**Projekti PELI/IOT (RYHMÄ 2)**

DOKUMENTIN VERSIOHISTORIA

|  |  |  |  |
| --- | --- | --- | --- |
| VERSIONUMERO | PÄIVÄMÄÄRÄ | MUUTOSPERUSTE | TEKIJÄ / HYVÄKSYJÄ |
| 1.0 |  | Projektisuunnitelma valmis |  |
| 0.3 | 2.4.2020 | Palaverit ja loput kohdat täytetty. | Jaakko Ikäheimo |
| 0.2 | 23.3.2020 | Päivitystä joihinkin kohtiin | Konsta Holm |
| 0.1 | 22.3.2020 | Projektisuunnitelman raakaversio | Jaakko Ikäheimo |
| 0.04 | 19.2.2018 | Lukujen 5 ja 6 yhdistäminen, luvun 4.3 lisäys, koko ohjetekstin muokkaus | Tuula Hopeavuori |
| 0.03 | 25.10.2017 | Kpl 2 projektin tavoitteiden kuvaamisen lisääminen | Kari Jyrkkä |
| 0.02 | 21.10.2017 | Kpl 2.2 tekemisen kuvaus; 3.1 Roolijako lisätty; 6 suunnitelmaa täsmennetty | Jaakko Kaski |
| 0.01 | 14.6.2015 | Dokumentin pohja valmis | Kari Jyrkkä |

Sisällysluettelo

1 Dokumentin tarkoitus 3

2 PROJEKTIN SISÄLTÖ 3

2.1 Tausta ja lähtökohdat 3

2.2 Tehtävä ja tavoitteet 4

2.2.1 Käytettävät teknologiat ja niiden paikka projektissa 4

Ohjain (Nucleo) 4

2.2.2 Sensoreista ja niiden kalibroinnista 6

2.2.3 Luonnontieteet (fysiikan ja matematiikan osio) 7

3 PROJEKTIN ORGANISOINTI 8

3.1 Toimittajan projektiryhmä 8

3.2 Tilat 9

3.3 Työkalut, laitteistot ja verkkoyhteydet ja budjetti 9

4 TOTEUTUSSUUNNITELMA 9

4.1 Projektin vaiheistus ja aikataulu 9

4.2 Projektin toimitukset 11

4.3 Riskienhallinta 12

5 OHJAUS- JA VIESTINTÄSUUNNITELMA 12

5.1 Palaverikäytännöt 12

5.2 Raportointi ja tiedottaminen 13

5.3 Dokumentointi 13

# Dokumentin tarkoitus

Tämän dokumentin tarkoitus on kuvata Sulautetun järjestelmän sovellusprojektin kevään 2020 projektin suunnittelu ja sen organisointi, toimien täten alustavana projektisuunnitelmana. Suunnitelma sisältää koko projektin ja siihen osallistuvien henkilöiden toiminnan suunnittelun, organisoinnin, valvonnan ja johtamisen. Suunnittelu aloitetaan määrittelemällä projektin sisältö sekä ne päämäärät ja rajoitteet, jotka kuvaavat ja sitovat projektia.

Suunnitteluprosessi sisältää vaiheet, joissa arvioidaan rakennettavan sovelluksen koko ja muut tarvittavat resurssit, projektin aikataulu, riskien huomioonottaminen ja hallinta sekä projektinhallinta. Tuloksena syntyy projektisuunnitelma, joka kertoo, kuinka tavoitteet saavutetaan käytössä olevilla resursseilla.

Dokumentti toimii myös projektin aikana tehtävien päätösten ja käytettyjen teknologioiden välityskanavana projektijakson vastuuopettajille.

Tämä dokumentti on tarkoitettu projektiorganisaation käyttöön ja se liitetään myös osaksi lopullisen projektin dokumentaatiota.

# PROJEKTIN SISÄLTÖ

## Tausta ja lähtökohdat

Projekti on osa tieto- ja viestintätekniikan insinöörin koulutusohjelmaan kuuluvaa opintojaksokokonaisuutta, ja on yksi useammasta opintojen aikana toteutettavasta projektista.

Projektin luonteesta johtuen, sen tilaajana toimii Oulun ammattikorkeakoulun tekniikan yksikkö, joka myös järjestää opintojakson, minkä yhtenä suoritusedellytyksenä tämä projekti toteutetaan. Koulutusohjelman aikaisemmissa opintojaksoissa projektin toteuttajat ovat jo toteuttaneet muutaman projektin kyseiselle tilaajalle.

Projektin toteuttajien tavoitteena on suoriutua projektijaksosta ansioikkaasti onnistuneen projektin ja sekä toteutetun että testatun tuotteen myötä.

## Tehtävä ja tavoitteet

Projektin tarkoitus on tuottaa pilvessä palvelimelle toteutettu selainpeli, mitä pystytään kontrolloimaan Nucleon sensoreilla Raspberry Pin kautta palvelimeen muodostetun yhteyden avulla. Peliä pystytään pelaamaan myös ilman ohjainta nettiselaimen ja näppäimistön avulla.

Projekti toteutetaan kehittäjien mielenkiinnosta ja sen kohderyhmänä toimii muut TVT19KMO:n oppilaat sekä käytetyistä teknologioista kiinnostuneet ohjelmistokehittäjät.

### Käytettävät teknologiat ja niiden paikka projektissa

### Ohjain (Nucleo)

Projektin sulautettuna järjestelmänä toimii Nucleo-mikrokontrollerin (4), ja alustavasti, analogisensorin ja painonappialustan avulla toteutettu ”peliohjain”. Käyttäjä pystyy ohjaimen avulla kontrolloimaan projektin aikana toteutettavaa peliä. Ohjaimen syöte välitetään sarjayhteydellä eteenpäin… Projektin edetessä ohjaimeen saatetaan liittää lisää sensoreita esim. pelin tilan muuttamiseksi.

**Ohjain-palvelin-rajapinta (Raspberry Pi)**

Projektin aikana toteutettavan ohjaimen ja pelin kommunikointi toteutetaan Raspille (5) toteutettavan, Pythonilla (6) ohjelmoidun rajapinnan avulla. Rajapinta välittää ohjaimesta sarjayhteyden kautta käyttäjän tuottaman syötteen palvelimelle, mikä päivittää syötteen perusteella pelin tilaa.

Pythonilla toteutetaan Raspille myös käyttöliittymä ohjaimen ja mahdollisesti myös pelin tilan tarkastelua varten. Käyttöliittymän toteuttamisessa käytetään apuna graafisten käyttöliittymien luomiseen tarkoitettua Tkinter-kirjastoa.

**Palvelin**

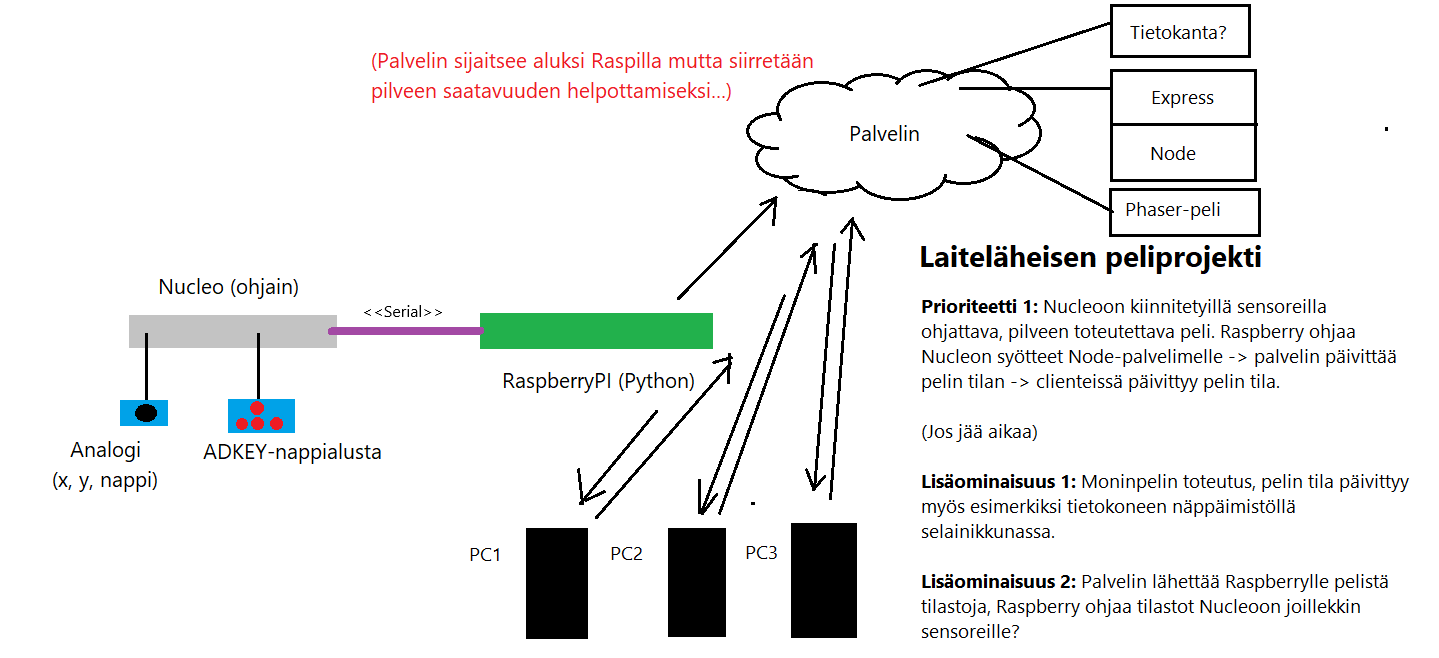
Palvelimen tehtävä on välittää itse peli loppukäyttäjille, ja ohjaimessa tai selaimessa ilmenevät tapahtumat muille käyttäjille eli selainikkunoihin (tai ohjaimelle). Palvelin rakennetaan Node-ympäristöön (7), missä käytetään apuna erilaisia kirjastoja kuten Express.js (8) ja Socket.IO (9). Palvelinta ylläpidetään aluksi Raspilla mutta se tullaan siirtämään lopuksi pilvipalveluun saatavuuden parantamiseksi.

**Peli**

Projektin peli toteutetaan Javascript-ohjelmointikielellä käyttäen apuna webbipelien luomiseen tarkoitettua Phaser-kirjastoa (10). Kirjasto sisältää kaiken tarvittavan projektin aikana toteutettavan pelin luomiseksi, kuten Matter.js-kirjastoon pohjautuvan fysiikkamoottorin. Pelin tapahtumien välitys palvelimelle toteutetaan socket-yhteyden ja Socket.IO-kirjaston avulla.

**Tietokanta**

Pelin toteutuksessa ei ainakaan alustavasti käytetä tietokantaa. Jos projektin aikana ilmenee tarve tietokannalle, esim. pelin tilastotietojen ja tilan tallennukseen, toteutetaan palvelimelle koodi tietokannan (SQL, MongoDB tai PostgreSQL) kanssa kommunikoimiseksi.



Kuva 1. Alustava suunnitelma projektin arkkitehtuurista, teknologioista ja toiminnallisuuksista.

### Sensoreista ja niiden kalibroinnista

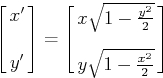
Alustavasti projektissa ei käytetä kuin kahta erittäin yksinkertaista sensorimoduulia:

* ADKeypad (1.)
* KY-023 Joystick module

ADKeypad-painonappialusta palauttaa tietyn amplitudisen jännitevasteen jokaiselle painetulle napille. Koska painonappialustaa on hyödynnetty jo aiemmassa projektissa, löytyy siitä luettavan analogisen jännitearvon muuntamiseen tietyksi painonapiksi koodi aiemmin kirjoitetusta ADKey-luokasta. Luokka implementoi lukumetodissaan myös debouncen, mikä tekee yksittäisistä ja jatkuvista nappien painallusten tarkkailemisesta huomattavasti helpompaa.

KY-023 joystick -moduuli sisältää kolme eri ulostuloa: x-akselin ja y-akselin potentiometrin lukemat, sekä analogin napin tilan. Koska potentiometrit palauttavat lukeman väliltä [0V, referenssijännite] neliöasteikolla, tulee ne pelin syötettä varten kartoittaa välille [-1, 1] neliöasteikosta ympyräasteikkoon (3). Tämä kartoitus ja sensorin kalibrointi pyritään toteuttamaan testaamalla joystickin potentiometrien tuottamaa arvoa eri kääntökulmille ja lisäämällä tarvittaessa potentiometreiltä luettuun raakadataan hieman offsettia. Esimerkkejä tavoiteltavista joystickin lukemista:

* ylhäälle käännetty (0, -1)
* alas käännetty (0, 1)
* vasemmalle käännetty (-1, 0)
* oikealle käännetty (1, 0)



Kuva 2. Neliöasteikon kartoitus ympyräasteikolle.

Jos aikaa jää pyritään näiden lisäksi vielä toteuttamaan joitain pelin tilan muuttujien, kuten esimerkiksi pisteiden, esittäminen Nucleoon liitetyillä komponenteilla (ledit, näyttö). Tämän lisäksi on suunniteltu vielä pelin tilan muuttamista esimerkiksi lämpötilasensorista luetun arvon perusteella, mutta se jää nähtäväksi kerkeääkö tällaista ominaisuutta implementoida… Projektissa käytetyistä sensoreista ja komponenteista luvassa enemmän tietoa projektin edetessä ja loppuraportissa.

### Luonnontieteet (fysiikan ja matematiikan osio)

Projektissa käytetään fysiikan periaatteita toteutettavan pelin ohjelmakoodissa. Suurin osa käytettävästä fysiikasta kuten esimerkiksi kappaleisiin kohdistettavista voimista hoidetaan Phaser-kirjastossa olevan fysiikkamoottorin avulla. Fysiikkamoottorin avulla pyritään saamaan realistisen tuntuiset ”aidot” fysiikat peliin kuten kappaleiden ja pelaajan painon tunteminen ohjatessa. Peliin mahdollisesti luodaan myös gravitaatiovoiman lisäksi muita luonnonvoimia kuten tuulta ja tietyillä alustoilla kitkaa tai kimpoamista. Käytetyistä fysiikan periaatteista annetaan tarkempi kuvaus projektin loppuraportissa.

# PROJEKTIN ORGANISOINTI

## Toimittajan projektiryhmä

Toimittajan projektiryhmään kuuluvat taulukon 1 henkilöt.

TAULUKKO 1. Projektiryhmän henkilöt

|  |  |
| --- | --- |
| NIMI: Jaakko Ikäheimo | |
|  | Ohjelmistosuunnittelija |
| YHTEYSTIEDOT | Organisaatio: TVT19KMO, OAMK OY  Sähköposti: c8ikja00@students.oamk.fi  Puhelin: 0405494838 |

|  |  |
| --- | --- |
| NIMI: Konsta Holm | |
|  | Ohjelmistosuunnittelija |
| YHTEYSTIEDOT | Organisaatio: TVT19KMO, OAMK OY  Sähköposti: t8hoko00@students.oamk.fi  Puhelin: |

|  |  |
| --- | --- |
| NIMI: Arttu Rusanen | |
|  | Ohjelmistosuunnittelija |
| YHTEYSTIEDOT | Organisaatio: TVT19KMO, OAMK OY  Sähköposti: t6ruar00@students.oamk.fi  Puhelin: |

## Tilat

Projektin aikana käytetään kunkin ohjelmistosuunnittelijan itselleen sopivaksi kokemaansa työtilaa, eli pääasiallisesti kotitoimistoa. Nämä tilat valittiin projektin luonteesta ja maailmalla jylläävästä koronavirus-epidemiasta johtuen.

Näiden kiinteiden tilojen lisäksi työryhmä käyttää myös työskentelytilanaan ryhmän Discord-palvelinta (11), missä ryhmänjäsenet pystyvät kommunikoimaan reaaliaikaisesti.

## Työkalut, laitteistot ja verkkoyhteydet ja budjetti

Koska projektissa käytetyt laitteet ovat tilaajalta tai toteuttajilta peräisin, sekä käytetyt digitaaliset työkalut ilmaisia, projektin suunnitteluvaiheessa arvioitu budjetti pysyy nollassa eurossa.

Kukin projektiryhmän jäsen käyttää itselleen sopivia ohjelmointityökaluja sekä omaa verkkoyhteyttään.

# TOTEUTUSSUUNNITELMA

## Projektin vaiheistus ja aikataulu

Koska projektille asetettu kesto on vain lähes kuukauden mittainen, sille ei pystytä laatimaan hyvin selkeää vaiheistusta tai aikataulua.

Projektin tekeminen aloitetaan kevään 2020 neljännen kouluperiodin kolmannella viikolla ja sen päättymisajankohdaksi asetetaan periodin viimeinen viikko. Projektin aikana tuotettu ohjelmakoodi pyritään kuitenkin jo toteuttamaan ja testaamaan viikkoa ennen tätä.

Projektin tekemiselle on varattu noin 5 kalenteriviikkoa. Projekti pyritään suorittamaan noudattaen EVO-mallia, myös ottaen hieman mallia Scrum ketterässä kehitysmenetelmässä käytetyistä päiväpalavereista ja product backlogista, eli toteuttaen ominaisuus kerrallaan, välttävät ominaisuudet ensiksi toteutettaen. Jos aikaa jää toteutetaan lisäominaisuudet.

TAULUKKO 1. Projektivaiheet

|  |  |  |
| --- | --- | --- |
| PROJEKTIN VAIHE | AIKATAULU | VAIHESEEN KÄYTETÄÄN PÄIVIÄ |
| Projektin suunnitteluvaihe | Viikko 13 | 3 |
| Järjestelmän suunnitteluvaihe | Viikko 13 | 4 |
| Moduulitoteutusvaihe | Viikko 14-15 | 14 |
| Moduulitestausvaihe | Viikko 14-15 | 2 |
| Integrointitoteutusvaihe | Viikko 16 | 7 |
| Integrointitestausvaihe | Viikko 16 | 2 |
| Projektin lopetusvaihe | Viikko 17 | 2 |
|  | **YHTEENSÄ** | **32** |

## Projektin toimitukset

Kaikki projektin toimitukset ovat saatavilla projektin GitHub-repositoryssa (12).

Projektin suunnitteluvaiheessa tiedossa olevat toimitukset näkyvät taulukossa 2.

*2.TAULUKKO 2. Projektin toimitukset*

|  |  |  |
| --- | --- | --- |
| TOIMITETTAVA TULOS | AIKATAULU | HYVÄKSYMISTOIMET |
| Projektisuunnitelma | 29.03. | Tarkastus (toteuttajat) |
| Palaverimuistiot | 24.04. | Tarkastus (sihteeri/toteuttajat) |
| Ohjaimen koodi Mbedille | 05.04. | Testaus (toteuttaja) |
| Ohjain-peli rajapinnan koodi Raspberry Pille | 12.04. | Testaus (toteuttaja) |
| Pelin palvelimen koodi | 19.04. | Testaus (toteuttajat/testaajat) |
| Pelin koodi | 19.04. | Testaus (toteuttajat/testaajat) |
| Demo-video | 24.04. | Tarkastus (toteuttajat) |
| Projektiposteri | 24.04. | Tarkastus (toteuttajat) |
| Projektin loppuraportti | 24.04. | Tarkastus (toteuttajat) |
| Projektiesitys | 24.04. | Tarkastus (toteuttajat) |

## 

## Riskienhallinta

Projektin ainoat suunniteltavissa olevat riskit ovat laitteiden mahdolliset toimintahäiriöt. Laitevikojen ilmetessä pyrimme käyttämään kehitysryhmän muiden jäsenten laitteita.

Projektin aikana ilmeneviin riskeihin pyritään reagoimaan mahdollisimman nopeasti jo heti riskin ilmenemisen aikana. Lisäksi riskejä kartoitetaan enemmän projektin viikkopalavereissa ilmenevien ongelmien myötä.

# OHJAUS- JA VIESTINTÄSUUNNITELMA

## Palaverikäytännöt

Projektin viikkopalaverit pyritään ajoittamaan projektin ajan jokaisen viikon keskiviikolle. Palaverit tapahtuvat ryhmän Discord-kanavalla, mitä kautta myös kutsut ja mahdollinen agenda toimitetaan. Palaverimuistiot liitetään projektin GitHub-repositoryyn samana päivänä, kun palaveritkin pidetään. Jos kaikki eivät pääse palaveriin, pyritään ne ajoittamaan uudelleen samalle viikolle.

Taulukon 3 mukaiset projektipalaverit:

TAULUKKO 3. Projektipalaverit

|  |  |  |
| --- | --- | --- |
| PROJEKTIPALAVERI | AIKATAULU | PALAVERIN TARKOITUS |
| Projektipalaveri 1 | 25.03. | Aloituspalaveri |
| Projektipalaveri 2 | 01.04 | Tilannepalaveri |
| Projektipalaveri 3 | 08.04 | Tilannepalaveri |
| Projektipalaveri 4 | 15.04 | Tilannepalaveri |
| Projektipalaveri 5 | 22.04 | Tarkistus-/lopetuspalaveri |
| (Projektipalaveri 6) | (29.04) |  |

## Raportointi ja tiedottaminen

Sisäiseen viestintään ryhmä käyttää pääasiassa Discordia. Pidämme virallisia kokouksia kerran viikossa, joista on muistiot, ja ”epävirallisia” kokouksia myös kerran viikossa. Discordin avulla ryhmänjäsenet pystyvät myös jakamaan tiedostoja ja kommunikoimaan toistensa kanssa reaaliajassa. Viestinnässä käytetään myös apuna GitHubiin puskettavien committien viestejä…

Käymme mahdollisia ongelmatilanteita läpi projektin ohjaajien kanssa. Ohjaajien kanssa kommunikoimme käyttäen koulun sähköpostia ja mahdollisissa hätätilanteissa AC-yhteyttä.

## Dokumentointi

Projektin alustavana dokumentaationa käytetään opintojakson Moodle-työtilasta löytyneitä dokumentteja, jotka täytetään projektin kuluessa. Dokumentaatio tallennetaan sekä projektiryhmän Moodle-kansioon että GitHub-repositoryyn.

Projektin aikana tuotettu dokumentaatio ja ohjelmakoodi säilytetään GitHubissa. <https://github.com/JIkaheimo/embedded-project-2k20>

Repositoryn README-tiedostoon on koottu tiivistelmä projektin tuotoksista eikä niistä mainita tässä sen enempää.

Lähteet

1. ADKeypad: <https://www.elecfreaks.com/estore/octopus-adkeypad.html>

2. KY-023 Joystick: <http://sensorkit.en.joy-it.net/index.php?title=KY-023_Joystick_module_(XY-Axis)>

3. Neliökoordinaattien siirtäminen ympyräkoordinaatistoon: <http://mathproofs.blogspot.com/2005/07/mapping-square-to-circle.html>

4. Nucleo STM32: <https://en.wikipedia.org/wiki/STM32>

5. Raspberry Pi: <https://fi.wikipedia.org/wiki/Raspberry_Pi>

6. Python-dokumentaatio: [https://docs.python.org/3/](https://docs.python.org/3/%20)

7. Node.js: <https://en.wikipedia.org/wiki/Node.js>

8. Express.js: <https://en.wikipedia.org/wiki/Express.js>

9. Socket.IO-dokumentaatio: <https://socket.io/docs/>

10. Phaser 3 -dokumentaatio: <https://photonstorm.github.io/phaser3-docs/>

11. Discord: <https://en.wikipedia.org/wiki/Discord_(software)>

12. GitHub: <https://fi.wikipedia.org/wiki/GitHub>