

Peltihamsteri-sovellusprojekti

Järjestelmätestaussuunnitelma

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

Versio 1.0.0
Julkinen
14. toukokuuta 2019

Jyväskylän yliopisto
Informaatioteknologian tiedekunta
Jyväskylä

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.2019		
Tilaaaja	__.__.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: 21

Tiivistelmä: Dokumentti on suunnitelma Peltihamsteri-projektin kehittämän ohjelmiston kattavaa järjestelmätestausta varten. Dokumentissa on määritelty testauskäytännöt, testausympäristö, täsmälliset raportointivaatimukset ja -ohjeet, sekä lähtötilanteet ja oletukset eri testijoukkoja varten.

Avainsanat: sovellusprojekti, järjestelmätestaus, suunnitelma, testauskäytännöt, raporttipohja, testausohje, lähtötilanteet, oletukset, raportointiohjeet

Muutoshistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.0.1	6.3.2019	Kirjoitettiin ensimmäinen luonnos järjestelmätestaussuunnitelmasta ja tehtiin muuta alustavaa dokumentointia.	AY
0.0.2	7.3.2019	Täydennettiin ajosimulaattorin testausvaatimuksia ja dokumenttipohjaa.	AY
0.0.3	8.3.2019	Täydennettiin verkkotesti- ja silmänliikekameran testausvaatimuksia.	JS, AY
0.0.4	11.3.2019	Muutettiin pohja yhdenmukaisemmaksi muiden dokumenttien kanssa.	AY
0.0.5	12.3.2019	Luotiin testikaavakepohja ja täydennettiin Android-testilaitteen vaatimuksia.	LL, AY
0.0.6	21.3.2019	Testitapaukset muokattiin ja ryhmiteltiin uudelleen vaatimusmäärittelyn muutosten mukaiseksi (vaatimusmäärittelyn versio 0.2.0), testikaavakkeeseen lisätty sivunumerointi, versionumerointi korjattiin kuvaavammaksi. Lisäksi tehtiin erinäisiä kieliasukorjauksia.	MK, LL, MM, JS, AY
0.1.0	25.3.2019	Tekstin kirjoitusasua hiottiin hieman, joitakin järjestelmätestaussuunnitelman epätasällisyyksiä poistettiin ja testitapauksia uudelleenmuotoiltiin hieman tasällisemmäksi.	AY
0.1.1	8.4.2019	Aloitettiin suunnitelmaan päivitystä palautteen mukaisesti; kieliasua korjattiin selkeämmäksi, johdantoa päivitettiin nykytilan mukaiseksi ja poistettiin testitapausten vaatimusmäärittelynumerointi dokumentin selkeyttämiseksi.	AY
0.1.2	9.4.2019	Siirrettiin esimerkkitoiminnot osaksi testitapausten määrittelyä, uudelleenmuotoiltiin sekä suunnitelmaa että raporttipohjaa selkeämmäksi, tarkennettiin määritelmiä lisää ja päivitettiin osittain testit vaatimusmäärittelyn versioon 0.3.1.	AY
0.1.3	10.4.2019	Tarkennettiin käyttöliittymän testejä ja muokattiin ulkoasua.	AY

Versio	Päivämäärä	Muutokset	Tekijät
0.2.0	11.4.2019	Yhdenmukaistettiin ylätunnisteen muotoilua muita dokumentteja vastaavaksi, poistettiin dokumentin tiedostonimi mallipohjasta tarpeettomana, tehtiin sekalaisia tarkentavia korjauksia kieliasuun ja testitapausten määrittelyyn.	AY
0.2.1	18.4.2019	Muotoiltiin uudelleen palautteen perusteella rakennetta ja raportointiohjetta. Lisäksi raportointivaatimuksia on tarkennettu palautteen mukaisesti yksiselitteisemmäksi ja paremmin toistettavaksi. Testitapauksia on myös alustavasti yksinkertaistettu paremmin nykyistä sovelluksen tilaa ja suunnitelmaa vastaavaksi.	AY
0.2.2	24.4.2019	Tarkennettiin testitapauksia tilaajan palautteen perusteella, lisättiin sovelluksen päätetty nimi.	JS, AY
0.2.3	25.4.2019	Lisättiin liitemerkinnät raporttipohjaan, lisäksi tehtiin kirjoitusasun tarkennusta.	AY
0.3.0	26.4.2019	Suoritettiin ensimmäinen järjestelmätestauskerta, korotettiin versionumero.	AY
0.3.1	6.5.2019	Tehtiin muutoksia version 0.3.0 palautteen pohjalta kielioppiin ja rakenteeseen. Lisättiin järjestelmätestausraporttipohjaan otsikkosivu ja raportin yksilöiville tiedoille paikka. Lisäksi uudelleenmuotoiltiin muutamia testejä yksiselitteisempään muotoon.	AY
0.4.0	8.5.2019	Korotettiin versionumero, muutettiin Androidin testitapauksia nykytilan mukaiseksi, ja pilkottiin testitapausta 2.5 paremmin vaatimusmäärittelyn mukaiseksi.	MK, LL, AY
0.4.1	10.5.2019	Muotoiltiin testitapauksia uudelleen vaatimusmäärittelyn läpikäynnin yhteydessä.	MM, MK, LL, JS, AY
0.4.2	13.5.2019	Korjattiin virheellinen sivukoko raporttipohjassa, tehtiin pieniä kieliasuparannuksia ja korjattiin virheellinen ajosimulaattorin testikentän nimi.	AY
1.0.0	14.5.2019	Korotettiin versionumero ensimmäiseksi hyväksytyksi versioksi, ja muutettiin otsikkosivun kirjoitusasua.	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	Testauskerran tavoitteet ja oletukset	2
2.2	Testausympäristö	3
3	Testauskertojen raportointi	4
3.1	Yleiset raportointiohjeet	4
3.2	Testitapausten tilat	5
3.3	Testauskerran yhteenveto	6
4	Testijoukot	7
4.1	Sovelluksen perustoiminnallisuus	7
4.1.1	Vaatimukset ja suoritusohjeet	7
4.1.2	Testitapaukset	8
4.2	Tulostiedostojen ominaisuudet	11
4.2.1	Vaatimukset ja suoritusohjeet	11
4.2.2	Testitapaukset	11
4.3	Verkkoyhteystestit	13
4.3.1	Vaatimukset ja suoritusohjeet	13
4.3.2	Testitapaukset	13
4.4	Ajosimulaattori	15
4.4.1	Vaatimukset ja suoritusohjeet	15
4.4.2	Testitapaukset	15
4.5	Silmänliikekamera	17
4.5.1	Vaatimukset ja suoritusohjeet	17
4.5.2	Testitapaukset	17
4.6	Toissijaisen tehtävän Android-laite	19
4.6.1	Vaatimukset ja suoritusohjeet	19
4.6.2	Testitapaukset	19
4.7	EEG-laite	20
4.7.1	Vaatimukset ja suoritusohjeet	20
4.7.2	Testitapaukset	20
5	Lähteet	21

1 Johdanto

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

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

Sovelluksen järjestelmätestaukseen sovellettavat testauskäytännöt kuvataan luvussa 2 ja testauskerroilla suoritettavat testitapaukset luvussa 4. Testitapaukset on suunniteltu projektin vaatimusmäärittelyyn [2] perustuen, ja järjestetty oletetun työnkulun mukaan.

Testaussuunnitelman laatimisessa on hyödynnetty Isäxi-projektin järjestelmätestaussuunnitelmaa [4] sekä Monisiro-projektin järjestelmätestaussuunnitelmaa [5]. Lisäksi on hyödynnetty sovellusprojektien \LaTeX -dokumenttipohjaa [6].

2 Testauskäytännöt ja -ympäristö

2.1 Testauskerran tavoitteet ja oletukset

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

Eri testijoukoille on määritelty oletuksia, jotka ovat testitapausten kannalta olennaisia. Osa oletuksista on esitetty toimintaohjeiden muodossa, joilla mallinnetaan soveltuvien osien oikeaa koetilannetta, ja muodostaa 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 testitapauksien 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, tai silmänliikekameran ohjelmistoasetukset). Nämä lähtötilat on esitelty tarkemmin testijoukkojen määrittelyn yhteydessä.

3 Testauskertojen raportointi

3.1 Yleiset raportointiohjeet

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

Yksittäiselle testijoukolle voi olla asetettu myös omia raportointivaatimuksia, jotka on mainittu kunkin testijoukon luvun alussa. Testijoukkoa suoritettaessa on huolehdittava että kaikki raportointivaatimusten mainitsemat tiedot kerätään talteen ja liitetään testausraportin yhteyteen. Kerätyt liitetiedot on myös luetteloitava testausraporttiin siihen erikseen varattuun kohtaan. **Huomaa, että yksittäinen järjestelmätestauskerta ei välttämättä ole toistettavissa, jos esim. valittuja asetuksia ei ole liitetty raportin yhteyteen!**

3.2 Testitapausten tilat

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

OK	Testitapaus suoritettu ilman havaittuja ongelmia tai poikkeamia.
Huomautus	Testitapaus suoritettu, mutta testitapausten aikana ilmeni jotain erityyisen huomion arvoista. Testitapausten suoritus ei kuitenkaan ole selkeästi puutteellista tai virheellistä.
Puute	Testitapausten suoritus on jollakin tavalla selkeästi puutteellista.
Virhe	Testitapausten aikana havaittiin merkittäviä poikkeamia tai muita ongelmia, siinä määrin ettei testitapausta voida pitää läpäistynä.
Ohitettu	Testitapausta ei suoritettu.

Kaikkiin muihin kuin *OK*-tilan yhteyteen on kirjattava peruste sille, miksi kyseisen tilan valinta on sopivin testitapaukselle.

3.3 Testauskerran yhteenveto

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

Hyväksytty	Testauskertan suoritettiin onnistuneesti, eikä mitään merkittäviä puutteita tai ongelmia havaittu testauskerran aikana.
Hyväksytty huomautuksin	Testauskerta on olennaisesti onnistunut, mutta testijoukkojen valintaan tai suoritukseen liittyy merkittäviä huomioita (esim. ohitettiin kokonainen testijoukko, tai poikettiin merkittävästi oletuksista tai toimintaohjeista). Nämä huomiot on kirjattava raporttiin joko yhteenvetoon tai huomioita koskevan testijoukon yhteyteen.
Korjattavaa	Testauskerran suorituksessa havittiin merkittäviä puutteita tai ongelmia, eikä testauskertaa voi ajatella pääosin onnistuneena.

4 Testijoukot

4.1 Sovelluksen perustoiminnallisuus

4.1.1 Vaatimukset ja suoritusohjeet

Testitapausten rajauksia ja ohjeita:

- 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.
- Vain Suomessa yleisesti käytetyt Unicode-merkit ovat käytössä testien yhteydessä; esimerkiksi erityistä renderöintitukea tarvitsevat, oikealta vasemmalle kirjoitettavat kielet tai äärimmäiset rajatapaukset (esim. Zalgo-tekstit) jätetään huomiotta.
- Testitapauksissa käytettävä CSV-muodon määritelmä on testijoukon *Tulostiedostojen ominaisuudet* yhteydessä luvussa 4.2.

Suoritusohjeet:

- Testaa virheellisiä (esim. polkuerottimen sisältävää tai täysin tyhjiä) tunnisteita. Kokeile myös täysin kelvollisia tunnisteita (vain yleisiä merkkejä).
- Testaa keskenään ristiriitaisia laiteasetuksia, esimerkiksi määrittämällä sama porttinumero usealle eri laitteelle.
- Testaa vähintään seuraavia laiteyhdistelmiä:
 - tyhjä (ei laitteita valittu),
 - yksi vapaavalintainen laite, sekä
 - kaksi vapaavalintaista eri laitetta (esim. ajosimulaattori ja silmänliikekamera).
- Toistaiseksi ei ole tiedossa keskenään yhteensopimattomia laitevalintoja, joten testitapaukselle 1.1b katsotaan riittäväksi, että edellisessä luettelossa valitut laitteet toimivat sekä yksinään että keskenään. Laiteasetusten ristiriidat testataan erikseen tapauksessa 1.3c.
- Testaa vähintään näitä dataosioyhdistelmiä:
 - tyhjä (ei dataosioita valittu),

- ei aikaleimoja (valittu vapaavalintaiset 1–N osiota, mutta siten ettei mikään niistä ei ole aikaleima), sekä
 - aikaleimoilla (valittu vapaavalintaiset 1–N osiota, ja kaikki aikaleimat jos niitä on mahdollista valita).
- Sovelluksen yhteydessä on erikseen dokumentoitu, mitä ominaisuuksia vaaditaan liitettäviltä ulkoisilta tiedostoilta. Kokeile liittää sekä vaatimukset täyttävää dataa että myös vaatimukset täyttämätöntä (esim. satunnaisgeneroitua) dataa

Liitä testausraporttiin liitteeksi

- kuvaus testatuista tunnisteista,
- kuvaus testatuista laiteyhdistelmistä,
- kuvaus testatuista dataosioyhdistelmistä,
- nauhoitukseen käytetyt asetukset asetustiedoston muodossa, sekä
- kuvaus asetuksista, joilla lisätiedostoja on tuotu, varsinaiset tiedostot, jotka tuotiin tai yritettiin tuoda, ja asetukset, joilla tulostiedostoja on viety. Tulostiedostot vaaditaan raportoitavaksi eri testijoukossa.

4.1.2 Testitapaukset

1. Laite- ja dataosioasetukset:

- 1.1a** Käytettävät laitteet voidaan valita graafisesta käyttöliittymästä.
- 1.1b** Sovellus ilmoittaa selkeästi mikäli laitevalinnat ovat keskenään yhteensopimattomia.
- 1.2** Käytetyt dataosiot voidaan valita graafisesta käyttöliittymästä.
- 1.3a** Laitteille voidaan asettaa laitekohtaisia asetuksia.
- 1.3b** Laiteasetukset ja valitut dataosiot voidaan lukea erillisestä tiedostosta graafisen käyttöliittymän kautta.
- 1.3c** Sovellus varoittaa määritettyjen asetusten ongelmista tai estää virheellisten asetusten käyttämisen.
- 1.3d** Käyttöliittymässä valitut laiteasetukset ja dataosiot voidaan tallentaa erilliseen asetustiedostoon.
- 1.3e** 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.

1.3f 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 ole valittuna.

2.5b Tallennusta ei voida aloittaa, jollei vähintään yhtä dataosiota ole valittuna.

2.5c Tallennusta ei voida aloittaa, jos kokeen, koehenkilön ja tehtävän tunnisteita ei ole syötetty.

2.6 Graafinen käyttöliittymä antaa palautetta sydämensykkeen muodossa 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 ne ovat CSV-muotoisia 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 metatietoihin.

4. Graafisen käyttöliittymän muut ominaisuudet:

- 4.1** Graafinen käyttöliittymä on englanninkielinen (poislukien esim. käyttöjärjestelmän mahdollisesti lokalisoimat virheviestit).
- 4.2** Sovelluksen käyttöohjeen voi avata graafisesta käyttöliittymästä.

4.2 Tulostiedostojen ominaisuudet

4.2.1 Vaatimukset ja suoritusohjeet

Testitapauksien ohjeita:

- CSV:n virallisesti määrittävä RFC4180-standardi löytyy osoitteesta: <https://tools.ietf.org/html/rfc4180>.
- Testitapauksissa CSV-muotoiseksi tulkitaan seuraavat standardista yksinkertaistetut edellytykset täyttävä data:
 - Tiedosto koostuu riveistä, joista jokaisella rivillä on sovitulla erotinmerkillä (yleisesti ,) eroteltuja sarakkeita.
 - Ensimmäinen rivi voi sisältää (mutta ei välttämättä sisällä) eri sarakkeiden nimet.
 - Sarakkeita on oltava sama määrä jokaisella rivillä.
 - Mikäli yhden rivin sarake sisältää rivinvaihtomerkkejä tai erotinmerkkejä osana dataa, on kyseinen data ilmaistava lainausmerkkien sisällä.
 - Mikäli data sisältää lainausmerkkejä, on kyseinen data ilmaistava lainausmerkkien sisällä ja yksittäinen lainausmerkki ilmaistava kahdella peräkkäisellä lainausmerkillä.

Liitä testausraporttiin liitteeksi:

- Sovelluksen tallentamat väliaikais- ja tulostiedostot, sekä kuvaus asetuksista joilla tulostiedostot on viety.

4.2.2 Testitapaukset

5. Tulostiedoston ominaisuudet:

- 5.1 Tulostiedoston tiedostonimi sisältää tiedot kokeesta, koehenkilöstä, tehtävästä ja tehtävän suoritusajasta.
- 5.2 Tulostiedosto on CSV-muotoinen.
- 5.3 Tulostiedoston ensimmäinen rivi nimeää eri sarakkeet.
- 5.4 Tulostiedosto sisältää yhden koehenkilön yhden tehtävän valittujen laitteiden valitut dataosiot.

- 5.5 Tulostiedoston sarakkeita ovat sovelluksen yhtenäisen aikaleiman sarakkeet (UNIX-aika, kehysnumero ja alikehysnumero), sekä valitut dataosiot.
- 5.6 Tulostiedoston jokainen rivi sisältää sovelluksen aikaleiman.

4.3 Verkkoyhteystestit

4.3.1 Vaatimukset ja suoritusohjeet

Vaatimukset:

- Yhden testilaitteen on oltava yhdistetty muuten kuin suoraan samaan tietokoneeseen jossa 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).
- 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.

Suoritusohjeet:

- 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 säilytettävä johdonmukaisina myös katkonaisina.

Liitä testausraporttiin liitteeksi nämä:

- Kuvaus käytetystä testiverkosta (esim. langallinen/langaton), mahdollinen kellojen erotus, käytetyt osoitteet ja portit, katkaisu- ja palautusajat

4.3.2 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.

4.4 Ajosimulaattori

4.4.1 Vaatimukset ja suoritusohjeet

Vaatimukset:

- Eepsoft-ajosimulaattori pitää olla asennettuna ja saatavilla joko samalla koneella jossa testitapaukset suoritetaan, tai erillisellä lähiverkon kautta yhteydessä olevalla laitteella.
- Ajosimulaattoriin tulee olla asennettu Syncsterin tukema yhteyskripti `peltihamsteri_udp.lua`, ja kyseinen skripti pitää ladata simulaattoria käynnistettäessä. Tarkista, että skriptissä olevat IP-osoite ja portti osoittavat oikeaan paikkaan.

Suoritusohjeet:

1. Valitse ajosimulaattorista kartaksi `mLaakso`, ja suoritettavaksi tehtäväksi `User -> Here -> Occlusion -> Start`. Tämä käynnistää okklusioharjoituksen, jossa ajonäkymä pimenee ellei valittua painiketta (`Extral`) paineta säännöllisesti.

Voit myös vaihtoehtoisesti kokeilla seuraavia karttoja:

- vaihtoehtoinen okklusioajo: `City`, tehtävä `User -> Here -> Occlusion practice -> Start`, tai
 - DONE-merkintä: `3way`, tehtävä `User -> UseOnly (5)`.
2. Aja kenttää jonkin matkaa. Käytä rattiohjaimen tai näppäimistön painiketta, joka kytkee pimennyksen hetkeksi pois päältä. Kokeile halutessasi ajaa käsi-vaihteella ja kokeile sekä vilkkuja että käsijarrua.
 3. Lopetettuasi paina *End of Exercise* ajosimulaattorin taukovalikosta. Tämä lähettää sovellukselle EOE-merkinnän, samalla tavalla kuin okklusioharjoituksen käyttämät FADE-merkinnät lähetetään.

4.4.2 Testitapaukset

7. Ajosimulaattoridatan kaappaus:

- 7.1 Ohjeiden mukaisella toiminnalla muodostetut dataosiot toistuvat datassa oikein, sisältäen: aikaleimat, X, Y, Z, kaasuu, jarruu, ohjauspyörän asento, nopeus (km/h, m/s, ja kierrosnopeus), rengasosu-

mat ja rengas pois tieltä sekä etäisyys kaistan keskipisteeseen (vain city-kartassa).

- 7.2 FADE- ja EOE-merkinnät toistuvat oikeissa kohdissa datassa, huomioiden suhteellisen aikaleiman.

4.5 Silmänliikekamera

4.5.1 Vaatimukset ja suoritusohjeet

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.

Suoritusohjeet:

- Ohjelmoi muutama eri Area of Interest (AOI) D-Labissa, ja nauhoita D-Labilla tästä silmänliikedataa. Siirrä katsettasi eri AOI:den välillä. Vie D-Labin tuottama tiedosto Export-toiminnolla. Näkyvätkö nämä AOI:t D-Labin tiedostossa ja Syncster-sovellukseen tuoduissa tiedostossa samassa järjestyksessä?
- Kokeile sulkea toinen tai molemmat silmät. Vastaavien koordinaattien pitäisi muuttua nolllaksi silmänliikedatassa.
- Kokeile vaihtaa aikaleima järjestysluvuiksi. Mitä tapahtuu? Reagoiko sovellus hallitusti muutokseen?

Liitä testausraporttiin liitteeksi:

- Kuvaus D-Labin asetuksista (asetustiedosto, sanallinen kuvaus muutoksista), sekä D-Labin tallentamat tulostiedostot

4.5.2 Testitapaukset

8. Silmänliikedatan kaappaus:

- 8.1a** Ohjeiden mukaisella toiminnalla muodostetut dataosiot toistuvat datassa oikein, sisältäen D-Labin asetusten mukaisesti: 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 sekä katseen X ja Y kohteeseen suhteutettuna.

- 8.1b** Aikaleimattu AOI-data on mahdollista tuoda sovellukseen, siten että kerätyt tiedot toistuvat ajallisesti oikein muun datan kanssa.
- 8.2** Sovellus kykenee tulkitsemaan tallennuksen aikana vaihtuvia dataformaatteja ja varoittaa käyttäjää, mikäli muuttunutta dataformaattia ei pystytä tulkitsemaan tai synkronoimaan riittävän tarkasti.

4.6 Toissijaisen tehtävän Android-laite

4.6.1 Vaatimukset ja suoritusohjeet

Vaatimukset:

- Testijoukon suoritukseen käytettävässä Android-laitteessa (tai -laitteissa) on oltava asennettuna keräämiseen käytettävä sovellus.

Suoritusohjeet:

- Kokeile mahdollisuuksien mukaan eri Androidin versioita, jotta voidaan varmistua yhteensopivuudesta eri käyttöjärjestelmäversioiden kanssa.
- Kokeile erilaisia sormennäpätysrytmejä, sekä nopeita että hitaita. Näiden on toistuttava oikein aikaleimoissa; tämän voi todentaa esim. Excel-graafilla.

Liitä testausraporttiin liitteeksi:

- Valittujen Android-laitteiden järjestelmäohjelmistojen versiot ja laitteiden mallit.

4.6.2 Testitapaukset

9. Kosketusdatan kaappaus:

- 9.1 Ohjeiden mukaisella toiminnalla muodostetut dataosiot toistuvat datassa oikein, sisältäen kosketuksen absoluuttisen ja suhteellisen aikaleiman laitteen kellon mukaan.

4.7 EEG-laite

4.7.1 Vaatimukset ja suoritusohjeet

Vaatimukset:

- DSI-Streamerin (v1.08.28) on oltava käynnissä, ja kypärän kytkettynä tietokoneeseen Bluetoothilla.

Suoritusohjeet:

- Kokeile liikuttaa esim. leukaa datan kaappauksen aikana. Leuan liikuttaminen luo kaapattuun dataan runsaasti vaihtelua, jonka pitäisi toistua selkeästi tulostiedostossa esim. Excelillä visualisoituna.

4.7.2 Testitapaukset

10. Aivosähkökäyrädatan kaappaus:

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

5 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, 14.5.2019
- [4] Sauli Flinkman, Jere Junntila, 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