

Peltihamsteri-Sovellusprojekti

Järjestelmätestaussuunnitelma

**Mari Kasanen
Leevi Liimatainen
Marina Mustonen
Juhani Sundell
Arttu Ylä-Sahra**

Versio 0.2.0
Julkinen
11. huhtikuuta 2019

Jyväskylän yliopisto
Informaatioteknologian tiedekunta
Jyväskylä

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.2019		
Tilaja	__.__.2019		
Ohjaaja	__.__.2019		

Tietoa dokumentista

Tekijät:

- Mari Kasanen (MK) `r.mari.s.kasanen@student.jyu.fi`
- Leevi Liimatainen (LL) `leevi.m.m.liimatainen@student.jyu.fi`
- Marina Mustonen (MM) `marina.s.mustonen@student.jyu.fi`
- Juhani Sundell (JS) `juhani.k.sundell@student.jyu.fi`
- Arttu Ylä-Sahra (AY) `arttu.e.yla-sahra@student.jyu.fi`

Dokumentin nimi: Peltihamsteri-projekti, Järjestelmätestaussuunnitelma

Sivumäärä: 13

Tiivistelmä: Dokumentti on suunnitelma Peltihamsteri-projektin kehittämän ohjelmiston kokonaisvaltaista järjestelmätestaamista varten. Dokumentissa on määritelty testauskäytännöt, testausympäristö, sekä lähtötilanteet ja oletukset eri testijoukkoja varten (esim. miten testata ajosimulaattoria siten, että kaikkea testattavia datatyyppisiä muodostuu tallennettuun tiedostoon)

Avainsanat: sovellusprojekti, järjestelmätestaaminen, suunnitelma, esimerkkitoiminnot

Muutoshistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.0.1	6.3.2019	Kirjoitettu ensimmäinen luonnos järjestelmätestaussuunnitelmasta ja tehty muuta alustavaa dokumentointia.	AY
0.0.2	7.3.2019	Täydennetty ajosimulaattorin testausvaatimuksia ja dokumenttipohjaa.	AY
0.0.3	8.3.2019	Täydennetty verkkotesti- ja silmänliikekameran testausvaatimuksia.	JS, AY
0.0.4	11.3.2019	Muutettu pohja yhdenmukaisemmaksi muiden dokumenttien kanssa.	AY
0.0.5	12.3.2019	Luotu testikaavakepohja ja täydennetty Android-testilaitteen vaatimuksia.	LL, AY
0.0.6	21.3.2019	Testitapaukset muokattu ja ryhmitelty uudelleen vaatimusmäärittelyn muutosten mukaiseksi (vaatimusmäärittelyn versio 0.2.0), sivunumerointi luotu testikaavakkeeseen, versionumerointi korjattu kuvaavammaksi ja tehty erinäisiä kieliasukorjauksia.	MK, LL, MM, JS, AY
0.1.0	25.3.2019	Tekstin kirjoitusasu hiottu hieman, joitakin järjestelmätestaussuunnitelman epätasällisyyksiä poistettu ja testitapauksia uudelleenmuotoiltu hieman täsmällisemmäksi.	AY
0.1.1	8.4.2019	Aloitettu suunnitelmaan päivitystä palautteen mukaisesti; kieliäsuu korjattu selkeämmäksi, johdantoa päivitetty nykytilan mukaiseksi ja jätetty vaatimusmäärittelynumerointi pois dokumentin selkeyttämiseksi.	AY
0.1.2	9.4.2019	Siirretty esimerkkitoiminnot osaksi testitapausten määrittelyä, uudelleenmuotoiltu sekä suunnitelmaa että raporttipohjaa selkeämmäksi, tarkennettu määrittelmiä lisää ja päivitetty osittain testit vaatimusmäärittelyn versioon 0.3.1	AY
0.1.3	10.4.2019	Tarkennettu käyttöliittymän testejä ja muokattu ulkoasua.	AY
0.2.0	11.4.2019	Yhdenmukaistettu ylätunnisteen muotoilua muita dokumentteja vastaavaksi, poistettu dokumentin tiedostonimi mallipohjasta, tehty sekalaisia tarkentavia korjauksia kieliäsuun ja testitapausten määrittelyyn	AY

Tietoa projektista

Peltihamsteri-sovellusprojekti kehittää sovelluksen Jyväskylän yliopiston kognitiotieteen ajosimulaatiolaboratoriossa kerättävän datan synkronointiin ja hallintaan.

Tekijät:

- Mari Kasanen (MK) `r.mari.s.kasanen@student.jyu.fi`
- Leevi Liimatainen (LL) `leevi.m.m.liimatainen@student.jyu.fi`
- Marina Mustonen (MM) `marina.s.mustonen@student.jyu.fi`
- Juhani Sundell (JS) `juhani.k.sundell@student.jyu.fi`
- Arttu Ylä-Sahra (AY) `arttu.e.yla-sahra@student.jyu.fi`

Tilaaja:

- Hilikka Grahn `hilikka.grahn@jyu.fi` 040-8053342
- Tuomo Kujala `tuomo.kujala@jyu.fi` 0400-247392

Ohjaajat:

- Jonne Itkonen `jonne.itkonen@jyu.fi` 050-4432381
- Jukka-Pekka Santanen `santanen@jyu.fi` 040-8053299

Yhteystiedot:

- Sähköpostilistat: `peltihamsteri@korppi.jyu.fi`,
`peltihamsteri_opetus@korppi.jyu.fi`
- Sähköpostiarkistot: `http://korppi.jyu.fi/kotka/servlet/list-archive/peltihamsteri`,
`http://korppi.jyu.fi/kotka/servlet/list-archive/peltihamsteri_opetus`
- Työhuone: AgC 226.4 / 040-3573826

Sisältö

1	Johdanto	1
2	Testauskäytänteet- ja ympäristö	2
2.1	Johdanto	2
2.2	Testausympäristö	2
2.3	Testikertojen kirjaaminen	3
3	Sovelluksen perustoiminnallisuus	5
4	Verkkoyhteystestit	8
5	Ajosimulaattori	9
6	Silmänliikekamera	10
7	Toissijaisen tehtävän Android-laite	11
8	EEG-laite	12
9	Lähteet	13

1 Johdanto

Peltihamsteri-projekti kehittää kevään 2019 Sovellusprojekti-kurssilla sovelluksen Jyväskylän yliopiston kognitiotieteen ajosimulaatiolaboratoriolle datan synkronoimiseen ja yhdistämiseen. Sovelluksen avulla synkronoidaan ajosimulaatiokokeesta eri laitteista saatava data ja tulostetaan se tiedostoon, jossa dataa on helppo käsitellä. [1].

Sovellus kehitetään pääsääntöisesti käyttäen C#-ohjelmointikieltä, mutta sisältää myös muilla kielillä kirjoitettuja osia. Esimerkiksi ajosimulaattorin tarvitsema apuskripti on kirjoitettu Lualla.

Testaussuunnitelma kuvaa sovelluksen järjestelmätestaukseen sovellettavat testausmenetelmät (luvussa 2) ja testauskerroilla suoritettavat testitapaukset (luvut 3-8). Testitapaukset on suunniteltu projektin vaatimusmäärittelyyn [2] perustuen, ja järjestetty oletetun työnkulun mukaan.

Testaussuunnitelman laatimisen apuna on hyödynnetty Isäxi-projektin järjestelmätestaussuunnitelmia [4], sekä Monisiro-projektin järjestelmätestaussuunnitelmia [5]. Lisäksi on hyödynnetty sovellusprojektien L^AT_EX-dokumenttipohjaa [6].

2 Testauskäytännöt- ja ympäristö

2.1 Johdanto

Tässä luvussa esitellään tarkemmin testauskäytännöt ja -ympäristöä.

Järjestelmätestaus tehdään, jotta voitaisiin löytää mahdollisia virheitä ja puutteita ohjelman toiminnasta oikeaa käyttöä mallintavassa tilanteessa. Järjestelmätestaus toteutetaan toiminnallisena mustalaatikkotestauksena testauksena integroidulle kokonaiselle testiympäristölle.

Eri testijoukoille on määritelty oletuksia, jotka ovat testitapausten kannalta olennaisia. Osa oletuksista on esitetty toimintaohjeiden muodossa, joilla tarkoitus on mallintaa soveltuvien osien oikeaa koetilannetta, ja esim. tuottaa sellaista dataa, jota tarvitaan testitapausten onnistumisen todentamiseen.

2.2 Testausympäristö

Testauskerta tulee suorittaa Windows 10-järjestelmällä, joka täyttää seuraavat ehdot:

- Järjestelmä sisältää ajantasaisen, mahdollisimman uuden .NET-ympäristön.
- Järjestelmän kellon tulee olla asetettu oikeaan paikalliseen aikaan, mahdollisuuksien mukaan synkronoituna NTP-palvelimeen.
- Järjestelmässä ei saa olla käynnissä ylimääräisiä ohjelmia testitapausten suorittamisen aikana, vaan vain välttämättömät ohjelmat saavat olla samaan aikaan käynnissä. Tällaisia ohjelmia ovat esim. mahdolliset muiden laitteiden vaatimat apuohjelmat. Tällä rajoitteella vähennetään epätoivottujen sivuvaikutusten mahdollisuutta esim. verkkoliikenteen tai levy-I/O:n hidastelusta johtuen.

Osaan testitapauksista voi littyä erityisiä oletuksia lähtötilanteen kannalta (esim. ajosimulaattorin valittu kartta, silmänliikekameran ohjelmistoasetukset). Nämä oletukset on esitelty tarkemmin testijoukkojen määrittelyn yhteydessä.

2.3 Testikertojen kirjaaminen

Jokaisesta testauskerrasta laaditaan testausraportti, käyttäen valmista mallipohjaa [3]. Mallipohjaan kirjataan ne testijoukot joita käytiin läpi testauskerran aikana, eri testitapausten tulokset, sekä mahdollisia huomioita mikäli testitapausten käyttäytyminen poikkeaa olennaisesti tässä dokumentissa määritellystä käyttäytymisestä.

Jokainen yksittäinen testitapausta on merkitty koodilla (esim. 1.4c). Ensimmäinen luku kuvaa testitapausta, toinen tarkempaa tapausta, ja kirjain mahdollisesti keskenään riippuvaisten ja/tai samankaltaisten testitapausten eroa. Yksittäisen testitapausten tulos on jokin seuraavista:

OK	Testitapausta läpäisty ilman havaittuja ongelmia tai poikkeamia
Virhe	Testitapausten aikana havaittiin merkittäviä poikkeamia tai muita ongelmia. Raporttiin merkitään kuvaus poikkeavasta toiminnasta tai ilmenneistä ongelmista
Puute	Testitapausta läpäisty, mutta raporttiin merkityllä tavalla puutteellisenä
Huomautus	Testitapausta ei läpäisty eikä hylätty. Raporttiin merkitään kuvaus siitä, miksi testitapaukselle ei voida merkitä muuta tulosta
Ohitettu	Testitapausta ohitettiin raporttiin merkitystä syystä

Lisäksi jokaisesta testauskerrasta kirjataan lyhyt yhteenveto ja johtopäätös. Johtopäätös on aina jokin seuraavista:

Hyväksytty	Testauskertaa on olennaisesti onnistunut, eikä mitään merkittäviä puutteita tai ongelmia havaittu testauskerran aikana
Hyväksytty huomautuksin	Testauskertaa on olennaisesti onnistunut, mutta testien valintaan tai suoritukseen liittyy merkittäviä huomioita (esim. poikettu merkittävästi oletuksista tai toimintaohjeista). Nämä huomiot on kirjattava raporttiin joko yhteenvetoon tai huomioita koskevan testijoukon yhteyteen.

Korjattavaa

Testauskerran aikana havittiin merkittäviä puutteita tai ongelmia, siinä määrin ettei testauskertaa voi ajatella pääosin onnistuneena

3 Sovelluksen perustoiminnallisuus

Testijoukon yleiset vaatimukset: Testijoukon lähtötilanteen ja suorituksen pitää täyttää seuraavat vaatimukset

- Testaa vähintään näitä laiteyhdistelmiä:
 - Tyhjä (ei laitteita valittu)
 - Yksi vapaavalintainen laite
 - Kaksi vapaavalintaista eri laitetta (esim. ajosimulaattori ja silmänliikekamera)
- Testaa vähintään näitä dataosioyhdistelmiä:
 - Tyhjä (ei dataosioita valittu)
 - Ei aikaleimoja (valittu vapaavalintaiset 1-N osiota, mutta siten ettei mikään niistä ei ole aikaleima)
 - Aikaleimoilla (valittu vapaavalintaiset 1-N osiota, ja kaikki aikaleimat, jos niitä on mahdollista valita)
- Kokeen tai koehenkilön tunniste, tehtävän nimi tai kommentti eivät saa sisältää tiedostonimiin kelpaamattomia merkkejä (esim. polkuerotin). Virheellisten merkkijonojen käytön pitää johtaa virheilmoitukseen ennen tallennuksen aloittamista. Ongelmallisesti piirtyviin Unicode-merkkijonoihin (esim. RTL-vaihdokset kesken tekstin, liitosmerkkien liiallinen käyttö) ei oteta kantaa.

Testitapaukset

1. Laite- ja dataosioasetukset

- 1.1a** Kokeessa käytettävät laitteet voidaan valita.
- 1.1b** Käytettävät laitteet voidaan valita graafisesta käyttöliittymästä.
- 1.2** Sovellus ei aseta rajoitteita laitevalinnoille, vaan laitteet ovat toisistaan riippumattomasti valittavissa.
- 1.3a** Käytettävistä laitteista voidaan valita ne dataosiot, jotka halutaan tallentaa tulostiedostoon.
- 1.3b** Käytetyt dataosiot voidaan valita graafisesta käyttöliittymästä.
- 1.4a** Laitteille voidaan asettaa laitekohtaisia asetuksia, ja sovellus varoittaa asetusten ongelmista tai estää virheellisten asetusten käyttämisen.

- 1.4b** Laiteasetukset ja valitut dataosiot on mahdollista tuoda erillisestä tiedostosta.
- 1.4c** Laiteasetukset ja valitut dataosiot voidaan tuoda erillisestä tiedostosta graafisen käyttöliittymän kautta.
- 1.4d** Käyttöliittymässä valitut laiteasetukset ja dataosiot voidaan viedä erilliseen asetustiedostoon.
- 1.4e** Asetustiedosto sisältää tiedot kokeen tunnisteesta, kokeen tehtävistä, tallennuksessa käytettävistä laitteista ja niiltä halutuista dataosioista, laitteisiin yhdistämiseen käytettävistä IP-osoitteista ja porteista, sekä tiedostopolun johon tallennuksen tiedostot tallennetaan.
- 1.4f** Asetustiedosto on XML-muotoinen ja luettavissa tavallisella tekstieditorilla.

2. Koetietojen ja tallennuksen hallinta

- 2.1** Kokeen tunniste voidaan syöttää graafiseen käyttöliittymään.
- 2.2** Koehenkilön tunniste voidaan syöttää graafiseen käyttöliittymään.
- 2.3** Tehtävän tunniste voidaan syöttää graafiseen käyttöliittymään.
- 2.4a** Tallennus voidaan käynnistää graafisesta käyttöliittymästä.
- 2.4b** Tallennus voidaan pysäyttää graafisesta käyttöliittymästä.
- 2.5a** Tallennusta ei voida aloittaa jollei vähintään yhtä laitetta ja vähintään yhtä dataosiota ole valittuna.
- 2.5b** Tallennusta ei voida aloittaa, jos kokeen, koehenkilön ja tehtävän tunnisteita ei ole syötetty.
- 2.6** Graafinen käyttöliittymä antaa palautetta siitä, saapuuko laitteisiin dataa.

3. Tallennettujen tietojen käsittely

- 3.1a** Tulostiedoston muodostaminen voidaan tehdä graafisesta käyttöliittymästä.
- 3.1b** Laitteiden antamat tiedot voidaan synkronoida ja yhdistää graafisen käyttöliittymän kautta.
- 3.2a** Laitteiden antamien tietojen yhteyteen voidaan liittää sovelluksen aiemmin muodostamia tallennustiedostoja (poislukien lopullinen tulostiedosto) käyttöliittymän kautta.

- 3.2b Laitteiden antamien tietojen yhteyteen voidaan liittää laitteiden omien sovellusten muodostamia tiedostoja, mikäli niiden muoto on CSV-standardin mukainen ja tiedostot sisältävät tarpeelliset aikatiedot synkronointia varten.
- 3.2c Käyttäjälle näytetään virheilmoitus, jos sovellukselle syötetty tiedosto ei sovellu sovelluksen käyttöön, tai tiedoston sisältämää dataa ei voida synkronoida muuhun dataan.
- 3.3 Sovellus kirjaa ylös tallennuksen aloitusajan myöhempää käyttöä varten.

4. Tulostiedoston ominaisuudet

- 4.1 Tulostiedoston tiedostonimi sisältää tiedot kokeesta, koehenkilöstä, tehtävästä ja tehtävän suoritusajasta aikaleiman (esim. UNIX-aika) muodossa.
- 4.2 Tulostiedosto on CSV-muotoinen, noudattaen RFC4180-standardin määritelmää.
- 4.3 Tulostiedoston ensimmäinen rivi nimeää eri sarakkeet.
- 4.4 Tulostiedosto sisältää yhden koehenkilön yhden tehtävän valittujen laitteiden valitut dataosiot.
- 4.5 Tulostiedoston sarakkeita ovat sovelluksen yhtenäisen aikaleiman sarakkeet (UNIX-aika, kehysnumero ja alikehysnumero), ja valitut dataosiot.
- 4.6 Tulostiedoston jokainen rivi sisältää sovelluksen aikaleiman.

5. Graafisen käyttöliittymän muut ominaisuudet

- 5.1 Graafinen käyttöliittymä on englanninkielinen (poislukien esim. käyttöjärjestelmän mahdollisesti lokalisoimat virheviestit).
- 5.2 Sovelluksessa on käyttöohje, jonka voi avata graafisesta käyttöliittymästä.

4 Verkkoyhteystestit

Testijoukon yleiset vaatimukset: Testijoukon lähtötilanteen ja suorituksen pitää täyttää seuraavat vaatimukset

- Yhden testilaitteen on oltava yhdistetty muuten kuin suoraan samaan tietokoneeseen missä sovellusta käytetään, esimerkiksi lähiverkon yli
- Etälaitteen kello tulee synkronoida samaan aikaan kuin testilaitteen kello, tai aikaerotus on muuten kirjattava ylös ja huomioitava testitapauksissa
- Yhteyden on oltava luotettava (pakettien häviöprosentti alle 7,5%) ja vähäviiveinen (viive alle 80ms)
- Verkkoyhteyden katkeamista ja viivettä on mallinnettava seuraavasti:
 - Katkaise yhteys irrottamalla verkkokaapeli tai muulla soveltuvalla tavalla. Mittaa aikaa noin 20 sekuntia, ja palauta verkkoyhteys. Kirjaa ylös katkaisu- ja palautusajat.
 - Tarkista ettei tulostiedoston datassa ilmene muita poikkeavuuksia kuin datan mahdollinen puuttuminen yhteyden katkaisuajalta. Aikaleimojen on säilyttävä johdonmukaisina myös katkonaisina.
- Hyväksyttäviä portteja ovat portit väliltä 1-65535. Ei-numeeristen tai alueen ulkopuolisten porttien pitää johtaa virheilmoitukseen.
- Hyväksyttävät IP-osoitteet ovat joko IPv4 (esim. 127.0.0.1)- tai IPv6-muotoisia (esim. 0:0:0:0:0:0:0:1, ::1), standardien mukaisesti esitettyjä osoitteita. Muuntyyppisten osoitteiden käytön on johdettava virheilmoitukseen.

Testitapaukset

6. Verkkotoiminnallisuuden luotettavuus

- 6.1** Sovellus varoittaa verkkoyhteyden katkeamisesta ja yrittää yhdistää uudelleen mikäli tarpeen.
- 6.2** Sovelluksesta on mahdollista säätää eri laitteiden osoite- ja porttiasetuksia graafisella käyttöliittymällä.
- 6.3** Sovellus kykenee käsittelemään hallitusti tilanteita joissa data saapuu sovellukselle viiveellä tai muussa kuin aikajärjestyksessä.

5 Ajosimulaattori

Testijoukon yleiset vaatimukset: Testijoukon lähtötilanteen ja suorituksen pitää täyttää seuraavat vaatimukset

- Eepsoft-ajosimulaattori pitää olla asennettuna ja saatavilla, joko samalla koneella kuin missä testitapaukset suoritetaan, tai erillisellä lähiverkon kautta yhteydessä olevalla laitteella
- Ajosimulaattoriin tulee olla asennettu `peltihamsteri_udp.lua` -skripti, ja kyseinen skripti pitää ladata simulaattoria käynnistettäessä. Tarkista, että skriptissä olevat IP-osoite ja portti osoittavat oikeaan paikkaan
- Testitapausten todentamiseen soveltuvan datan keräämiseksi on ajosimulaattoria käytettävä seuraavasti:
 1. Valitse ajosimulaattorista kartaksi `city`, ja suoritettavaksi tehtäväksi `User->Here->HP->Occlusion practice`. Tämä käynnistää okklusioharjoituksen, jossa ajonäkymä pimenee ellei valittua painiketta (`Extral`) paineta säännöllisesti.
 2. Aja kenttää jonkin matkaa. Käytä rattiohjaimen tai näppäimistön painiketta, joka kytkee pimennyksen hetkeksi pois päältä. Kokeile halutessasi ajaa käsivaihteella, ja kokeile sekä vilkkuja että käsijarrua
 3. Lopetettuasi, paina End of Exercise ajosimulaattorin taukovalikosta.

Testitapaukset

7. Ajosimulaattoridatan kaappaus

- 7.1 Ohjeiden mukaisella toiminnalla tuotetut tiedot toistuvat datassa oikein: aikaleimat, X,Y,Z, kaasun, jarrun, ohjauspyörän asento, nopeus (km/h, m/s, ja kierrosnopeus), rengasosumat ja rengas pois tieltä ja etäisyys kaistan keskipisteeseen (vain `city`-kartassa).
- 7.2 FADE- ja EOE (End of Exercise)-merkinnät toistuvat oikeissa kohdissa datassa, huomioiden suhteellisen aikaleiman.

6 Silmänliikekamera

Testijoukon yleiset vaatimukset: Testijoukon lähtötilanteen ja suorituksen pitää täyttää seuraavat vaatimukset

- D-Lab-ohjelmiston (v3.52) on oltava asennettuna, ja silmänliikekamera kytkettynä tietokoneeseen
- D-Labin Relay-toiminnon on oltava käytössä, ja asetetun porttinumeron on vastattava ohjelmassa asetettua porttinumeroa
- On varmistuttava, että sovellus lähettää aikaleimaa sekunteina, eikä kehysten järjestyslukuina. Kehysten järjestyslukuja ei voi käyttää datan synkronointiin luotettavasti.
- AOI:n toiminnallisuutta on kokeiltava seuraavasti:
 - Ohjelmoi muutama eri Area of Interest (AOI) D-Labissa, ja kerää tästä silmänliikedataa. Siirrä katsettasi eri AOI:den välillä. Näkyvätkö nämä AOI:t D-Labin tiedostossa ja Peltihamsteri-sovelluksen tiedostossa samoilla ajoilla?
 - Kokeile sulkea toinen tai molemmat silmät. Vastaavien koordinaanttien pitäisi muuttua nollassi silmänliikeditassa

Testitapaukset

8. Silmänliikeditan kaappaus

- 8.1 Ohjeiden mukaisella toiminnalla tuotetut tiedot toistuvat datassa oikein: aikaleima, silmien omat ja yhteiset X- ja Y-koordinaatit, pupillien pinta-ala, leveys ja korkeus, sakkadit (nopea silmänliike), sakkadin kesto, sakkadin kulma, fiksaatioiden määrä ja kesto, katseen X ja Y kohteeseen suhteutettuna, katseen tämänhetkinen AOI.
- 8.2 Sovellus kykenee tulkitsemaan vaihtuvia dataformaatteja, ja varoittaa käyttäjää mikäli dataformaattia ei pystytä tulkitsemaan tai synkronoimaan riittävän tarkasti.

7 Toissijaisen tehtävän Android-laite

Testijoukon yleiset vaatimukset: Testijoukon lähtötilanteen ja suorituksen pitää täyttää seuraavat vaatimukset

- Kokeile erilaisia sormennäpätysrytmejä, sekä nopeita että hitaita. Näiden on toistuttava oikein aikaleimoissa; tämän voi todentaa esim. Excel-graafilla

Testitapaukset

9. Kosketusdatan kaappaus

- 9.1** Ohjeiden mukaisella toiminnalla tuotetut tiedot toistuvat datassa oikein: kosketuksen absoluuttinen ja suhteellinen aikaleima laitteen kellon mukaan, X- ja Y-sijainti näytöllä.

8 EEG-laite

Testijoukon yleiset vaatimukset: Testijoukon lähtötilanteen ja suorituksen pitää täyttää seuraavat vaatimukset

- DSI-Streamerin (v1.08.28) on oltava käynnissä, ja kypärän kytkettynä tietokoneeseen Bluetoothilla.
- Kokeile liikuttaa esim. leukaa datan kaappauksen aikana. Leuan liikuttaminen tuottaa runsaasti vaihtelua kaapattuun dataan, jonka pitäisi toistua selkeästi tulostiedostossa esim. Excelillä visualisoituna.

Testitapaukset

10. Aivosähkökäyrädatan kaappaus

- 10.1** Ohjeiden mukaisella toiminnalla tuotetut tiedot toistuvat datassa oikein: kaikki kypärän antamat kanavat (1-24, trigger) ja aikaleimat.

9 Lähteet

- [1] Marina Mustonen, "Peltihamsteri-projekti, Projektisuunnitelma", Jyväskylän yliopisto, Informaatioteknologian tiedekunta, 6.3.2019
- [2] Mari Kasanen, Leevi Liimatainen, Marina Mustonen, Juhani Sundell ja Arttu Ylä-Sahra, "Peltihamsteri-projekti, Vaatimusmäärittely", Jyväskylän yliopisto, Informaatioteknologian tiedekunta, 6.3.2019
- [3] Mari Kasanen, Leevi Liimatainen, Marina Mustonen, Juhani Sundell ja Arttu Ylä-Sahra, "Peltihamsteri-projekti, Järjestelmätestausraporttipohja", Jyväskylän yliopisto, Informaatioteknologian tiedekunta, 11.4.2019
- [4] Sauli Flinkman, Jere Juntila, Tuukka Jurvakainen ja Anette Karhu, "Isäxi-projekti, Järjestelmätestaussuunnitelma", Jyväskylän yliopisto, Informaatioteknologian tiedekunta, 24.5.2018
- [5] Severi Jääskeläinen, Samuel Kaiponen, Heta Rekilä ja Sinikka Siironen, "Monisiro-projekti, System Testing Plan", Jyväskylän yliopisto, Informaatioteknologian tiedekunta, 1.6.2018
- [6] Antti-Juhani Kaijanaho, Ville Isomöttönen ja Matthieu Weber, "Sovellusprojektien dokumenttipohjamalli", haettu 21.3.2019 osoitteesta <http://www.mit.jyu.fi/palvelut/sovellusprojektit/materiaalit/projdoc.cls>, Jyväskylän yliopisto, Informaatioteknologian tiedekunta, 16.9.2006