TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Sergei Zarembo 164359

ADAPTATION AFFECT ON FINE MOTOR MOTIONS

Bachelor's thesis

Supervisor: Sven Nõmm
PhD

Tallinn 2019

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Sergei Zarembo 164359

KOHANDAMISE MÕJU PEENMOOTORSETE LIIGUTUSTELE

Bakalaureusetöö

Juhendaja: Sven Nõmm
PhD

Tallinn 2019

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Sergei Zarembo

27.05.2019

Abstract

The main purpose of this work is to study how adaptation affects performing fine motor motion tests, since current methods of data collection are not informative enough. As the solution, have been developed data recording application for MS Surface tablet device.

Using created application is possible to record data about six different parameters with high frequency (about 200 points per second). This application could be used in medicine for determining diseases, which manifest through fine motor motions, for example, tremor of Parkinson’s disease.

Developed application has its own unique features, which can facilitate and automatize drawing data recording, collected data analyzation. Also, data recording style is much more informative, than commonly used methods of similar data recording.

Current version of application has a relatively great set of tasks, which user can select for performing; can visualize drawings; record and save data on user’s device by 6 parameters; support for 4 languages; simple and intuitive UI.

As an additional purpose of this work is data collection with created application, its analysis and comparing. To present calculated data values graphs and tables were created. Also, statistical tests were performed to get an idea about values changes during tasks performing days.

As result was concluded, that created application meets the requirements and can be used for described purposes. In analysis results was concluded, that it is possible to observe changes, that took place during the testing days. However, was made a conclusion, that to get more accurate observe on values changes and classify them, more data should be collected with more participating subjects.

This thesis is written in English and is 36 pages long, including 6 chapters, 13 figures and 4 tables.

Annotatsioon

Kohandamise mõju peenmootorsete liigutustele

Selle töö peamine eesmärk on uurida, kuidas kohandamine mõjutab peenmootori liikumiskatsete läbiviimist, kuna praegused andmekogumismeetodid ei ole piisavalt informatiivsed. Lahenduseks on loodud andmete salvestamise rakendus MS Surface tahvelarvuti seadmele.

Loodud rakenduse kasutamisega on võimalik salvestada andmeid kuue erineva parameetriga kõrge sagedusega (umbes 200 punkti sekundis). Seda rakendust võib kasutada meditsiinis haiguste kindlakstegemiseks, mis avaldub läbi peenmootori liikumiste, näiteks Parkinsoni tõve või treemori.

Arenenud rakendusel on oma unikaalsed omadused, mis võivad hõlbustada ja automatiseerida joonistusandmete salvestamist, kogutud andmete analüüsimist. Samuti on andmete salvestamise stiil palju informatiivsem kui sarnaste andmete salvestamise tavapärased meetodid.

Rakenduse praegusel versioonil on suhteliselt suur hulk ülesandeid, mida kasutaja saab teostamiseks valida; saab visualiseerida jooniseid; salvestada kasutaja seadme andmeid 6 parameetriga; 4 keele toetamine; lihtne ja intuitiivne kasutajaliides.

Selle töö täiendavaks eesmärgiks on andmete kogumine loodud rakendusega, nende analüüs ja võrdlemine. Arvutatud andmete esitamiseks loodi graafikud ja tabelid. Samuti viidi läbi statistilised testid, et saada ideed väärtuste muutumisest päevade testi tegemisel.

Selle tulemusena jõuti järeldusele, et loodud rakendus vastab nõuetele ja seda saab kasutada kirjeldatud eesmärkidel. Analüüsitulemustes jõuti järeldusele, et on võimalik jälgida muutusi, mis toimusid testimispäevadel. Samas tehti järeldus, et väärtuste täpsema jälgimise muutmiseks ja nende klassifitseerimiseks tuleks rohkem andmeid koguda suurem osalevate subjektidega arvuga.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 36 leheküljel, 6 peatükki, 13 joonist, 4 tabelit.

List of abbreviations and terms

DPI	*Dots per inch*
TUT	Tallinn University of Technology
UWP	Windows Universal Platform
MS	Microsoft
UI	User Interface
API	Application Programming Interface
MM	Motion mass
Tm	Trajectory mass
Vm	Velocity mass
Am	Acceleration mass
Jm	Jerk mass
Anm	Angular mass
Pm	Pressure mass
RAM	Random Access Memory
JSON	JavaScript Object Notation
XAML	Extensible Application Markup Language
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
VB	Visual Basic
CPU	Central Processing Unit
SSD	Solid-State Drive

Table of contents

List of figures

List of tables

1 Introduction

Different areas of medicine are using fine motor drawing tests for making possible to determinate deviations and problems, related to fine motor motions. For example, tremor and Parkinson's disease. Nowadays, for diagnostics of those diseases are mostly often used sheet of paper and pen. However, this method might not be informative enough to make comprehensive analysis. With suitable application it is possible to automize data collection and analyzing processes. Besides, collected with certain device data might be much more informative and accurate, than the methods that are used today for these tasks.

The main purpose of this work is to study how adaptation affects performing fine motor motion tests. Since, already existing application and platform application was developed for have their own disadvantages, as solution was developed data recoding application for MS Surface tablet, which can display required for fine motor tests special tasks, record data from user drawings and save it in correct format locally on the user device and have its own features.

As the second purpose of this work, is presented data recording for several tests with created application and its future analysis. The purpose of analysis was trying to find changes in motor performance and determinate adaptation in performed tests.

Created application should meet the requirements, which are described in more detail in further parts of the work. Most requirements are based on already existing application for iPad platform and its functionality: created application should be approximately with the same functionality, drawing tasks sets, should be able to record same parameters and save them in the correct way.

However, in process of development those requirements were changed often, some of the features and requirements, taken from iPad application were removed, but instead were implemented new features, tasks and requirements.

In application development part are described more detailed application requirements, main parts of development process, overview of UI, application validation and information about its installation.

For data analysis data was recordings were produced records for four days. In recordings participated 10 subjects. With collected data were performed analysis: determination of motion mass parameters, their visualization and implementation of statistical tests for getting data about changes between days for subjects and determining how great changes are.

The current thesis contains 6 main sections: Introduction; State of the art, Application development part, where is described application development process and its overview; Analysis part, which contains information about performed analysis, Discussion and Conclusion, where general results and made work are described.

2 State of the art

Before performing current work, already were created and published several researches by other authors, related to this or similar topics. In those researches were determined and described main MM (Motion Mass) parameters, methods of their obtaining and methods of performing analysis. Read about performed researches and their results is possible

In previous researches as solution for data recording, was already created similar application for iPad platform, however, as application, as platform had their own shortcomings and relatively limited functionality. [1], [2]

3 Data recording application

In order to achieve the goal of the thesis specialized software application was developed. This was mostly dictated by requirements imposed by data acquisition.

In this part are presented requirements analysis, device technical aspects, described the process of recording data application creation, used technologies, application requirements, features, application validation methods, building and installation processes.

3.1 Requirements analysis

3.1.1 Hardware

Application should be created and tested on Microsoft Surface Pro Device. Technical aspects of device:

- 64-bit operation system, Windows 10 Pro
- 8.00 GB RAM memory
- Intel(R) Core(TM) i5-7300U CPU @2.60GHz 2.71GHz
- 256 GB SSD
- Intel® HD Graphics 620 (i5)
- Support for Surface Pen, Surface keyboard, Surface mouse, touch

3.1.2 Recording features of iPad application

Already existing application based on iPad tablet is able to record following parameters:

- Tilt angles between pen and tablet surface on two axes (altitude and longitude)
- Pressure, which person exerts on the pen while drawing
- Coordinates by two axes (x/y) in which pen is in contact with tablet surface
- Timestamp in seconds with an accuracy of one millisecond

iPad application is recording array of points with frequency approximately 180 points per second.

According to these requirements has been conducted research, then conclusion was made, that JavaScript meets the requirements set.

3.2 JavaScript application

Firstly, was planned to use JavaScript/HTML/CSS for application creation. For data collecting was used HTML5 Pointer Events API [4], using which is possible to get required data parameters while drawing.

Relying on fact, that JavaScript is mostly used for web development, to make application usable as desktop application, was used Electron framework [5]

Data recording was occurred in the same format, as it was in iPad application.

After creating minimal functionality of application, the testing was conducted. Application has shown ability to record and save data locally in correct format. However, created application has shown incredibly low frequency on data points recoding in comparison with iPad application: about 50 points per second. With future optimization this value was slightly increased. Since the nature of chosen programming language, the performance and frequency of data recording application were relatively low (the highest gotten points recording frequency was about 70 point per second).

Because of low points recording frequency, it was decided to abandon the further development of the application in this language and search for alternatives.

3.3 UWP Application

3.3.1 UWP review

UWP (Universal Window Platform) – is a platform, which allows to develop applications for Windows Operating System based devises, such as Personal Computers, mobiles MS Surface and others, using common API’s for all devices.

The main UWP features:

- Control over different devices resolutions
- Facilitates user data storage on device
- Unification of various input devices (mouse, windows pen, touch)

Windows Universal Platform supports next languages: XAML, VB, C#, C++, HTML, JavaScript. [6]

3.3.2 UWP Application development

3.3.2.1 Chosen language

For application development more detailed were investigated UWP opportunities for developing described application, especially drawing and MS Surface Pen input recording.

After reading documentation and additional searching, it was concluded, that an application that meets the described requirements can be developed using this platform.

From the list of languages supported by platform were chosen XAML for UI creation and C# for writing application logic. This bundle of languages has been chosen because of similar syntax and general similarity with languages already known by author.

3.3.2.2 Data recording API

Most attention was paid for data recording possibilities. UWP provides PointerPointProperties class [8], using which makes possible to get the following input device data:

- Input device type (mouse, pen, touch).
- Tilt angles of input device between x-z/y-z surface, relatively to display surface.
- X/Y coordinates of input device pointer on display in pixels.
- Pressure, that input device exerts on device surface.

PointerPointProperties class also provides additional information about current input device, such as orientation, however, in written application only described parameters are recording, since other parameters would be redundant.

In case of pen as input device, pressure is recorded in range from 0 to 1.0 as C# float datatype. Tilt angles have range from 0.0 to 90.0 degrees in right from perpendicular state of pen relatively to surface and from 0.0 to 90.0 degrees in left from perpendicular as float datatype (in perpendicular state relatively to two axes point would be recorded as (0.0, 0.0) degree). Pressure is recoded in range from 0.0 to 1.0 as float datatype.

X and Y coordinates are recorded in range from 0 to the value of specific device screen resolution as double datatype.

In case of absence of information about a specific parameter by device, for example, pressure or tilt angles, existing fields have default values: pressure 0.5 and angles 0.0.

Since documentation does not give any information about the accuracy the data is recorded with, this value remains unknown.

3.3.2.3 User drawing representation

For user drawings representation initially was used standard C# Canvas Class, which offers to locate graphical elements on it and represent it. Manipulating with this object by code also offers to imitate drawing by placing dots or lines objects on it in pointer current position. However, this method had significant drawbacks:

- With high number of objects on canvas its performance has decreased, and new objects rendering was slowed down and rendering speed was greatly dependent on device performance (this problem was particularly conspicuous with tasks, which should have been presented in form of a large number of points, c.a. 10 000 dot-objects or when user was drawing for quite a long time).
- Drawing representation did not look smooth, but with sharp corners and ripple. Implementation of complex drawing smoothing methods was denied, since it would be slowing the main process of data points recording.

Because of those disadvantages, were investigated other opportunities of user drawings representation.

As result, was found Windows Ink API [5]. From this moment application started using InkCanvas object, which supports pen interaction, renders strokes by itself and has a large list of settings to customize strokes representation. Further versions of application started to use InkCanvas as main object of user drawings representation.

3.3.2.4 Data points recording

Despite on advantages of InkCanvas it has one significant disadvantage. Since it takes all priority over events, basic data recording is based on, data point function should be fired by different way.

As solution in application is started to be used CoreInkIndependentInputSource, which gives opportunity to works in general as with usual event listener (method, which fires when special given event is happening), but it takes priority over InkCanvas rendering process and allows to use data point recording function in proper way.

3.3.2.5 Combining representation and recording

Combining PointerPointProperties class, InkCanvas and CoreInkIndependentInputSource gave significant improvements in application performance and in testing process was found, that application can record data points with frequency c.a. 200 points per second.

3.3.2.6 Data recording and storage format

Data points are recorded and stored in two-dimensional C# array. In comparison with iPad application, developed application has undergone several improvements in data points recording style. In addition to recording data when device surface and a pen are in the contact, application also recording data when pen is hovering in a short distance over device surface. This feature allows to collect data about pen movements in air over the surface and record micro-movements of user, when he or she pulls the pen off the surface. Currently, application allows user to record data points in two a bit different styles: “iPad style” and “Surface style”.

The first data format was implemented in iPad data recording application. In “iPad style” the output array is two-dimensional, however, all data about recorded points is stored only in the first sub-array. The example of this data recording is presented below:

$$DATA = [[subarray\_1]]$$

In “Surface style” data is recorded also in two-dimensional array; however, recording can have more than one subarray. The previous sub-array ends and starts the new one in four cases:

- When the pen is making transition from hovering state to a state of contact with device surface
- When the pen is making transition from the state of contact with device surface to hovering state.
- When the pen is leaving “hovering” area over device surface
- When device is losing connection with active pen.

This data recording style is more informative, than iPad recording style and iPad application, because it can record moments when user pulls the pen off and simplifies future data analyses. The example of this data recording style is presented below:

$$DATA = [[sybarray_{1_{hover}}], [sybarray_{2_{contact}}], [sybarray_{3_{hover}}]]$$

In additional, application saves such data as:

- User id (user’s name, which user enters on the main screen application)
- Timestamp of the performed test (timestamp is recorded in the moment of the first pen contact with device surface in specific test)
- Session, which has default value “MS”, to make the difference with data files recorded with iPad device.
- Hand with default value “M”. In current application version is no option to switch this value.
- Test type – currently performed test name in system.

3.3.2.7 Saving data files

Data files with performed test information is saving then user selects “Next test” button (presented as right-direction arrow). File is saved as .json file in user “Documents” directory in “testdrive” folder. If named folder does not exist, application automatically creates it. File saves with specific name pattern, which includes timestamp in format YYYYMMDD-HHMMSS, user id, performed test name and postfix “MS”.

For parsing json patterns and serializing user data json files was used Json.NET library.

Below it is possible to see some files, which are using described naming pattern:

20190517-161249-subject1-poppelreuter12_thing-MS.json

20190502-152340-subject2-plines-MS .json

20190502-144747-subject3-Letters_s_small-MS .json

3.3.2.8 Tasks formats and their representation

Initially was planned to present tasks to users only using .json tasks patterns by lines on C# Canvas. However, in process of application development some of json patterns were replaced with identic images. Also, some tasks currently are hard coded in application code, such as “two vertical parallel lines” task (because lines have to be located in 20% from left and right screen borders and implement this task from json data file would be impossible) and “multicube” task.

Because of tasks difficulty, figure shapes in some tasks, for example, poppelreuters, points in those json files should be presented by great amount of closely located dots, what causes one problem: their representation takes too much time, c.a from 5 to 15 seconds. To solve this problem, those tasks are presented by images with the same scale. This improves application performance and facilitates “heavy” tasks representation.

Current application has two entity classes for tasks objects. It is possible to divide all tasks in two groups: tasks, which have to be represented by reference json file and tasks, which are using another method of task pattern representation.

For task, which are not represented by json reference file, is written “special tasks controller class”. Using this class is possible to represent not-on-json based tasks by its id in system. Execution of json based tasks representation is going through the Task Representor class.

Since, tasks, which are not based on images, might be represented as by dots, as by lines, tasks representor class can visualize tasks in these two different ways.

Some tasks should be represented as complex tasks: they have subtasks or sub images. Tasks representor class can also represent specified task part by task name and subtask index.

3.3.2.9 Tasks base

Application contains several tasks, which are stored in tasks base and specific tasks base classes.

Each task in tasks base have unique name in the system to present it in tasks choosing menu or in task hint if other accompanying information is missing.

On json based tasks have a list of json pattern file names, which are participating in task visual representation and parameter, which determinates would specific task be represented in task representor class by dots or by lines (by default tasks are represented by lines).

Special tasks have all attributes, that json based tasks have, but in additional they have unique id number in system, which is used in special tasks controller class and unlike json based tasks, they might have list of image files (if task is based on image or images). Due the lack of opportunity to set image patterns coordinates to locate them in specific place on canvas, special tasks base also responsible for pattern images shifts by x/y coordinates.

All json and image tasks patterns are loaded from application assets directory.

In application the following tasks are implemented and available to perform:

- Two vertical lines test (user should draw two horizontal lines between them)
- Spiral (user should draw a spiral inside another visualized spiral)
- Letters (user should repeat presented letters)
- Continuing different presented patters
- Tracing different presented patterns
- Coping different presented patterns
- Coping presented honeycomb figure

- Finding and tracing presented cube in the cubes grid
- Writing numbers in sequence from zero to nine
- Drawing clock in accordance with the additional text of the task
- Different poppelreuter tests (user should find shown figure in interlacing with other figures and trace it [3])

3.3.2.10 Tasks sequence controller

In current application version there are implemented several tasks. User is free to select tasks in any combination he or she wants to. However, tasks sequence is not sensitive to the order of choosing. Tasks sequence is controlled by single method, which uses tasks representor, special tasks controller classes, current task and subtask indexes to represent specified tasks in correct order in suitable for it time.

3.3.2.11 User settings saving

Application can save user settings, such as last chosen language, tasks sets, data recording style and custom tasks blocks.

This feature is based on ApplicationDataContainer class [16]. Changes made by users are saving only on the user’s device.

3.3.2.12 Language support

In application has been implemented different languages support. Currently, application supports four languages: English, Estonian, Portuguese and Russian.

User can select preferred language on application start page by selecting globe icon and choosing language in appeared submenu (as default is selected English).

Only tasks names and additional task description are translated in named languages. User interface parts are not translated. However, most of the buttons have corresponding icon, what facilitates using application and makes it intuitively understandable.

Translations are saved locally in application assets folder by separate json files. Names of language json files and languages identifications in system are named according to ISO 639-2 standard language codes [9].

3.3.2.13 User Interface

Developed application have three main screens:

- Main page
- Tasks list
- Drawing screen

On the first page user should enter his or her name, which will be saved and used to record json data file with specified username for future specific user identifying in data analysis.

Selecting “Three horizontal lines” button user can exit from application.

By selecting “globe” button user can select the language he or she prefers.

By clicking “next” button user will be navigated to the next (Tasks list) page.

On the picture below is possible to see application main page.

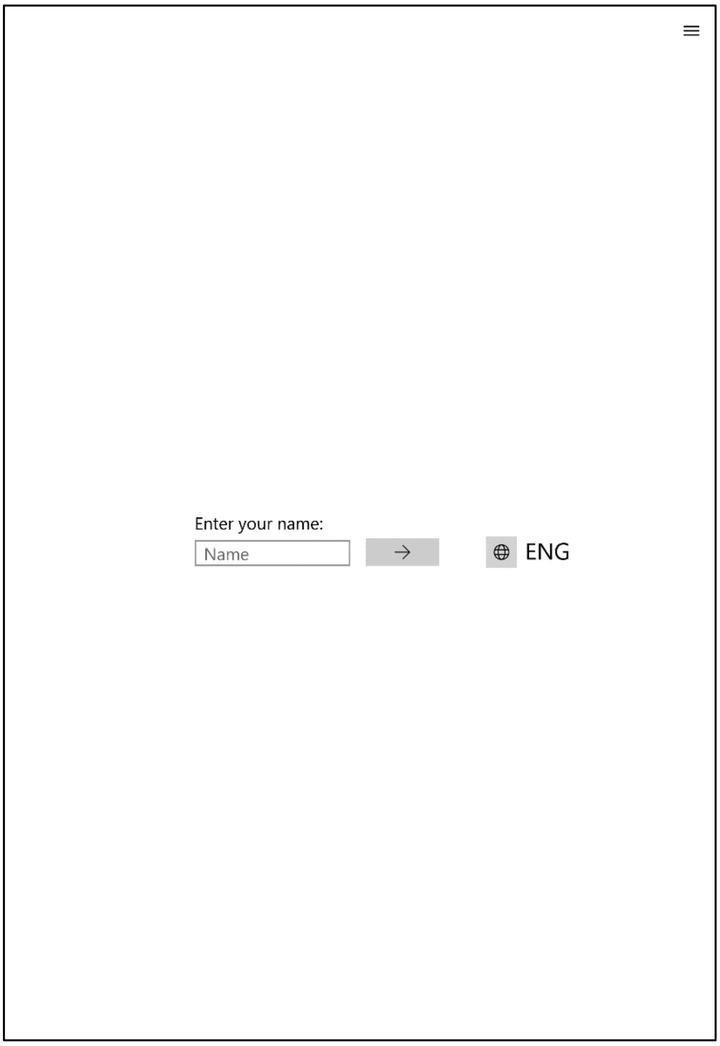


Figure 1. Main page of application with current version of user interface

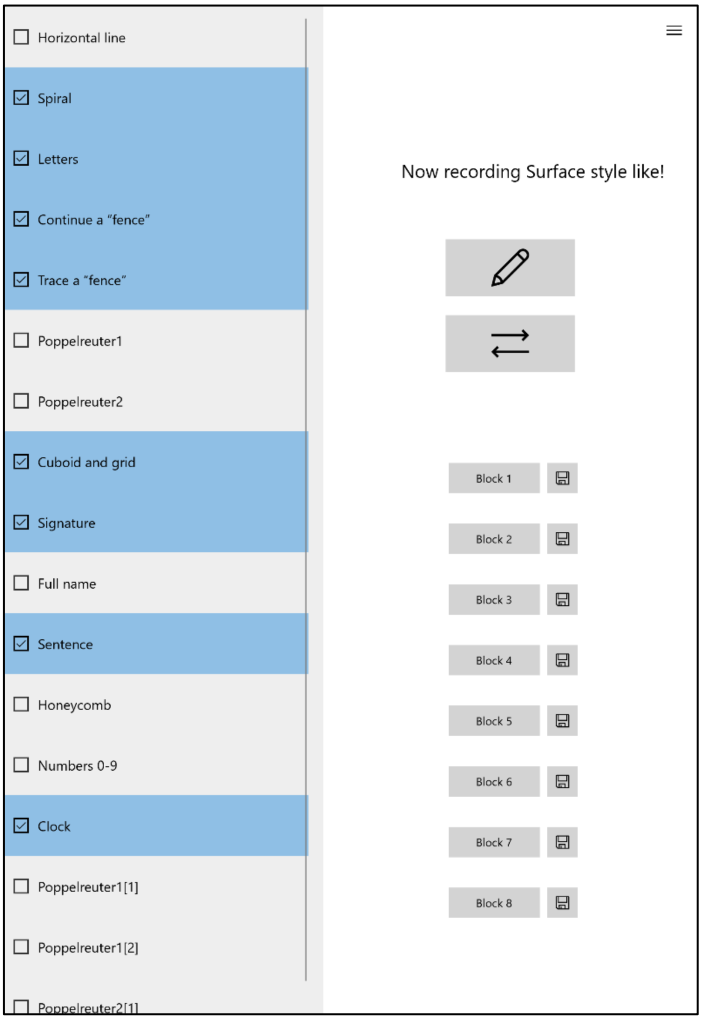
On the second application page user can select tasks from list in the left he or she is interested in by clicking on them. Selected tasks have blue background.

User can switch to another task block by selecting “two arrows button”. If user want to change data recording style, he can do it by selecting three horizontal lines button in top right corner and selecting needed style in opened submenu. Currently active recording style is specified in the middle of the screen. Through submenu user also can navigate to the main page or exit from the application.

On this page user can load previously saved sets of tasks or save new sets instead of existing by selecting “save” icon in the right from specific set button.

By clicking “pencil” icon user will be navigated to the Drawing page.

Below is presented screenshot of the tasks list page.



□ Horizontal line

☑ Spiral

☑ Letters

☑ Continue a “fence”

☑ Trace a “fence”

□ Poppelreuter1

□ Poppelreuter2

☑ Cuboid and grid

☑ Signature

□ Full name

☑ Sentence

□ Honeycomb

□ Numbers 0-9

☑ Clock

□ Poppelreuter1[1]

□ Poppelreuter1[2]

□ Poppelreuter2[1]

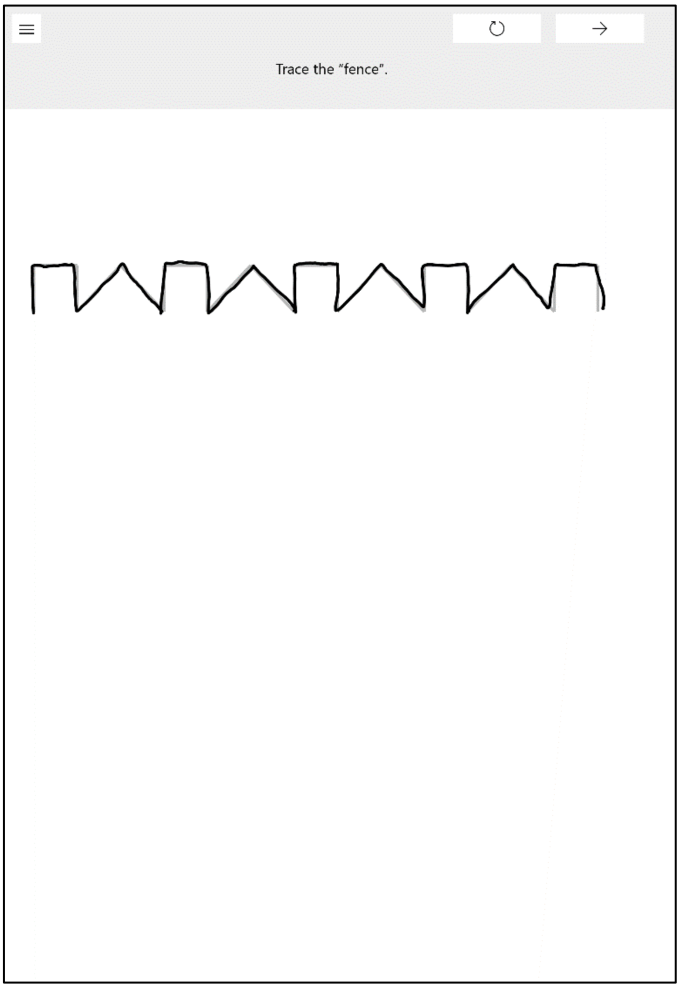
Now recording Surface style like!

Figure 2. View of the second application page: tasks list

On the drawing page user should complete tasks selected on the previous page. First selected task is displayed automatically. By selecting “next” button user progress is saving and new json data file is recorded. After this action, new task from selected tasks list will be displayed until user completes all selected tasks. User can not perform the next task if no visible strokes were made during current task.

If user wants to redo current task, he or she can select “repeat” button, which will clear canvas and delete all user progress for this task.

By selecting “Three horizontal lines” button user can navigate to the main page, tasks list or exit from the application. Below is presented the screenshot of the application drawing page with pltrace task performing:



Trace the "fence".

Figure 3. View of the application drawing page with performing pltace task

3.3.2.14 User flow diagram

Below you can see user flow diagram, which shows user's movements in system.

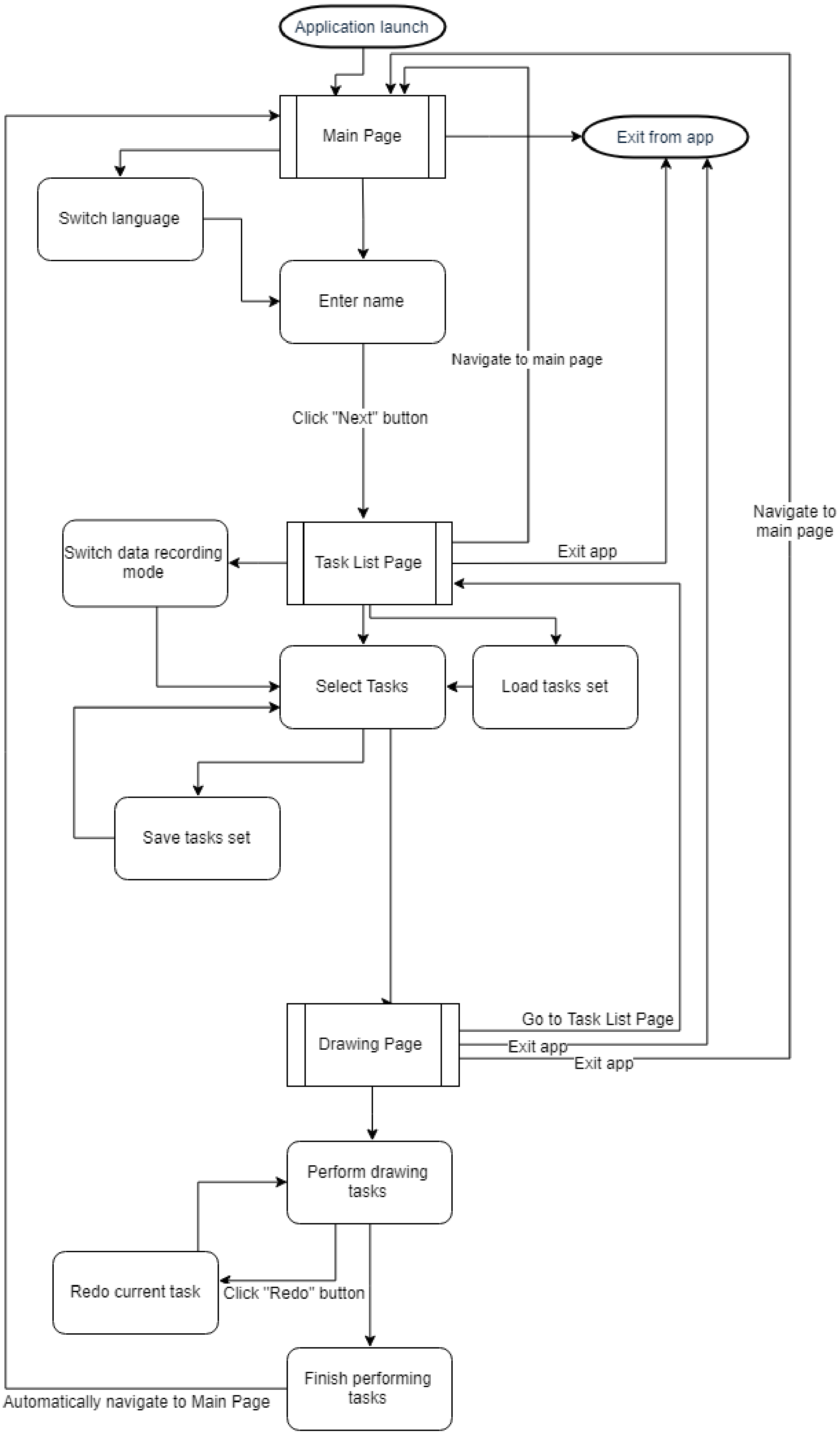


Figure 4. Diagram of user’s movements in application.

3.3.2.15 Application building, distribution, installation

Application is built using Visual Studio platform tools. The final application components are located in single directory and ready for distribution. Package contains application bundle, which is needed for installation and Microsoft certificate, related to application. As additional in package are located dependencies for installation if they are missing on certain device.

To install application person first should install certificate, related to it. To do this person should find in application installation package file with .cer extension and open it. In appeared window person should select “Install Certificate…” button, then in opened dialog checkbox Store Location select “Local Machine”, click next and prove action as administrator. In next opened page person should select “Place all certificates in the following store” checkbox and click on the “Browse” button. In opened dialog person should select “Trusted Root Certification Authorities” and click “OK”. Then in opened window click “Next” button, then “Finish” button. On this moment system should inform person, that certificate installation was successful.

Next, person should click on application .appxbundle extended file to start application installation. In opened window person should click next buttons until application starts installation. At this moment application should be installed on device.

If the process of installation was stopped by operation system because of lack frameworks needed for application work, person should go to dependencies folder in application package, then go to the folder corresponding to its operating system and install all .appx bundles in this directory.

To this moment application should be installed on the person’s device and be ready to use. User can find application by searching its name in Windows Search.

3.3.2.16 Application validation

Application performed geometrical validation with existing iPad application. To compare recorded data from iPad and created application for MS surface two subjects were performing in same tasks sets on both devices. Then data from both devices was visualized on MS application.

Below are presented two parts of screenshots with visualized data:

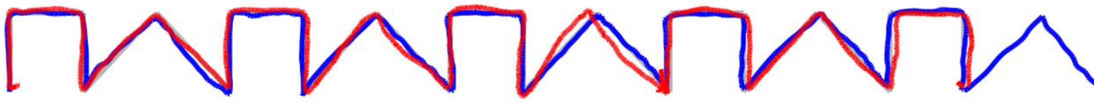


Figure 5. On the part of screenshot is presented subject’s performing in pltrace task on iPad (blue color) and MS Surface (red color).

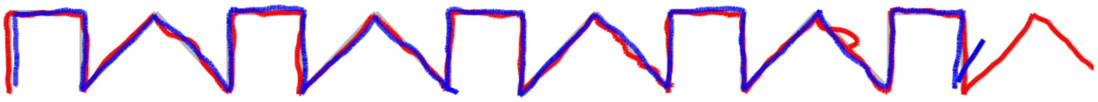


Figure 6. On the part of screenshot is presented another subject’s performing in pltrace task on iPad (red color) and MS Surface (blue color). The tail in the end of blue curve is represented pen hovering mode.

It is possible to see, that in both cases the visualized data from iPad application is longer in the right part. This is because tasks on iPad are represented in album orientation mode, but in MS application all tasks are visualized in portrait mode and original pattern does not fit on the screen. For example, pltrace task is chopped by few segments in right. However, parts existing in both patterns overlaying each other with very high level of matching. Also, represented drawings fit to the original pattern (light gray on images). Thus, it is possible to proof, that created application can record, save and represent data points coordinates in the correct way.

With some other task patterns, it is relatively difficult to validate application in the same way, because certain tasks were slightly reformatted and were shifted by x/y coordinates or chopped, to make tasks visualization nice and make them located closer to the center of the screen.

The most different drawing performing are making pens tips, since they are made from completely different materials: the pen tip for iPad is make from plastic material, but the pen tip for the MS Surface is made of silicone-like material. When drawings with iPad are performed, the friction force between the tip and the surface is small enough, compared to MS Surface. In comparison with iPad, friction force between silicone tip and device is significantly greater and feels like more realistic drawing, that can be also advantage of using created application and MS Surface for data recording.

For full scale application validation against the existing iPad application would be required complicated mechatronic setup and does not belong to the framework of the present research. Nevertheless, geometrically samples acquired by means of MS Surface are consistent with those acquired in iPad.

3.3.3 Application development conclusion

As the part of this thesis, was created, tested and validated data recording application for MS Surface tablet. At first, application should have repeated functionality of already existing application for iPad device, however, during development, the application has deviated from this course and for current moment it differs significantly from the original version: in application was changed display orientation mode, developed unique user interface, implemented new tasks, which are different from iPad version, but already existing tasks were also implemented or adapted for better appearance and changed files naming system.

One of the most important features of application is new recording mode, which is much more informative, than implemented one for the iPad.

In this part of thesis document were described requirements for created application, process of in development, main features of application, it’s building, installation and validation.

4 Data Analysis

In this part are described the way how data was collected, sorted, used methods and technologies for it analysing, main features for analysing, performed statistical tests. Also presented results of analysis.

4.1 Experimental settings

For data collection was used created application, which is described in part 2.3. In data collection participated 10 subjects, who are predominantly TUT students.

Subjects were performing in 4 tasks, implemented in application, four days stretch.

4.2 Tasks description

Below is described four tasks subjects were performing:

- Drawing spiral inside another spiral (name in system: spiral): subject should draw spiral inside another shown spiral from outside to the inside. The purpose of this task is to get representation of subjects drawing skills in smooth curves drawing.

- Continuing “square-triangle” pattern (name in system: plcontinue): subject should continue shown pattern from its ending point. This task is of interest to perform, because each subject has his or her own eye accuracy, understanding of the endpoint to continue drawing and habit of task performing.

- Tracing “square-triangle” pattern (name in system: pltrace): subject should trace shown pattern from start point to the endpoint. Main purpose of this task is to define subject’s behavior in task performing, because in this task subjects have strong instructions for task completing, so they less use their eye accuracy, in comparison with plcontinue task.

- Tracing “grid” pattern (name in system: multicube): subject should trace shown in the bottom part of screen pattern. Because of complex pattern figure, each subject will have his or her own behavior and sequence of drawing pattern fragments.

4.3 Recorded parameters

Since data was collected using created application, there were collected 6 main parameters feature analysis are based on: x/y coordinates, x-z/y-z tilt angles, pressure and timestamp of each data point.

Recorded parameters are described more detailed in part 2.3.2.2 Data recording API.

4.4 Motion mass parameters methods

Motion mass parameters are parameters, which are describing smoothness and general amount of movements performed. Those parameters can give the observation over real data recording and return value for certain fine motor test performing. Parameters computations are based on array of points and sequence joins between them. [8]

In analysis of this thesis there are 6 motion mass parameters involved: trajectory mass, velocity mass, acceleration mass, jerk mass, pressure mass and angle mass.

Trajectory mass is a sum of movements distances performed by subject in test. In the other words, trajectory mass value represents the distance of pen movements, when it was in contact with the surface. The equation of trajectory mass is presented below:

$$Tm = \sum\nolimits_{i=1}^{T} L_i$$

where *i=1* is start point of drawing, T is the last point and

$$L_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

where *x* and *y* with indexes are coordinates of recoded data points.

Velocity mass is a sum of velocities absolute values in collected data points. The equation of velocity mass:

$$Vm = \sum_{i=1}^{T} |V_i|$$

where $V_i = \frac{L_i}{\Delta t_i}$ and $\Delta t_i = t_{i+1} - t_i$

Acceleration mass is a sum of accelerations absolute values in collected data points. The equation of acceleration mass:

$$Am = \sum_{i=1}^{T} |A_i|$$

where $A_i = \frac{V_i}{\Delta t_i}$

Jerk mass is a sum of all absolute jerk values in collected data points. Value of jerk in each data point equals to derivative of acceleration in this point. The equation of jerk mass is presented below:

$$Jm = \sum_{i=1}^{T} |J_i|$$

where $J_i = \frac{A_i}{\Delta t_i}$

Angle mass is a sum of all angles (by three points) between datapoints in collected data points array (on surface).

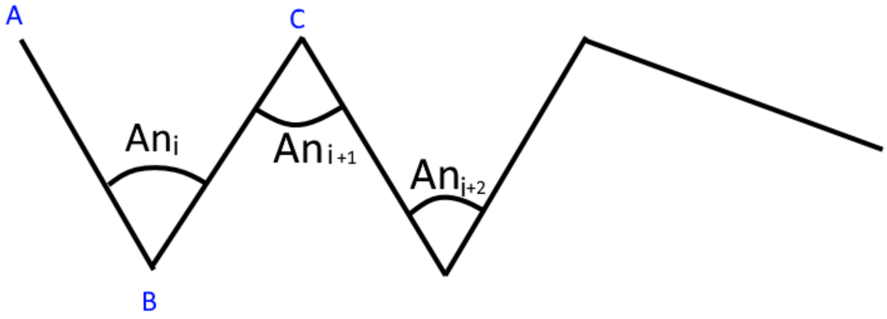


Figure 7. Example of the angle by three points. Angle $An_i => ABC$

The equation for angle mass:

$$\text{Anm} = \sum_{i=1}^{T} An_i$$

where An_i is an angle for each recorded data point by three points and

$$An_i = \operatorname{acos}(\frac{\overline{BC}\ *\ \overline{CD}}{\overline{|BC|}\ *\ \overline{|CD|}})$$

Pressure mass is a sum of absolute differences pressure values between datapoints i and $i+1$. The equation for pressure mass:

$$Pm = \sum_{i=1}^{T} |\Delta p|$$

where $\Delta p = \ p_{i+1} - p_i$

4.5 Used technologies

For collected data analysing and visual representation as graphs was used Python and its modules, such as math [10], numpy [11], scipy [12], matplotlib [13].

4.6 Parameters calculation

First, for parameters calculation, recorded parameters were ordered by timestamp and task name in file name. This structure was created for more comfortable data analysis. With current version is possible to calculate 6 described motion mass parameters.

For each motion mass parameter were created line graphs, which represent subject's behavior changes in certain tasks performing with time (each subject is visualized with different color), where on x-axis are located 4 days and on y-axis values represent chosen parameter value for certain subject in each of 4 days.

Since within this thesis only the relative values and their differences are of interest, all y-values are represented in MS Surface pixels.

4.7 Statistical analysis method

For statistical analysis were implemented two statistical methods and tests: one sample T-test [14] and Welch's test [15]. Both tests are giving t-values, which represent result of the test and p value, which determinates the probability of getting approximately the same value or extremely different with current gather of data.

According to the p-value of perform statistical test, it is possible to conclude do means of values or classes are different with each other. For statistical testing was chosen significance value of 0.05. For all p-values, which are greater, than this value it is possible to accept null hypothesis (H_0), which says, that differences by certain parameters between recorded data days are not significant. Otherwise, if p-value is smaller, than chosen value of significance, the alternative hypothesis (H_1) is accepted. According to alternative hypothesis, values differences between data recorded days for certain parameters are signifyingly great.

4.7.1 Single sample T-test

Single sample T-test is used to determinate if certain sample value is statistically different from sample midpoint value. The equation of one single sample:

$$t = \frac{\bar{x} - \mu}{\sqrt{\frac{s^2}{n}}}$$

where $\bar{x}$ is the sample mean, μ is certain population mean and n is sample size.

4.7.2 Welch's test

Welch's test is a special form of Student's T-test. It is used to describe the differences between two means (classes). The equation of Welch's test is presented below:

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}}$$

where $\overline{X_1}$ and $\overline{X_2}$ are samples means, N_1 and N_2 are samples sizes.

4.8 Analysis results

Below are presented results of calculated parameters and statistical methods values.

4.8.1 Calculated Parameters

Below are presented graphs for each calculated motion mass parameters for pltrace task, where every line represents changes in certain subject behavior. For some graphs there is no strongly visible differences between subjects, since some of the values are high, with comparison with other values and graph lines are overlaying each other. Despite on this fact, there are still great differences between values of those lines.

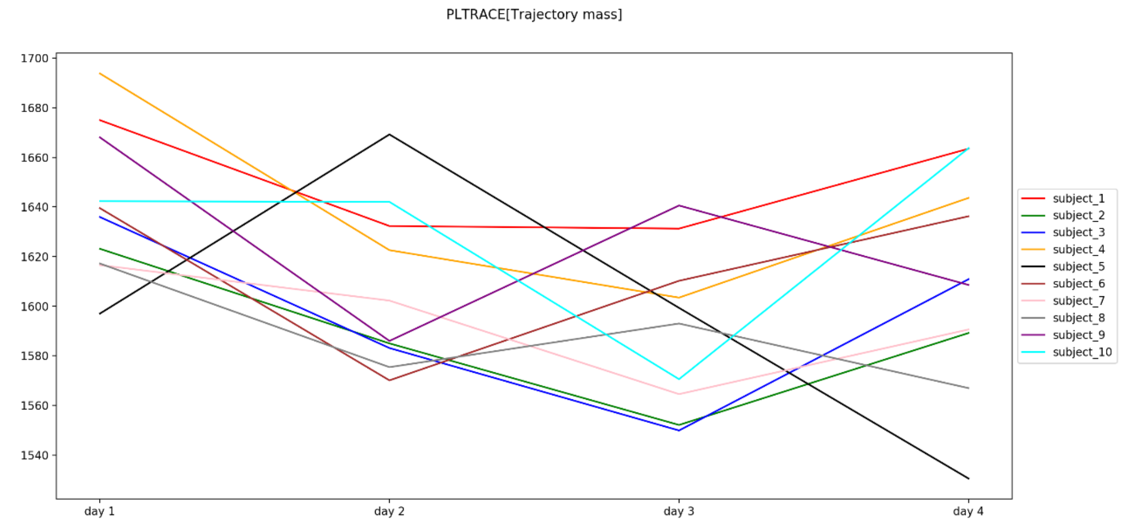


Figure 8. Graph of pltrace task for trajectroy mass parameter for every subject per day

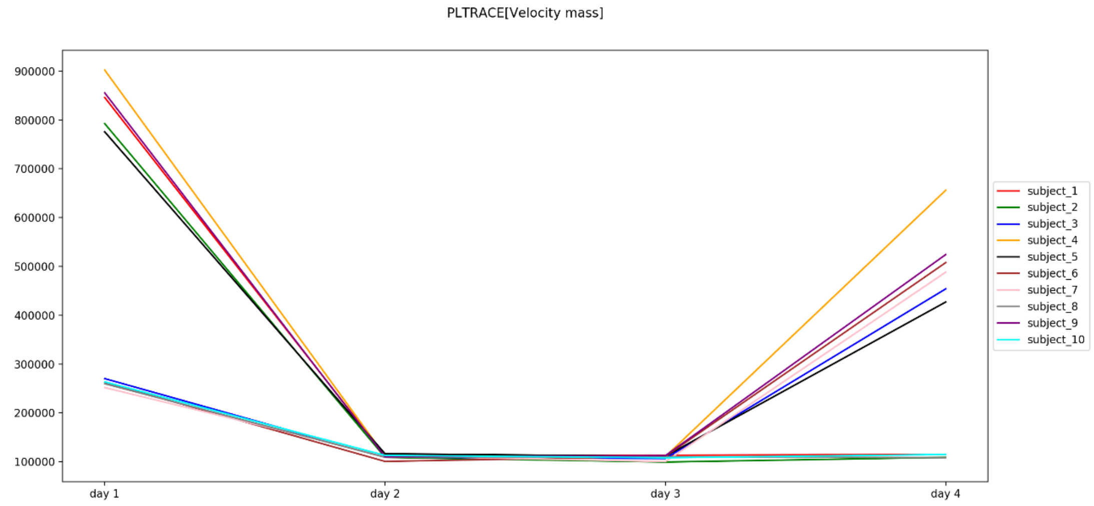


Figure 9. Graph of pltrace task for velocity mass parameter for every subject per day

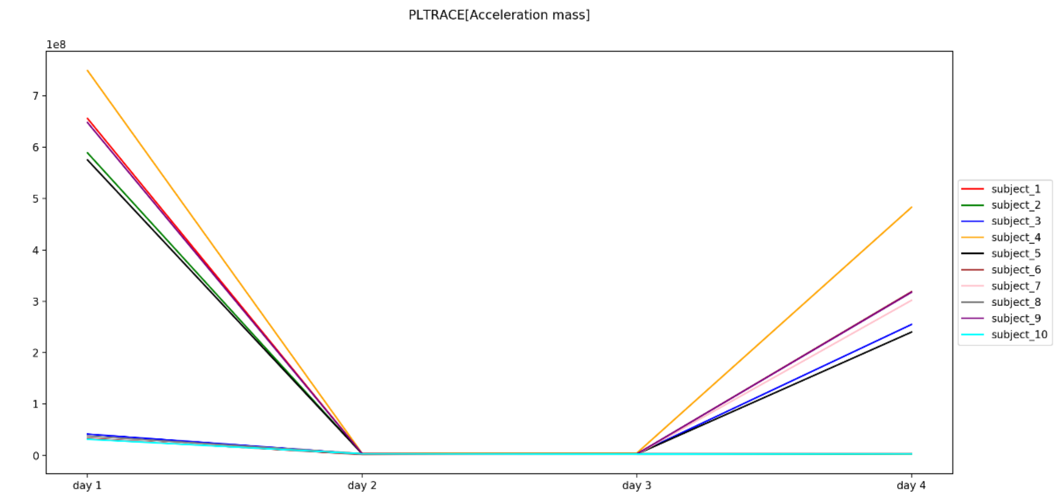


Figure 10. Graph of pltrace task for acceleration mass parameter for every subject per day

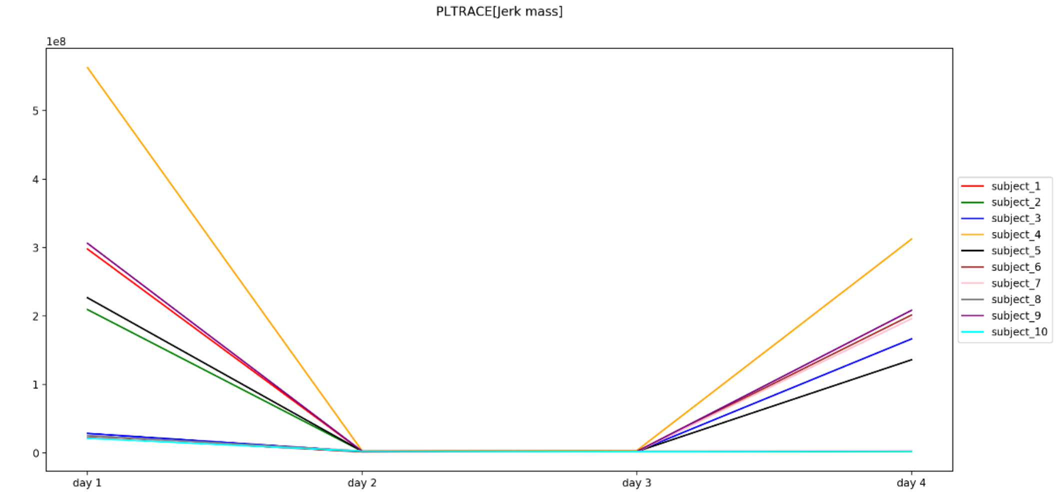


Figure 11. Graph of pltrace task for jerk mass parameter for every subject per day

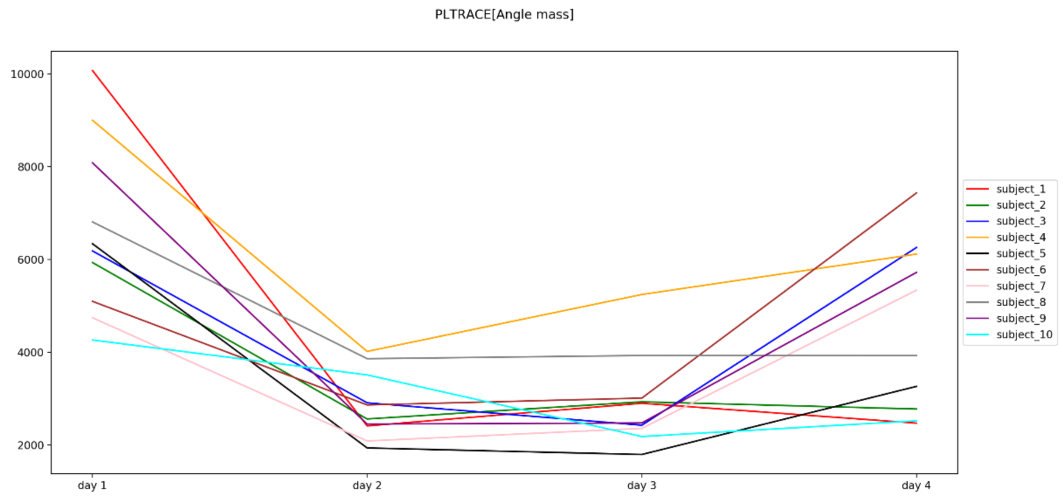


Figure 12. Graph of pltrace task for angle mass parameter for every subject per day

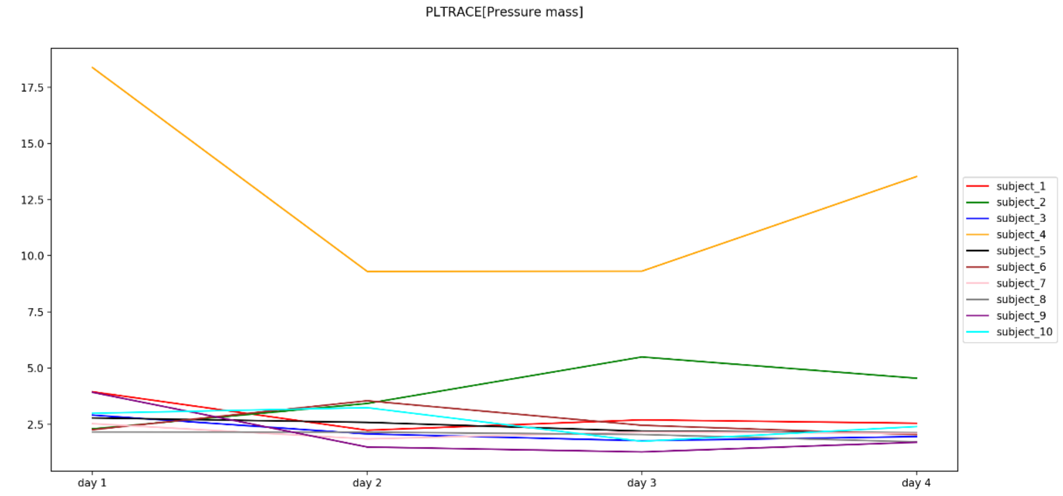


Figure 13. Graph of pltrace task for pressure mass parameter for every subject per day

According to presented graphs, some parameters are not significantly changed in four days. For example, trajectory and angle masses have approximately the same spread of values as on the first day, as on the last.

With pressure mass, there is slightly different situation: majority of people had about the same value for every day, exceptions are only subject_4 and subject_8, but most of the values arranged more closely to each other on the fourth day, than on the first. Also, on the fourth fay values are located lower on the graph and it means, that over time people began to put less pressure on the pen.

On graphs with velocity, acceleration and jerk masses possible to see pattern of behavior for some tested subjects. If on the first day of testing almost all values are widespread, which is possible to explain with the fact, that for some tested subject this might be the first experience of completing similar drawing tasks or even the first experience of using such devise, on the last day values are still heavily scattered. But what is most interesting is that those parameters, computed for the second and third day are about the same and not so high, as for the first and the last days of testing. Basically, for one group of subjects those parameters were relatively low throughout the testing time, however, for the other group those parameters were high on the first day, then for next two days they dropped down and for the last day they increased again to the starting level.

4.8.2 Statistical methods result

Below are described results of statistical methods.

In case of single sample t-test, as the sample were used vectors of differences between days (vector from values per subject, which were gotten in first day and vector from values for each other day were compared) by different motion mass parameters.

For example, first vector of difference (between first and second days) for pressure mass with three subjects would look like the following:

$$V_1 = (D_2(S_1)_{Pm} - D_1(S_1)_{Pm},\ D_2(S_2)_{Pm} - D_1(S_2)_{Pm},\ D_2(S_3)_{Pm} - D_1(S_3)_{Pm})$$

where D_i is day of data recording and S_i is certain subject.

In case of Welch's test as classes were used vectors, built from values of special parameter per subject. As in case of single sample t-test, those vectors were compared with each other. For Welch's test the first class for pressure mass to compare would look this way:

$$V_1 = (D_1(S_1)_{Pm},\ D_1(S_2)_{Pm},\ D_1(S_3)_{Pm})$$

In results of both statistical tests are presented t-values and p-values tables. With value of significance 0.05, p-values which are included in this value are marked with light blue color and it means, that in case of t-test those values are significantly different from sample midpoint and in case of Welch's test two means are very different from each other. It is possible to conclude, that received values greater than significance value, are relatively similar to each other and not significantly different from sample midpoint or two means are relatively similar.

According to statistical hypothesis, in case of p-values, which are higher, than 0.05, the null hypothesis (H_0) is accepted, which says, that changes occurring during data recording are not significant different and parameters calculated for certain days are approximately the same. Otherwise, should be accepted the alternative hypothesis (H_1), which says, that certain parameters values from recorded between two days for certain parameters are quite different.

4.8.2.1 Results of the single sample t-test

Below are presented table of t-values and p-values of t-test.

T-values:

Table 1. t-values for single sample t-test for pltrace task for each MM parameter

Parameter	Differences between day1, day2	Differences between day1, day3	Differences between day1, day4
Tm	2.3901	5.1640	3.5135
Vm	4.578	4.5964	1.7881
Am	3.2829	3.2867	1.3557
Jm	2.9419	2.9489	0.9019
Anm	1.2989	1.3211	1.6890
Pm	6.1455	7.1391	2.4277

P-values:

Table 2. p-values for single sample t-test for pltrace task for each MM parameter

Parameter	Differences between day1, day2	Differences between day1, day3	Differences between day1, day4
Tm	0.0406	0.006	0.0066
Vm	0.0013	0.0013	0.1074
Am	0.0095	0.0094	0.2082
Jm	0.0164	0.0162	0.3906
Anm	0.2263	0.2191	0.1255
Pm	0.0002	0.0001	0.0381

4.8.2.2 Results of the Welch's test

Below presented table of t-values and p-values of Welch's test.

T-values:

Table 3. t-values for Welch's test for pltrace task for each MM parameter

Parameter	Day 1, Day 2	Day 1, Day 3	Day 1, Day 4
Tm	2.4088	3.5788	1.8406
Vm	4.5550	4.5758	1.6812
Am	3.2816	3.2850	1.2566
Jm	2.9399	2.9441	0.7252
Anm	0.7136	0.7450	0.4908
Pm	5.9486	5.5296	2.5105

P-values:

Table 4. p-values for Welch's test for pltrace task for each MM parameter

Parameter	Day 1, Day 2	Day 1, Day 3	Day 1, Day 4
Tm	0.0270	0.0022	0.0842
Vm	0.0014	0.0013	0.1119
Am	0.0095	0.0095	0.2297
Jm	0.0165	0.0164	0.4795
Anm	0.4885	0.4693	0.6300
Pm	0.0001	0.0001	0.0218

4.8.2.3 Conclusion for statistical test

In both statistical tests p-value tables with significance value 0.05 is observed, that with comparing values for two days, about the same values are staying up or under significance value. For example, in first and second columns almost all parameters values are staying under 0.05 and it means, that values means from comparison day 1, day 2 and day1, day 3 are very different from each other. The better situation is with Angular mass values – p-values for this parameter are greater, than significance value. In this case, it is possible to conclude that means values for Angular mass are slightly similar. Also, almost all parameters values in the last columns are greater than 0.5 and it has a meaning, that the most similar result with the first day of recording were received in fourth day.

4.9 Conclusion for data analysis

In analysis part were described conditions under which data was recorded, general description of the subjects involved in tasks, used methods and technologies for data analysis, description of motion mass parameters, general description of the performed, performed calculation of motion mass parameters. Also, for pltrace task were visualized calculated parameters for all subjects per each data recording day for 6 parameters as graphs.

In statistical tests part were performed t-test for one sample and Welch's statistical tests. As result, was concluded, that first have more similarity with fourth day, than with any other day.

Obtained values are of interest for further research. It is possible to see the trend of adaptation for second and third days, however, on the fourth day significant part of the group of subjects deviated from the general trend and returned to the approximately to the same values for velocity, acceleration and jerk masses.

Because the task of subject's behavior classification, and main behavior trends analysis was not the main point of this thesis and due to the lack of a sufficiently large amount of data, for current moment clear conclusion about subject's behavior and adaptation in similar drawing tasks performing cannot be made. This task needs future research and more detailed analysis for creating model of subject's behavior. To get more accurate results, the testing should be performed more than 4 times and more subjects should participate in it.

5 Discussion

To the moment of writing this document, data recoding application was developed. It has its own features and advantages, in comparison with iPad application. However, application still can be improved. For example, it is possible to implement new tasks and functionalities.

With current data set the clear conclusion about people's adaptation cannot be made, since there is small amount of data was collected, but classification of behavior was not the main purpose of this work. Because of that, more data recordings should be made with greater value of participating subjects.

In future, it might be possible to perform more detailed analysis, perform calculations for more motion mass parameters, determinate different groups of subjects by performing tasks behavior, create classifications for those groups and use methodology of machine learning to predict future possible results for subjects.

6 Summary

The present work relates to recognition of fine motor problems, it's recording, analyzing people's behavior and adaptation for fine motor tests performing. The work consists of two large parts: developing data recording application and analyzing data, recorded with created application.

Application is for MS Surface tablet, in accordance with the requirements. At first, was planned to create application, which would copy functionality and tasks set from already existing iPad application, but as final product was developed application with its unique functionality, tasks sets and interface. In thesis document were described main phases of application development, installation and validation. For the last on have been performed geometry validation, to use other ways to validate created application were not found enough resources.

In data analysis part were described the way how data for future analysis was collected and sorted. Also were described used technologies for recorded data analyses. Were described 6 motion mass parameters and statistical methods. In this thesis were performed calculations of 6 parameters and graphs were presented. Also, as statistical methods were used t-test for one sample and Welch's test. As result, was concluded that it is possible to observe differences in behavior of different subjects in fine motor drawing tasks performing and values of statistical testing have been shown, that changes between days are not similar for majority of MM parameters.

Generally, the main objectives of this work were achieved. Developed application have its own features and advantages in comparison with iPad application. In process of work it showed the ability to record, save data in correct way, perform different tasks and tasks sets representation. Performed analysis have shown differences and changes in subject's behavior and adaptation for fine motor drawing task, based on the recorded data for four days.

References

[1] S. Nõmm, K. Bardõš, A. Toomela, K. Medijainen, P. Taba (2018). Detailed analysis of the Luria's alternating series tests for Parkinson's disease diagnostics. Proceedings 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018 : 17-20 December 2018, Orlando, Florida, US.Ed. Wani, M. A., [et al.]. Piscataway, NJ: IEEE, 1347−1352.10.1109/ICMLA.2018.00219.

[2] S. Nõmm, K. Bardõš, I. Mašarov, J. Kozhenkina, A. Toomela, T. Toomsoo (2016). Recognition and analysis of the contours drawn during the Poppelreuter's test. In: 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA : 18-20 December 2016, Anaheim, California, Proceedings (170−175). Los Alamitos, Calif: IEEE Computer Society./10.1109/ICMLA.2016.0036.

[3] R. Sells and A. J. Larner, "The Poppelreuter figure visual perceptual function test for dementia diagnosis" [Online]. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pnp.193. [Accessed 27 05 2019]

[4] "Pointer events" [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Pointer_events. [Accessed 27 05 2019].

[5] "Electron Documentation" [Online]. Available: https://electronjs.org/docs. [Accessed 27 05 2019].

[6] "What's a Universal Windows Platform (UWP) app?" [Online]. https://docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide. [Accessed 27 05 2019]

[7] "PointerPointProperties Class" [Online]. Available: https://docs.microsoft.com/en-us/uwp/api/windows.ui.input.pointerpointproperties. [Accessed 27 05 2019]

[8] "Pen Input, Ink, and Recognition" [Online]. Available: https://docs.microsoft.com/en-us/windows/desktop/tablet/pen-input--ink--and-recognition. [Accessed 27 05 2019]

[9] "Codes for the Representation of Names of Languages" [Online]. Available: https://www.loc.gov/standards/iso639-2/php/code_list.php. [Accessed 27 05 2019]

[10] "math — Mathematical functions" [Online]. Available: https://docs.python.org/3/library/math.html. [Accessed 27 05 2019]

[11] "NumPy" [Online]. Available: https://www.numpy.org/. [Accessed 27 05 2019]

[12] "SciPy" [Online]. Available: https://docs.scipy.org/doc/scipy/reference/ [Accessed 27 05 2019]

[13] "matplotlib" [Online]. Available: https://matplotlib.org/contents.html. [Accessed 27 05 2019]

[14] "One Sample T-Test" [Online]. Available: https://www.statisticssolutions.com/manova-analysis-one-sample-t-test/. [Accessed 27 05 2019]

[15] "Welch's Test for Unequal Variances" [Online]. Available: https://www.statisticshowto.datasciencecentral.com/welchs-test-for-unequal-variances/. [Accessed 27 05 2019]

[16] "ApplicationDataContainer Class" [Online]. Available: https://docs.microsoft.com/en-us/uwp/api/windows.storage.applicationdatacontainer. [Accessed 27 05 2019]

Appendix 1 – Source code

UWP application’s source code:

https://gitlab.cs.ttu.ee/sezare/baka

Python source code:

https://gitlab.cs.ttu.ee/sezare/iapbpython