oraz Istance i innych (2010), w których wykorzystano tak skonfigurowane gesty, które wyraźnie różnią się od ruchów oczu podczas obserwacji ekranu (np. trójpunktowe).

Problem spoglądania na liczbę, jaką należy wprowadzić występuje także w przypadku metody *Dwell Time*, w której wymagane jest dłuższe skupienie na przycisku i w której takie szybkie spojrzenie w lewą stronę zaburzy proces akceptacji. W eksperymencie można by w czasie akceptacji umieścić liczbę na przycisku lub jako większe tło do częściowo przezroczystych przycisków, ale takie rozwiązanie nie ma zastosowania w sytuacji, w której użytkownik miałby wprowadzać dowolny pin z pamięci. Z kolei w układzie klawiatury *Circle* dobrym miejscem mogłoby być środek koła utworzonego z przycisków.

Problemy i rozwiązania metody Dwell Time

Jedyną zmianę jaką wprowadził bym w przypadku metody *Dwell Time* to zmiana *feedbacku*, którym się posługiwaliśmy w badaniu. Obecnie *feedback* sygnalizuje inicjalizację i akceptację. Można by do tego dodać stopniowe zmniejszenie się cyfry na aktywowanym przycisku w trakcie etapu postępu podobnie, jak w badaniu Majaranta i innych (2003).

Problemy i rozwiązania metody Fixation

Problemem metody *Fixation* były niekontrolowane ponowienia wprowadzenia cyfr. Do kliknięcia wystarczy pojedyncza fiksacja w obszarze przycisku, co z założenia ma skrócić czas wprowadzania. Jednak jeżeli wzrok pozostaje na danym przycisku po jego kliknięciu, to warunek akceptacji (jedna fiksacja) ponownie był spełniony i następowało ponowne “kliknięcie”. Rozwiązaniem tego problemu mogłoby być ustawienie czasu dodatkowe czasu akceptacji 50-150 ms dla ponawiania (czyli ponowne wprowadzanie tej samej cyfry wymagałoby użycia *de facto* metody *Dwell Time* z krótkim czasem aktywacji) lub dezaktywowanie przycisku do czasu wyjścia z niego wzroku (wówczas utrudnione byłoby ponowne wprowadzenie tej samej cyfry).

Bibliografia

- Accot, J., & Zhai, S. (1997). Beyond Fitts' law. In *CHI '97 extended abstracts on Human factors in computing systems looking to the future - CHI '97*.
<https://doi.org/10.1145/1120212.1120376>
- Allaire J.J (2009). Rstudio. <https://rstudio.com/>.
- Andrew T. Duchowski. Eye tracking methodology. Springer Nature.
- Bee, N., & André, E. (2008). Writing with Your Eye: A Dwell Time Free Writing System Adapted to the Nature of Human Eye Gaze. In *Lecture Notes in Computer Science* (pp. 111–122). https://doi.org/10.1007/978-3-540-69369-7_13
- Chi, C. F., & Lin, C. L. (1997). Speed and accuracy of eye-gaze pointing. *Perceptual and Motor Skills*, 85(2), 705–718.
- Drewes, H., & SchmiDwell Time, A. (2007). Interacting with the Computer Using Gaze Gestures. In *Lecture Notes in Computer Science* (pp. 475–488).
https://doi.org/10.1007/978-3-540-74800-7_43
- Figueiredo, P., & Fonseca, M. J. (2018). EyeLinks. In *Proceedings of the 2018 on International Conference on Multimodal Interaction - ICMI '18*.
<https://doi.org/10.1145/3242969.3243021>
- Heikkilä, H. (2013). Tools for a Gaze-Controlled Drawing Application – Comparing Gaze Gestures against Dwell Buttons. In *Human-Computer Interaction – INTERACT 2013* (pp. 187–201). https://doi.org/10.1007/978-3-642-40480-1_12
- Holmqvist, K. (2017). Eye Tracking: A comprehensive guide to methods and measures. Lund Eye-Tracking Research Institute.
- Huckauf, A., & Urbina, M. (2007). Gazing with pEYE. In *Proceedings of the 4th symposium on Applied perception in graphics and visualization - APGV '07*.

<https://doi.org/10.1145/1272582.1272618>

- Hyrskykari, A., Istance, H., & Vickers, S. (2012). Gaze gestures or dwell-based interaction? In *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '12*. <https://doi.org/10.1145/2168556.2168602>
- Istance, H., Hyrskykari, A., Immonen, L., Mansikkamaa, S., & Vickers, S. (2010). Designing gaze gestures for gaming. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications - ETRA '10*. <https://doi.org/10.1145/1743666.1743740>
- Kumar, C., Hedeshy, R., Scott MacKenzie, I., & Staab, S. (2020). TAGSwipe: Touch Assisted Gaze Swipe for Text Entry. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. <https://doi.org/10.1145/3313831.3376317>
- Majaranta, P., Ahola, U.-K., & Špakov, O. (2009). Fast gaze typing with an adjustable dwell time. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*. <https://doi.org/10.1145/1518701.1518758>
- Majaranta, P., Scott MacKenzie, I., Aula, A., & Rähkä, K.-J. (2003). Auditory and visual feedback during eye typing. In *CHI '03 extended abstracts on Human factors in computing systems - CHI '03*. <https://doi.org/10.1145/765891.765979>
- Majaranta, P., Scott MacKenzie, I., Aula, A., & Rähkä, K.-J. (2006). Effects of feedback and dwell time on eye typing speed and accuracy. In *Universal Access in the Information Society* (Vol. 5, Issue 2, pp. 199–208). <https://doi.org/10.1007/s10209-006-0034-z>
- Menges, R., Kumar, C., Sengupta, K., & Staab, S. (2016). eyeGUI. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction - NordiCHI '16*. <https://doi.org/10.1145/2971485.2996756>
- Laubheimer P.(2019).Accot-Zhai Steering Law: Implications for UI Design.
<https://www.nngroup.com/articles/steering-law/>
- Panwar, P., Sarcar, S., & Samanta, D. (2012). EyeBoard: A fast and accurate eye gaze-based text entry system. In *2012 4th International Conference on Intelligent*

- Human Computer Interaction (IHCI)*. <https://doi.org/10.1109/ihci.2012.6481793>
- Sarcar, S., Panwar, P., & Chakraborty, T. (2013). EyeK. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction - APCHI '13*.
<https://doi.org/10.1145/2525194.2525288>
- Treker. (2020). Midas. <https://en.wikipedia.org/wiki/Midas>
- Velichkovsky, B. B., Rumyantsev, M. A., & Morozov, M. A. (2014). New Solution to the Midas Touch Problem: Identification of Visual Commands Via Extraction of Focal Fixations. In *Procedia Computer Science* (Vol. 39, pp. 75–82).
<https://doi.org/10.1016/j.procs.2014.11.012>
- Velichkovsky, B., Sprenger, A., & Unema, P. (1997). Towards gaze-mediated interaction: Collecting solutions of the “Midas touch problem.” In *Human-Computer Interaction INTERACT '97* (pp. 509–516). https://doi.org/10.1007/978-0-387-35175-9_77
- Zhai, S., Hunter, M., & Smith, B. (2002). Performance Optimization of Virtual Keyboards. In *Human-Computer Interaction* (Vol. 17, Issue 2, pp. 229–269).
https://doi.org/10.1207/s15327051hci172&3_4
- Zhang, X., Xu, P., Zhang, Q., & Zha, H. (2011). Speed-accuracy trade-off in dwell-based eye pointing tasks at different cognitive levels. In *Proceedings of the 1st international workshop on Pervasive eye tracking & mobile eye-based interaction - PETMEI '11*.
<https://doi.org/10.1145/2029956.2029967>
- Jacob R. (2013) Eye tracking in advanced interface design. In: Barfield W, Furness TA editors. *Virtual Environments and Advanced Interface Design*. New York: Oxford University Press; 1995. p. 258 -288.
- D. Miniotas, O. Spakov, and G. Evreinov, “Symbol Creator: An Alter-native Eye-based Text Entry Technique with Low Demand for Screen Space,” in *Proceedings of INTERACT*, 2003, pp. 137–143.
- J. P. Hansen, D. W. Hansen, and A. S. Johansen, (2001) “Bringing Gaze-based Interaction

back to Basics,” Proceedings of Universal Access in Human-Computer Interaction, pp. 325–328, 2001.

xlabs, (2014), <https://www.xlabsgaze.com/>.

Załączniki

Załącznik 1. Link do pliku z kodem źródłowym języka R do analizy danych.

<https://github.com/Snikerso/Licencjat>

Aby uzyskać dostęp do pełnych danych, należy wysłać mail na adres e-mail:

snikersdro@gmail.com lub jacekmatulewski@umk.pl.

Załącznik 2. Instrukcja dla badanego

INSTRUKCJA

Zostaną Panu/Pani wyświetlone na ekranie różne rodzaje klawiatury numerycznej. Będą się one różniły sposobem ułożenia przycisków na ekranie. Pani/Pana zdaniem będzie wpisać podany po **lewej stronie ekranu czterocyfrowy PIN**, wykorzystując następujące metody wprowadzania cyfr:

- 1) Pierwsza z nich polega na **odpowiednio długim** skupieniu wzroku na danym przycisku, po którym następuje podświetlenie przycisku i wpisanie cyfry.
- 2) Druga polega na wejściu i wyjściu wzrokiem z danego przycisku:
 - najpierw skupiamy wzrok na jednym z kółek umieszczonym na ekranie
 - następnie przechodzimy wzrokiem na przycisk
 - i wracamy na to samo kółko, z którego zapoczątkował/a Pan/Pani ruch.

O tym, którą metodę należy użyć przy wpisywaniu cyfr zostanie Pan/Pani poinformowany przez osobę prowadzącą badanie.

Trudność zadania będzie się zmieniać.

W przypadku błędnego wpisania cyfry można ją **skasować korzystając z zamieszczonego w lewym dolnym rogu klawiatury, przycisku kasowania** (z przycisku korzystamy używając odpowiednio wymienionych powyżej metod).

Przed badaniem właściwym obie metody zostaną przećwiczone. W czasie próby można zadać pytania prowadzącemu celem wyjaśnienia wątpliwości dotyczących danej metody.

Załącznik 3. Formularz świadomej zgody na uczestnictwo w badaniu**Formularz świadomej zgody na udział w badaniu eksperymentalnym****Tytuł badania:** Badanie klawiatur numerycznych do interakcji wzrokowej

Ja niżej podpisany(a).....
 oświadczam, że przeczytałem/am i zrozumiałem/am wszystkie informacje dotyczące badania, w którym zamierzam wziąć udział oraz otrzymałem/am wyczerpujące, satysfakcjonujące mnie odpowiedzi na zadane pytania. Wyrażam dobrowolnie zgodę na udział w tym projekcie badawczym i jestem świadomy/świadoma faktu, iż w każdej chwili mogę się z nich wycofać bez podania przyczyny. Przez podpisanie zgody na udział w badaniu nie zrzekam się żadnych należnych mi praw. Otrzymam kopię niniejszego formularza opatrzoną podpisem i datą. Miałem/miałam dostatecznie dużo czasu, aby rozważyć mój udział w projekcie badawczym i podjąć decyzję.

Wyrażam również zgodę na przetwarzanie danych w tym badaniu zgodnie z obowiązującym w Polsce prawem (Ustawa o Ochronie Danych Osobowych z 29.08.1997). Zgadzam się na przekazanie wyników moich badań w formie uniemożliwiającej moją identyfikację do innych krajów, jedynie celem ich naukowej analizy. Dane analizowane przez odnośne władze, pracownie, w których wykonane zostaną eksperymenty i Komisję Etyczną, dostępne będą jedynie w postaci anonimowej. Zostałem/zostałam poinformowany/poinformowana, iż w przypadku wycofania zgody na udział w badaniu zgromadzone do chwili rezygnacji dane mogą zostać wykorzystane i przetwarzane jako część bazy danych tego projektu.

Zostałem/zostałam poinformowany/poinformowana, że administratorem danych będzie placówka, w której prowadzone jest badanie oraz o tym, że mam prawo wglądu do moich danych i ich poprawiania oraz o tym, że są one podawane dobrowolnie. Przez podpisanie tego dokumentu potwierdzam również, że zostałem/zostałam poinformowany/poinformowana o sposobie przetwarzania danych z badania i że dane te będą weryfikowane przez ich porównanie z moją dokumentacją oraz, że dane te są zbierane jedynie w celu naukowej analizy wyników.

Osoba badana:.....
Imię i nazwisko (drukowanymi literami)

PESEL

data badania

Podpis osoby badanej

Oświadczam, że omówiłem/omówiłam projekt badawczy z osobą badaną używając zrozumiałych, możliwie zrozumiałych sformułowań oraz udzieliłem/udzieliłam informacji dotyczących natury i znaczenia badania.

Osoba uzyskująca zgodę na badanie:.....
Imię i nazwisko (drukowanymi literami)

Podpis

data złożenia podpisu

Załącznik 4 Informacja dla osoby badanej

INFORMACJA DLA OSOBY BADANEJ

Projekt badawczy

Klawiatury numeryczne – interakcja wzrokowa

Proponujemy Pani/Panu udział w badaniu naukowym. Przed podjęciem decyzji o wzięciu udziału w badaniu prosimy o zapoznanie się z poniższą informacją.

Cel badania

Celem projektu jest zbadanie za pomocą eyetrackera (urządzenia rejestrującego ruch oczu) różnych wersji klawiatur numerycznych sterowanych wzrokowo.

Dobrowolność

Udział Pani/Pana w badaniach jest całkowicie **dobrowolny**, a odmowa nie pociągnie za sobą żadnych konsekwencji. Można wycofać się z badania na każdym jego etapie bez konieczności podawania przyczyn. Udział w badaniach jest również całkowicie **bezpłatny**.

Bezpieczeństwo

Nie stwierdzono żadnych niepożądanych czynników mogących stanowić zagrożenie dla zdrowia w przypadku takiego badania. Projekt zyskał akceptację Komisji ds. Etyki Badań Naukowych przy UMK.

Poufność informacji

Tożsamość Pani/Pana pozostanie poufna i nie pojawi się w żadnej publikacji naukowej. Dane osobowe zebrane podczas tego badania będą traktowane zgodnie z obowiązującą na terenie Polski ustawą o Ochronie Danych Osobowych z dn. 29 sierpnia 1997 roku (Dz. U. Nr. 133 poz 883 z podpunktami).

Podpisując formularz zgody na udział w badaniu, zezwala Pani/Pan przedstawicielom władz właściwym do spraw zdrowia, przedstawicielom Komisji Etycznej i niezależnym audytorom na dokonanie inspekcji Pani/Pana dokumentacji w celu weryfikacji poprawności wykonania badania.

Całość uzyskanych w trakcie badania wyników wykorzystana może być tylko do celów naukowych. Uczestnicy badania mają pełne prawo dostępu do swoich wyników.

Co mam teraz zrobić?

Jeśli Pani/Pan zdecyduje się na udział w badaniu, zostanie Pani/Pan poproszona/poproszony o podpisanie **Formularza Świadomej Zgody** na udział w badaniu.

Problemy i wątpliwości

Jeżeli ma Pani/Pan jakiegokolwiek pytania lub wątpliwości na każdym etapie badania może Pani/Pan zadawać pytania prowadzącym badanie lub skontaktować się z opiekunem części okulograficznej projektu Bibianą Bałaj (adres e-mail: bibianna.balaj@gmail.com).