

# Analysis of modern human-computer interfaces

## Analiza współczesnych interfejsów człowiek-komputer

Michał Cioczek\*, Tomasz Czarnota, Tomasz Szymczyk

*Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland*

### Abstract

Until recently, computers were devices intended for research and military processes, and today they are an integral part of human life. Such a rapid development of technology and new possibilities means that the science that deals with human-computer relations (HCI) has a lot of unexplored or insufficiently studied areas. Based on the available HCI research methods, this article is to show that the currently highly popularized computer mouse, which is a tool that enables HCI, will not always be the best solution for every user. Comparing a computer mouse with other commercially available controllers and testing them in a specially prepared test environment by a selected research group will be the best way to reliably state that the use of a computer mouse may not always be the best choice.

*Keywords:* Human Computer Interaction; ergonomics; interface; controller

### Streszczenie

Jeszcze do niedawna komputery były urządzeniami przeznaczonymi dla procesów badawczych i militarnych, a dziś stanowią integralną część życia człowieka. Tak szybki rozwój technologii i nowych możliwości sprawia, że nauka, która zajmuje się relacjami człowieka z komputerem (HCI) ma pełno niezbadanych lub niewystarczająco przebadanych obszarów. Niniejszy artykuł na podstawie dostępnych metod badawczych HCI ma pokazać, że obecnie mocno spopularyzowana mysz komputerowa, która jest narzędziem umożliwiającym HCI, nie zawsze będzie najlepszym rozwiązaniem dla każdego użytkownika. Zestawienie myszy komputerowej z innymi dostępnymi na rynku kontrolerami i przebadanie ich w specjalnie przygotowanym środowisku testowym przez wybraną grupę badawczą, będzie najlepszym sposobem do rzetelnego stwierdzenia, że użycie myszy komputerowa nie zawsze może okazać się najlepszym wyborem.

*Słowa kluczowe:* Interakcja Człowiek-Komputer; ergonomia; interfejs; kontroler

\*Corresponding author

E-mail address: [michal.cioczek@pollub.edu.pl](mailto:michal.cioczek@pollub.edu.pl) (M. Cioczek)

©Published under Creative Common License (CC BY-SA v4.0)

## 1. Wstęp

Współcześnie komputery oraz smartfony stały się nie tylko narzędziami pomocnymi człowiekowi, ale wręcz koniecznymi do funkcjonowania w życiu codziennym. Aspektem nauki, który stara się nie tylko zbadać relację między człowiekiem, a komputerem, ale i pomóc odkryć lub ulepszyć dostępne technologie umożliwiające taką zależność jest tzw. interakcja człowiek-komputer (ang. *HCI*, czyli *human-computer interaction*). Początki HCI sięgają połowy XX wieku, kiedy komputery zaczęły być wykorzystywane na szeroką skalę przez wojsko. Rozkwit tej nauki nastąpił natomiast w latach 80. XX wieku, kiedy komputery zaczęły pojawiać się w domach zwykłych ludzi [3].

Interakcja człowieka z komputerem jest niezwykle ciekawym i szerokim tematem, a że i niedostatecznie zbadanym, więc warto się w niego zagłębić. Szczególnie interesujące wydaje się wykorzystanie kontrolerów ruchowych umożliwiających wspomnianą interakcję. Na rynku są obecne różne kontrolery mające odmienną budowę i specyfikę. Wiele z nich dąży jednak do wspólnego celu, jakim jest przechwycenie obiektu, którym użytkownik będzie sterować i wykonywać nim akcje w przeznaczonym do tego celu środowisku (aplikacja). Ze względu, że najbardziej popularnym obiektem tego typu jest kursor ekranowy, jego przechwycenie

przez wybrane kontrolery ruchowe wydaje się być najlepszym wyborem do badania HCI.

Skąd pomysł, aby zbadać to zjawisko? Otóż obecnie ogrom ludzi korzysta z komputera i może się okazać, że szeroko spopularyzowana w ubiegłym wieku mysz komputerowa nie zawsze będzie najlepszym wyborem dla każdego użytkownika. W artykule opisano i wykorzystano metody badawcze HCI, które to sprawdzą [4].

Aby umożliwić przeprowadzenie takiego badania autorzy wybrali do porównania kilka z dostępnych na rynku kontrolerów będących urządzeniami wejścia, przygotowali środowisko testowe pozwalające na zebranie i przetworzenie wyników oraz wyznaczyli grupę badawczą.

Ponadto autorzy wybrali metody badawcze, które pozwolą zbadać interakcję człowieka z komputerem w sposób liczbowy, ale i metody pozwalające określić aspekt odczuć człowieka. Według badań w zakresie HCI okazuje się, że jedynie zestawienie wyników liczbowych wraz z wywiadem z grupą badawczą, pozwalają pokazać pełny obraz badanego zjawiska i umożliwić wyciągnięcie wniosków [4].

## 2. Metoda badań

Interakcja człowiek-komputer (HCI) jest dziedziną nauki, która posiada wiele metod badawczych pozwalających

jących na opisanie relacji użytkownika z systemem. Taka relacja została przedstawiona w modelu człowiek-praca-środowisko, który ma swoje podłoże w tzw. ergonomii, czyli dyscyplinie naukowej zajmującej się przystosowaniem narzędzi używanych przez użytkownika do jego anatomicznych i psychofizycznych potrzeb [3].



Rysunek 1: Model człowiek-praca-środowisko [9]

Do opisanego wspomnianej relacji jako metodę czy właściwie metody ogólne, należy podać metody jakościowe, na których analizie opierają się wszystkie metody związane z HCI. Do metod jakościowych należą:

- jakość konstrukcyjna (techniczna), która skupia się na budowie oraz wydajności interfejsu, czyli głównie na aspekcie wartości, które da się opisać liczbowo;
- jakość ergonomiczna, która skupia się na odczuciach użytkownika podczas korzystania z interfejsu (ergonomia);
- jakość użytkowa (użyteczność), która skupia się na satysfakcji użytkownika (czyli ukazuje zadowolenie użytkownika z interfejsu, którego używał) [3], [8].

### 2.1. Metoda obliczeniowa

Jako pierwszą z metod należy wskazać model KLM (ang. *Keystroke Level Model*), który wskazuje standardowe czasy trwania elementarnych operacji (Tabela 1). Model ten wywodzi się z techniki GOMS (ang. *Goals, Operators, Methods, Selection Rules*), która uwzględnia również zadania umysłowe, do jakich między innymi należy wybór sposobu wykonania operacji [8].

Metoda KLM zakłada, że użytkownik jest wprawiony, a więc umie posługiwać się danym interfejsem, zna swoje zadanie i kolejność kroków wykonywanej operacji. Dodatkowo są wyznaczone typowe czasy wykonania poszczególnych działań, dzięki czemu można oszacować ile będą trwały proste zadania oraz dłuższe sekwencje złożone z mniejszych kroków. W modelu KLM zakłada się, że użytkownik zna następujące sekwencje operacji:

- K - Press Key (kliknięcie przycisku klawiatury);
- P - Pointing (wskazanie przy pomocy kursora elementu docelowego);
- H - Hand (przeniesienie ręki pomiędzy myszą, a klawiaturą);
- M - Mental (mentalne przygotowanie się do wykonania operacji manualnej);
- D - Drawing (rysowanie drogi, którą pokonuje kursor);
- R - Response (odpowiedź systemu);
- B - Press Button (kliknięcie przycisku myszy) [9].

Tabela 1 Wybrane operacje użytkownika w modelu KLM [9]

AKCJA	OPIS	CZAS [s]
K	Naciśnięcie klawisza klawiatury	
	- wprawny operator	0,12
	- przeciętny operator	0,22
	- początkujący operator	0,28
	- niedoświadczony oper.	1,2
P	Wskazanie docelowego obiektu kursorem myszy	
	- na podstawie prawa Fittsa	$0,1 \log_2(0,5 + \frac{D}{S})$
	- przeciętnie	1,1
H	Przeniesienie dłoni do miejsca docelowego z/do myszy lub klawiatury	0,4
M	Przygotowanie się do wykonania operacji manualnej pod względem mentalnym	1,35
D	Rysowanie drogi, którą przebywa kursor	zależne od zadania
R	Odpowiedź systemu	zależne od zadania
B	Kliknięcie przycisku myszy	0,1

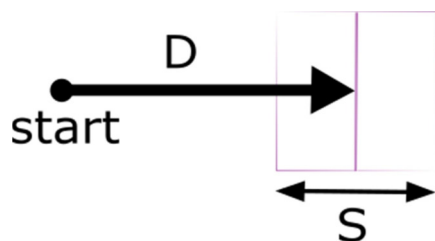
Na podstawie zaprezentowanych w tabeli wartości i danych, możliwe jest opracowanie scenariusza badawczego podzielonego na pojedyncze akcje. Tworzy się go sumując czasy poszczególnych czynności, budując tym samym łańcuchy operacji użytkownika. W rezultacie można oszacować średni czas realizacji całego zadania.

Do badania ruchu kursora systemowego przy pomocy kolejnych kontrolerów ruchowych najbardziej trafne jest wykorzystanie tzw. prawa Fittsa opisującego w sposób liczbowy motoryczne reakcje człowieka. Ten model pomija kwestie pomyłek i precyzji ruchu, a skupia się na wyznaczeniu optymalnego czasu dla wykonywanej operacji (zakłada się, że użytkownik jest wprawiony). Innymi słowy prawo Fittsa pozwala obliczyć czas konieczny do osiągnięcia obiektu graficznego (znajdującego się na ekranie monitora) przy pomocy kursora lub palcem, z założeniem, że istnieje obiekt startowy (Rysunek 2). Analizie muszą podlegać wyćwiczone wcześniej ruchy, które będą powtarzalne, a obiekt będzie w zasięgu ramienia. Prawo Fittsa można opisać wzorem znanym również jako formuła Shannona [8]:

$$T = a + b \times \log_2(c + \frac{D}{S}) \quad (1)$$

gdzie T to całkowity czas mierzony w sekundach wymagany do realizacji osiągnięcia obiektu docelowego, a to stała opisująca czas mierzony w sekundach potrzebny do reakcji niezbędnej do rozpoczęcia ruchu (opóźnienie), b to stała czasowa mierzona w sekundach związana z prędkością ruchu urządzenia wskazującego, którym steruje użytkownik, c to stała zależna od środowiska człowiek-maszyna, która może przyjmować wartości 0, 0,5 lub 1 (w przypadku systemów informatycznych).

nych będzie to 1),  $D$  to odległość liczona w jednostkach miary odległości (np. cm lub px) od punktu początkowego obiektu do jego środka, a  $S$  to szerokość obiektu docelowego liczona w jednostkach miary odległości (np. cm lub px).



Rysunek 2: Model geometryczny osiągnięcia celu na ekranie [8]

Kluczowe jest, że stałe  $a$  i  $b$  powinny być wyznaczone w sposób empiryczny i mogą przyjmować inne wartości dla różnych urządzeń użytych w badaniu [9].

## 2.2. Ocena z udziałem użytkowników

Poza metodami liczbowymi należy zbadać odczucia użytkownika oraz ergonomię interfejsu graficznego i kontrolerów. W tym celu należy wykonać ocenę z udziałem użytkowników, a więc trzeba zebrać grupę badawczą, która wykona testy i odpowie na pytania w przygotowanym odpowiednio wywiadzie.

Aby przeprowadzić testy użyteczności należy zadbac o następujące składowe:

1. Założenia metodyczne:
  - a) oszacowanie spodziewanych wyników i ustalenie prawdopodobnego wskaźnika satysfakcji użytkowników,
  - b) ustalenie jakie dane będą zbierane podczas testu,
  - c) ustalenie jakie kryteria powinny być spełnione, aby zebrane dane były użyteczne oraz jaki może być próg błędu,
  - d) ustalenie wymagań jakie musi spełniać grupa badawcza.
2. Aspekt etyczny:
  - a) przygotowanie testu o maksymalnej długości 90 minut w celu zapewnienia komfortu psychicznego członków grupy badawczej,
  - b) zapewnienie warunków podczas wykonywania testu, w których członkowie grupy badawczej będą czuli komfort pracy i bezpieczeństwo.
3. Zadania testowe:
  - a) wydrukowanie i omówienie z użytkownikiem zadań, które muszą być klarowne i dobrze zrozumiałe,
  - b) upewnienie się, czy zadanie jest krótkie i czy obejmuje wszystkie założone do zbadania aspekty,
  - c) jeśli do zadania należy użyć materiałów pomocniczych, należy je użytkownikowi wskazać lub zapewnić,
  - d) kolejne zadania powinny być trudniejsze od poprzednich, ale bez przesadnego stopnia trudności. Należy również pamiętać o fakcie, że z różnych powodów użytkownik może nie ukończyć danego zadania lub zadań.

4. Rekrutacja uczestników:
  - a) należy ustalić kryteria szukanych osób do grupy badawczej: liczebność, wiek, płeć, stopień doświadczenia, itp.,
  - b) testy użyteczności należy przeprowadzać w oddzielnych grupach lub pojedynczo,
  - c) określenie wymagań odnośnie do miejsca, w którym testy użyteczności zostaną przeprowadzone, do metod pozyskania uczestników oraz, jeśli jest taka potrzeba, do wysokości wynagrodzenia.
5. Zbieranie danych:
  - a) przygotowanie miejsca i technologii potrzebnej do przeprowadzenia badania oraz zadbanie, by urządzenia były skalibrowane i działały poprawnie,
  - b) zapewnienie pomocnika, który będzie gotów pomóc uczestnikom badania na każdym etapie testu,
  - c) stworzenie mechanizmu zbierania danych lub nagranie testu przy pomocy kamery,
  - d) przeprowadzenie wywiadu po zakończeniu testu użyteczności lub zalecenie wypełnienia ankiety oceny satysfakcji.
6. Miary użyteczności:
  - a) oszacowanie czasów ukończenia poszczególnych zadań przez uczestników testu,
  - b) stworzenie statystyki błędów oraz powodzeń i niepowodzeń dla zakończonych testów,
  - c) wskazanie liczby użytych funkcji i poleceń,
  - d) wskazanie, czy było zapotrzebowanie na pomoc obsługi bądź materiałów pomocniczych, a jeśli tak, to zapisanie czasów dla tego typu zachowań,
  - e) wskazanie liczby powtórzeń testów lub, w przypadku licznych niepowodzeń, szukanie alternatywnych metod rozwiązania problemu,
  - f) zapisanie liczby zgłoszeń dotyczących niezadowolienia oraz utraty kontroli nad sprzętem,
  - g) określenie wszystkich innych miar użyteczności, które będą potrzebne do zbadania interfejsu, ale nie zostały powyżej wymienione [5], [8].

## 3. Opis wykorzystanych urządzeń i technologii

Metody badawcze są niezwykle istotne, jednak są tylko teorią bez środowiska testowego, manipulatorów ruchowych i człowieka, który będzie tej technologii używać. Testy są wykonywane dopiero w momencie, gdy istnieje technologia do zbadania, więc autorzy niniejszego artykułu musieli taką technologię stworzyć lub wykorzystać tę istniejącą na rynku.

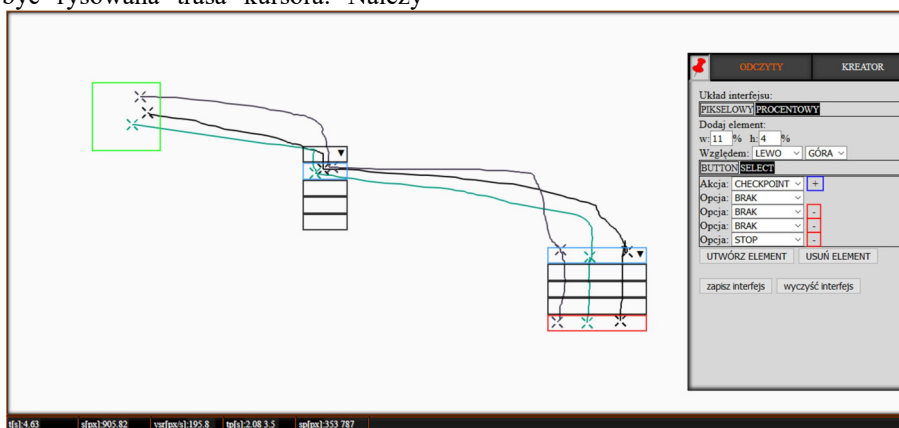
### 3.1. Aplikacja do testowania interfejsów urządzeń

Należy zacząć od opisu środowiska testowego, w którym grupa użytkowników mogłaby zbadać wybrane współczesne kontrolery wejścia pod względem HCI. W tym celu autorzy artykułu przygotowali aplikację webową umożliwiającą przygotowanie odpowiednich testów oraz pozwalającą na zbadanie i zapisanie wyników do dalszej analizy.

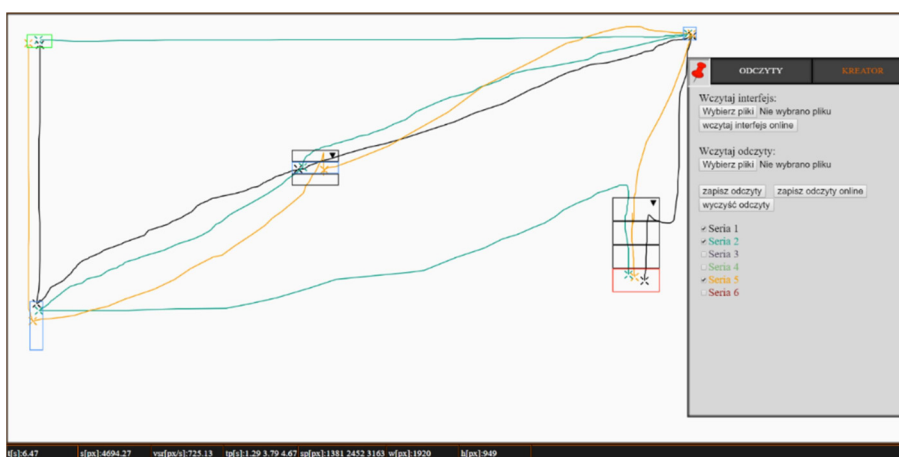
Aplikacja została przygotowana w hipertekstowym języku znaczników HTML w wersji 5 w oparciu o arkusz stylów CSS w wersji 3. Wykorzystano również język javascript oraz stworzoną w nim bibliotekę jQuery. Aplikacja działa w oknie przeglądarki internetowej i umożliwia pracę w trybie offline lub online (w tym przypadku musi być umieszczona na serwerze zewnętrznym). Wykorzystane standardy sprawiają, że konieczne jest używanie stosunkowo nowych wersji przeglądarek internetowych [2], [4].

Celem aplikacji jest możliwość stworzenia środowiska testowego dla kursora systemowego, który będzie sterowany różnorodnymi kontrolerami ruchowymi, zwanymi również manipulatorami ruchowymi. Aplikacja umożliwia stworzenie jednego bloku startu i jednego bloku końcowego oraz dowolnej liczby bloków pośrednich, a same bloki mogą być różnej wielkości. Po najechaniu kursorem na blok startu i naciśnięciu odpowiedniego przycisku (np. w przypadku myszy komputerowej LPM) zaczyna być rysowana trasa kursora. Należy

klikać w bloki pośrednie, a na końcu w blok końcowy. Po naciśnięciu ostatniego bloku trasa przestanie być rysowana pod warunkiem, że użytkownik nie pominął żadnego z wcześniejszych bloków. Cały ekran testowy działa w przeglądarce internetowej, można go projektować ze stałą szerokością lub z zachowaniem zasad responsywności. Interfejs oraz wyniki można zapisywać oraz wczytywać na lokalnym nośniku danych, np. na dysku twardym komputera lub online na serwerze zewnętrznym. Na samym dole aplikacji jest pasek, na którym wyświetlane są zbierane wyniki pomiarów, takie jak: całościowa droga liczona w pikselach, całościowy czas liczony w sekundach, całościowa prędkość liczona w pikselach na sekundę oraz drogi i czasy pośrednie między kolejnymi blokami. Na poniższym rysunku (Rysunek 3) przedstawiono przykład trasy wraz z bocznym kreatorem interfejsów, a na kolejnym (Rysunek 4) przykład innej trasy z bocznymi opcjami umożliwiającymi uwidocznienie lub ukrycie tras.



Rysunek 3: Widok interfejsu umieszczonego na płótnie z przeprowadzonymi trzema seriami badań.



Rysunek 4: Widok interfejsu z przeprowadzonymi kilkoma seriami badań.

### 3.2. Fragment kodu aplikacji

Listing 1: Funkcja umożliwiająca rysowanie ruchu kursora oraz obliczenie szukanych wartości.

```
// wykrycie ruchu kursora
$(document).mousemove(function(event) {
// zapisanie aktualnego czasu
t = window.performance.now();
// jeśli pomiar został rozpoczęty
if(started) {
```

```
// ustalenie warunków początkowych
if(currentMousePos.x == -1 && currentMousePos.y == -1) {
currentMousePos.x = event.pageX;
currentMousePos.y = event.pageY;
temptime = window.performance.now();
loopI = 0;
}
// wyliczenie różnicy czasowej pomiędzy ostatnim wykonaniem
funkcji, a tym obecnym
deltaT = Math.abs(t - temptime);
// nadpisanie nowo uzyskanym czasem poprzedni czas
```



```

temptime = t;
// obliczenie przebytej drogi od ostatniego wykonania funkcji z wykorzystaniem twierdzenia Pitagorasa
var deltaS = Math.sqrt(Math.pow(currentMousePos.x - event.pageX,2) + Math.pow(currentMousePos.y - event.pageY,2));
// dodanie drogi składowej do drogi całkowitej
s += deltaS;
sx += Math.abs(currentMousePos.x - event.pageX);
sy += Math.abs(currentMousePos.y - event.pageY);
// nadpisanie aktualnymi wartościami zmiennych przechowujących starą pozycję kursora
currentMousePos.x = event.pageX;
currentMousePos.y = event.pageY;
// wyświetlenie wartości na ekranie w ustalonych komórkach
$("#droga").text(Math.round((s + 0.00001) * 100) / 100);
$("#drogax").text(sx);
$("#drogay").text(sy);
// wyliczenie szybkości tymczasowej
vtemp = deltaS / deltaT * 1000;
// wyświetlenie wartości na ekranie w odpowiednich polach
$("#predkosc").text(vtemp);
$("#deltaT").text(deltaT);
// dodanie odczytów do tabeli przechowującej dane o ruchu kursora
readings[readings.length-1].push([loopI, currentMousePos.x, currentMousePos.y, deltaT, window.innerWidth, window.innerHeight]);
// rysowanie linii na płaszczyźnie canvas
prevX = currX;
prevY = currY;
currX = event.clientX - canvas.offsetLeft + $(document).scrollLeft();
currY = event.clientY - canvas.offsetTop + $(document).scrollTop();
draw();
// zwiększenie licznika wywołań funkcji
loopI++;
}
});

```

### 3.3. Zastosowane kontrolery ruchowe

Poza środowiskiem testowym, potrzebne są manipulatory, które umożliwiają poruszanie i wykonywanie akcji kursorem systemowym. Na potrzeby artykułu test został przeprowadzony na wybranych urządzeniach o różnej popularności oraz budowie interfejsu wejścia:

- myszka komputerowa Logitech G403,
- touchpad zintegrowany z Dell Latitude E6530,
- manipulator punktowy z Lenovo ThinkPad T590,
- joystick Thrustmaster T.16000M FCS
- pilot Magic AN-MR500G do LG Smart TV,
- beprzewodowy kontroler Xbox One,
- tablet Galaxy Tab S5e z ekranem sAMOLED,
- sensor ruchu Kinect Xbox One,
- kontroler Leap Motion,
- 5DT Data Glove Ultra (z 5 czujnikami) [10].

Przykłady wybranych kontrolerów:



Rysunek 5: Przykład kontrolera Microsoft Xbox One [7]



Rysunek 6: Przykład aplikacji ukazująca model dłoni odczytany przez kontroler Leap Motion [6].



Rysunek 7: Przykład rękawic 5DT Data Glove Ultra [1].

Aby dało się porównać wyniki test został przeprowadzany na jednakowych wyświetlaczach. użytą rozdzielczością było 1080p, czyli 1920x1080px. Większość urządzeń była testowana w systemie operacyjnym Windows 10. Część z urządzeń potrzebowała instalacji odpowiedniego sterownika (z Internetu, najczęściej ze stron producentów testowanych kontrolerów). W przypadku rękawicy 5DT konieczna była modyfikacja istniejącego na rynku sterownika, a Kinect wymagał podłączenia dodatkowego adaptera.

## 4. Badanie interfejsów urządzeń

### 4.1 Grupa badawcza

Pierwszym etapem badań, jeśli wszystko działa pod względem technologicznym, jest zebranie grupy badawczej. Istnieją wymogi co do takiej grupy, muszą być to osoby zorientowane w korzystaniu z manipulatorów, ale i takie, które nie korzystają z nich na tyle często, aby zaburzyć wyniki (obawa o zbyt dobre wyniki przy dużym doświadczeniu). W związku z tym została przeprowadzona ankieta, która pozwoliła wyłonić grupę badawczą. W tego typu badaniach zakłada się, że osoby biorące udział w teście podczas wykonywania zadań są w dobrej kondycji psychicznej i są wyspane. Testy muszą być przeprowadzane w przygotowanym wcze-

śniej laboratorium, aby było możliwe sprawne ich przeprowadzenie bez obawy o awarię, jak i w celu zapewnienia maksymalnego komfortu dla uczestników badania. Po wykonaniu testu niezbędne jest zebranie nie tylko wyników liczbowych, ale również przeprowadzenie wywiadu, który pozwoli ustalić stopień zadowolenia użytkowników w kontekście całego testu i poszczególnych urządzeń oraz na zebranie opinii tych użytkowników.

#### 4.2. Charakterystyka obiektu badań

Założeniem aplikacji, która posłużyła jako środowisko testowe w badaniu, było zebranie danych liczbowych, które można opisać w następujący sposób:

$x$  - liczba składowych tras (pomiędzy kolejnymi obiektami), gdzie  $x \in C$

$C$  - zbiór liczb całkowitych

$y$  - liczba bloków wykorzystanych w badaniu (gdzie  $y_s$  to liczba bloków startowych,  $y_p$  to liczba bloków pośrednich, a  $y_k$  to liczba bloków końcowych), z założeniem że  $y \in C$ , a więc:

$$y = y_s + y_p + y_k \quad (2)$$

$s[px]$  - droga liczona w pikselach jaką pokonuje kursor pomiędzy blokiem startowym, a blokiem końcowym

$s_{px}[px]$  - pośrednie drogi liczone w pikselach pomiędzy wszystkimi blokami testu (w aplikacji nie jest wyszczególniona ostatnia suma wartości tras pośrednich, ponieważ przyjmuje ona tę samą wartość co droga całkowita, czyli wartość  $s[px]$ )

$t[s]$  - całościowy czas wykonania zadania testowego liczony w sekundach

$t_{px}[s]$  - czasy pośrednie liczone w sekundach w sposób analogiczny jak składowe tras  $s_{px}$

$v_{sr}[px/s]$  - średnia prędkość kursora podczas realizacji testu liczona w pikselach na sekundę

seria  $z$  - liczba serii testowych jakie wykonuje użytkownik podczas badania, gdzie pierwsza seria przyjmuje wartość 1, a ostatnia seria wartość  $z$ , z założeniem że  $z \in C$

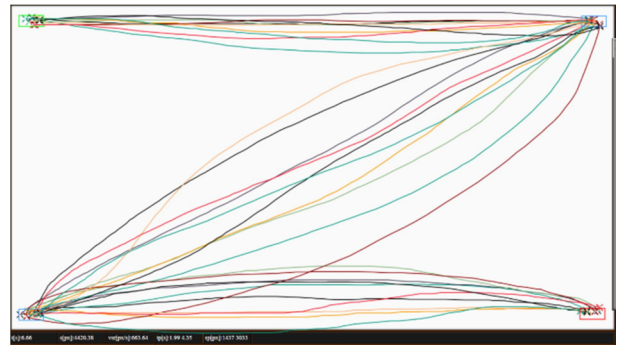
W celu oszacowania możliwej długości trwania testu przy pomocy użytych w badaniu kontrolerów, pomocnicze będzie wykorzystanie wspomnianego w rozdziale 2.1 prawa Fittsa pozwalającego obliczyć czas niezbędny do dotarcia do obiektu graficznego (w tym wypadku dotarcia kursorem od punktu początkowego do środka bloku).

#### 4.3. Rezultat badań

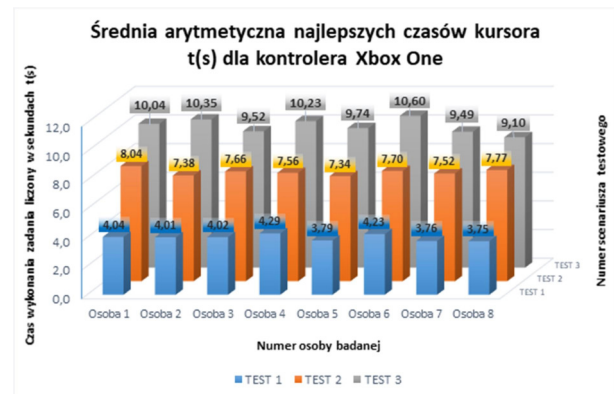
W rozdziale 3.3 wskazano 10 urządzeń, które zostały wykorzystane w badaniu. Samo badanie zostało przeprowadzone na 3 interfejsach testowych o zmiennym stopniu trudności. W celu uzyskania wiarygodnych wyników zostało wykonanych po 10 serii testowych, jednak tylko 5 z nich, które uzyskały najlepsze czasy, było poddanych dalszej analizie. Innymi słowy w całym badaniu każdy z uczestników wykonał 300 serii testowych, ale tylko połowa z nich została użyta do dalszej

analizy. Wyniki zostały zaprezentowane przy pomocy tabel (Tabela 2 - 5) oraz wykresów (Rysunek 8 - 10).

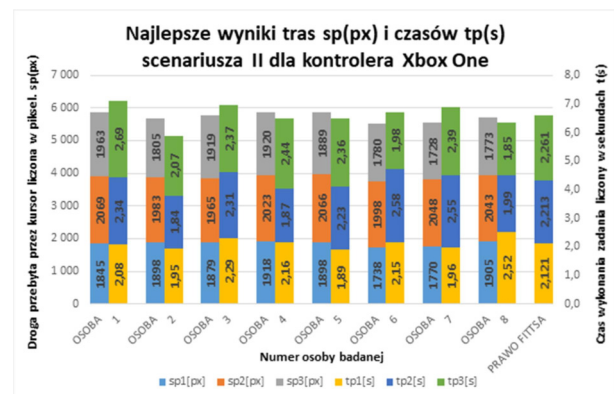
Ze względu na obszerność analizy każdego z kontrolerów podczas badania, w artykule została zaprezentowana jedynie część wyników dotyczących kontrolera Microsoft Xbox One. Pozostałe manipulatory wymienione w rozdziale 3.3 zostały poddane analogicznym badaniom.



Rysunek 8: Przykład wykonanego scenariusza testowego przy użyciu kontrolera Xbox One.



Rysunek 9: Wykres obrazujący średnie arytmetyczne dla 5 najlepszych czasów  $t(s)$  uzyskanych przez 8 testerów w 3 scenariuszach testowych przy użyciu kontrolera Xbox One.



Rysunek 10: Wykres obrazujący najlepsze wyniki tras  $sp(px)$  i czasów  $tp(s)$  uzyskane przez 8 testerów w II scenariuszu testowym przy użyciu kontrolera Xbox One wraz ze spodziewanymi czasami  $T(s)$  wyznaczonymi Prawem Fittsa.

Tabela 2: Zestawienie 5 najlepszych czasów t(s) uzyskanych przez 8 testerów w 3 scenariuszach testowych przy użyciu kontrolera Xbox One.

		UCZESTNICY BADANIA								
		1	2	3	4	5	6	7	8	SREDNIA
Scenariusz testowy 1: czas wykonania zadania w sekundach t(s)	SERIA 1	4,16	3,58	4,47	4,12	3,49	4,27	3,46	4,01	
	SERIA 2	4,06	4,12	4,24	4,28	3,96	4,78	3,97	3,08	
	SERIA 3	4,15	3,82	4,13	4,65	3,81	4,25	3,86	3,66	
	SERIA 4	4,10	4,25	3,60	3,66	3,82	4,15	3,73	3,96	
	SERIA 5	3,72	4,30	3,68	4,72	3,89	3,72	3,76	4,02	
	SREDNIA	4,04	4,01	4,02	4,29	3,79	4,23	3,76	3,75	
Scenariusz testowy 2: czas wykonania zadania w sekundach t(s)	SERIA 1	8,31	7,72	7,28	8,37	7,76	8,11	6,90	7,19	
	SERIA 2	7,11	7,97	6,97	6,47	7,49	6,71	8,16	8,04	
	SERIA 3	8,75	5,86	7,08	8,07	7,85	7,79	7,34	8,25	
	SERIA 4	8,13	7,85	8,25	7,38	7,10	7,71	7,74	8,99	
	SERIA 5	7,88	7,52	8,73	7,53	6,48	8,16	7,47	6,36	
	SREDNIA	8,04	7,38	7,66	7,56	7,34	7,70	7,52	7,77	
Scenariusz testowy 3: czas wykonania zadania w sekundach t(s)	SERIA 1	9,50	9,80	10,56	8,56	9,20	11,38	9,54	8,83	
	SERIA 2	10,39	10,93	10,34	10,99	9,71	9,04	9,90	9,73	
	SERIA 3	10,50	10,98	10,15	10,55	9,78	10,43	9,96	9,47	
	SERIA 4	9,60	10,10	8,02	10,01	10,29	11,64	8,34	9,30	
	SERIA 5	10,23	9,95	8,54	11,04	9,73	10,52	9,73	8,18	
	SREDNIA	10,04	10,35	9,52	10,23	9,74	10,60	9,49	9,10	
	<----	najkrótszy czas				<----	najlepszy uśredniony wynik			
	<----	najdłuższy czas				<----	najgorszy uśredniony wynik			

Tabela 3: Zestawienie parametrów potrzebnych do wyznaczenia czasu T(s) wymaganego do realizacji osiągnięcia obiektu docelowego przy użyciu kontrolera Xbox One oparte o Prawo Fittsa.

PRAWO FITTSA		$T = a + b * \log_2(D/S + 1)$					
TEST 1	D1	915	DLA KONTROLERA XBOX ONE:	a1	0,61	<----	opóźnienie dla trasy nr 1
	D2	950		a2	0,13	<----	opóźnienie dla trasy nr 2
	S1	100		b	0,35	<----	stała czasowa opisująca prędkość ruchu kontrolera
	S2	40					
TEST 2	D1	1790	DLA KONTROLERA XBOX ONE:	a1	0,39	<----	opóźnienie dla trasy nr 1
	D2	1950		a2	0,44	<----	opóźnienie dla trasy nr 2
	D3	1790		a3	0,53	<----	opóźnienie dla trasy nr 3
	S1	60		b	0,35	<----	stała czasowa opisująca prędkość ruchu kontrolera
TEST 3	D1	1720	DLA KONTROLERA XBOX ONE:	a1	0,37	<----	opóźnienie dla trasy nr 1
	D2	955		a2	0,77	<----	opóźnienie dla trasy nr 2
	D3	870		a3	0,07	<----	opóźnienie dla trasy nr 3
	D4	1755		a4	1,53	<----	opóźnienie dla trasy nr 4
S1	25	b	0,32	<----	stała czasowa opisująca prędkość ruchu kontrolera		
S2	100						
S3	25						
S4	95						
		Dx – odległość liczona w pikselach od punktu początkowego do środka obiektu					
		Sx – szerokość obiektu docelowego liczona w pikselach					

Tabela 4: Zestawienie najlepszych wyników tras sp(px) i czasów tp(s) uzyskanych przez 8 testerów w 3 scenariuszach testowych przy użyciu kontrolera Xbox One wraz ze spodziewanymi czasami T(s) wyznaczonymi Prawem Fittsa.

		UCZESTNICY BADANIA									
		1	2	3	4	5	6	7	8	ŚREDNIA NAJLEPSZYCH WYNIKÓW	
SCENARIUSZ TESTOWY 1 (NAJLEPSZA SERIA)	sp1[px]	1065	1122	1248	1215	1001	890	973	995		
	sp2[px]	1063	1082	1246	1106	1040	1070	1047	1012		
	tp1[s]	1,89	2,08	1,65	1,89	1,85	2,01	1,67	1,23		1,784
	tp2[s]	1,83	1,50	1,95	1,77	1,64	1,71	1,79	1,85		1,755
	T1[s]	1,780									
	T2[s]	1,750									
SCENARIUSZ TESTOWY 2 (NAJLEPSZA SERIA)	sp1[px]	1845	1898	1879	1918	1898	1738	1770	1905		
	sp2[px]	2069	1983	1965	2023	2066	1998	2048	2043		
	sp3[px]	1963	1805	1919	1920	1889	1780	1728	1773		
	tp1[s]	2,08	1,95	2,29	2,16	1,89	2,15	1,96	2,52		2,125
	tp2[s]	2,34	1,84	2,31	1,87	2,23	2,58	2,55	1,99		2,214
	tp3[s]	2,69	2,07	2,37	2,44	2,36	1,98	2,39	1,85		2,269
T1[s]	2,121										
T2[s]	2,213										
T3[s]	2,261										
SCENARIUSZ TESTOWY 3 (NAJLEPSZA SERIA)	sp1[px]	1723	1950	1886	1856	1760	2000	1711	1860		
	sp2[px]	974	954	1000	1058	1018	1065	985	930		
	sp3[px]	1309	956	850	853	885	821	930	1061		
	sp4[px]	1911	2403	1821	1765	2180	1836	1745	1920		
	tp1[s]	2,36	2,47	2,01	2,22	2,63	2,72	2,17	2,11		2,336
	tp2[s]	1,90	1,83	1,61	2,07	1,64	2,22	1,77	1,82		1,858
tp3[s]	1,83	1,59	1,76	1,40	1,95	1,71	1,81	1,76	1,726		
tp4[s]	3,41	3,91	2,64	2,87	2,98	2,39	2,59	2,49	2,910		
T1[s]	2,330										
T2[s]	1,858										
T3[s]	1,722										
T4[s]	2,901										
	<----	najlepszy wynik dla danego etapu				<----	najgorszy wynik dla danego etapu				



## 5. Wnioski

Na podstawie przeprowadzonych badań udało się zebrać liczne wyniki liczbowe oraz uzyskać ich wizualizację w tabelach i na wykresach. Po badaniu przeprowadzono wywiad z uczestnikami badania, który uzupełnił suche liczby o odczucia użytkowników, co pozwoliło uzyskać ciekawe wnioski.

Badanymi wartościami była droga kursora przebyta w pikselach, czas liczony w sekundach, prędkość kursora liczona w pikselach na sekundę oraz składowe tych wartości na pośrednich trasach. Pierwszym ciekawym wnioskiem dotyczącym wszystkich kontrolerów jest to, że najlepsze uzyskane czasy najczęściej nie pokrywały się z najkrótszą pokonaną trasą. Okazuje się, że precyzja ruchu kursora nie jest wprost proporcjonalna do czasu wykonania zadania ekranowego. Użytkownicy nie poruszali się po liniach prostych, a po krzywych przypominających łuki oraz fale. Prędkość ruchu kursora miała natomiast kluczowe znaczenie, a ta osiągała najwyższe wyniki przy średniej precyzji ruchu.

Zgodnie z założeniami HCI okazało się, że dopiero zestawienie wyników liczbowych z wywiadem z grupą badawczą pokazało pełny obraz badania. Przechodząc do omówienia wyników liderem pod względem uzyskanych czasów, nie licząc ekranu dotykowego w trybie klikania, okazała się najpopularniejsza mysz komputerowa, ale w opinii użytkowników nie zawsze mogłaby nim być. Uczestnicy badania stwierdzili, że swój wynik myszka osiągnęła tylko dzięki warunkom sprzyjającym poruszaniu się tym kontrolerem, gdyż badanie było wykonywane przy biurku, a mysz komputerowa wymaga płaskiego podłoża. Myszka nie należy też do najbardziej precyzyjnych kontrolerów pod względem poruszania się po liniach prostych.

Analizując kolejne kontrolery o lekko gorszych czasach, ale podobnej specyfice, touchpad umożliwia pracę bez podłoża, a manipulator punktowy taki jak trackpoint pracę w warunkach lekkiej wilgoci lub na mrozie. Joystick uzyskał średnie czasy w badaniu, ale pozwolił na wysoką precyzję ruchu. Nieco podobny kontroler Xbox One pomimo niewiele lepszych wyników czasowych był oceniony jako najprzyjemniejszy w użyciu ze względu na intuicyjny interfejs i wysoką ergonomię, gdyż można z niego korzystać w dowolnej pozycji ciała (np. na leżąco).

Ekran dotykowy przetestowano w trybie rysowania drogi oraz klikania bloków pośrednich. Jak było wspomniane czasy w trybie klikania deklasują nawet mysz komputerową, natomiast tryb rysowania trasy okazał się bardzo przyjemnym doświadczeniem, chociaż uzyskane wyniki czasowe były w nim średnie. Użytkownicy jednak czuli się bardzo komfortowo podczas tego badania na co wpływa zapewne obecna popularyzacja urządzeń dotykowych.

Kolejne kontrolery dały uczestnikom, pomimo uzyskania miernych czasów badania, ogromną satysfakcję i zabawę przy wykonywaniu testów. Kinect pozwolił na niezwykle doświadczenie, gdyż wbudowana kamera precyzyjnie wykrywała dłonie użytkowników, a ruchy kursorem oraz kliknięcia były równie dokładne. Pilot

Magic dał za to doświadczenie w korzystaniu z telewizora jak z komputera. W Leap Motion chociaż sposób poruszania kursorem bardzo zaintrygował uczestników badania, to złe rozpoznawanie gestów i męczące trzymanie rąk nad urządzeniem szybko zniechęciło z jego używania.

Na koniec zbadano rękawicę 5DT, która uzyskała najgorsze czasy, ale pozwoliła na wysoką precyzję ruchu. W opinii użytkowników, gdyby dopracować kwestię kalibracji, rękawica mogłaby się świetnie nadać do zdalnego sterowania maszynami takimi jak dźwigi.

Podsumowując kontrast między wynikami liczbowymi, a odczuciami po badaniu były zaskoczeniem zarówno dla samych uczestników badania jak i jego autorów. Autorzy spodziewali się, że myszka komputerowa nie zostanie liderem pod względem uzyskanych czasów, ale może okazać się ulubionym kontrolerem grupy badawczej, a jest zupełnie odwrotnie.

Niestety autorzy nie przewidzieli, że w przypadku środowiska testowego nie należy zadbać jedynie o aplikację do testowania, ale również o warunki otoczenia. W teście przy biurku to myszka jest liderem pod względem czasów, ale bez płaskiego podłoża zapewne by przegrała, dlatego należałoby powtórzyć badanie w kilku odmiennych warunkach otoczenia. Na szczęście same testy użyteczności wykazały już, że inne kontrolery w pewnych sytuacjach są o wiele lepsze od myszy komputerowej, co autorzy artykułu uważają za sukces. Zostaje jednak niedosyt i rekomendacja przeprowadzenia badań na podobnych zasadach, ale w odmiennych warunkach otoczenia.

## Literatura

- [1] 5DT Data Glove Ultra Controller, [http://5dt.com/wp-content/uploads/2011/06/hw\\_data\\_glove\\_wireless01.jpg](http://5dt.com/wp-content/uploads/2011/06/hw_data_glove_wireless01.jpg), [24.02.2020].
- [2] T. Atkins Jr. i inni, CSS Snapshot 2017, <https://www.w3.org/TR/css-2017/>, [24.02.2020].
- [3] D. Benyon, Designing Interactive Systems - A Comprehensive Guide to HCI, UX and Interaction Design, Pearson ELT (2014).
- [4] I. Hickson i inni, A vocabulary and associated APIs for HTML and XHTML, <https://www.w3.org/TR/html5/>, [24.02.2020].
- [5] S. Krug, Nie każ mi myśleć! O życiowym podejściu do funkcjonalności stron internetowych, Helion (2014).
- [6] Leap Motion Controller, <https://3bonpl1aiidtbao4s10xacvnpengine.netdna-ssl.com/wp-content/uploads/2017/03/leap-motion-3d-motion-gesture-controller-10-large.jpg>, [24.02.2020].
- [7] Microsoft Xbox One Controller, [https://steamcdn-a.akamaihd.net/steamcommunity/public/images/steamworks\\_docs/polish/xboxone\\_controller.png](https://steamcdn-a.akamaihd.net/steamcommunity/public/images/steamworks_docs/polish/xboxone_controller.png), [24.02.2020].
- [8] M. Miłosz, Ergonomia systemów informatycznych, Politechnika Lubelska, 2014.
- [9] M. Sikorski, Interakcja Człowiek-Komputer, Polsko-Japońska Wyższa Szkoła Technik Komputerowych, 2010.
- [10] Wikipedia. Wolna encyklopedia, <http://www.wikipedia.org>, [24.02.2020].