



Virtuaaliajopolku robotin tekoälyn ohjaamiseksi

Kari Maaheimo

Opinnäytetyö, AMK

Toukokuu 2022

Insinööri (AMK), Tieto- ja viestintätekniikka

Maaheimo, Kari

Virtuaaliajopolku robotin tekoälyn ohjaamiseksi

Jyväskylä: Jyväskylän ammattikorkeakoulu. Toukokuu 2022, 35 sivua.

Tietojenkäsittely ja tietoliikenne. Tieto- ja viestintäteknikan tutkinto-ohjelma. Opinnäytetyö AMK.

Julkaisun kieli: suomi

Verkkojulkaisulupa myönnetty: kyllä

Tiivistelmä

Suomen Puolustusvoimien Sotateknikan laitoksella Maanpuolustuskorkeakoulussa tutkitaan tekoälyä muun muassa Laykka-AMPGV-tutkimusprojektissa. Laykka on miehittämätön maastoajoneuvo, jonka halutaan kykenevän ajamaan autonomisesti sille määriteltyä reittiä pitkin. Laykkan autonomisen ajon ohjaaminen pyritään toteuttamaan Laykkan kameranäkymään piirretyllä virtuaalisella ajopolulla, jota pitkin Laykkan tekoäly voitaisiin kouluttaa ajamaan. Tällöin Laykkan ajoa voitaisiin ohjata määrittämällä kyseisen ajopolun reitti.

Tavoitteena oli toteuttaa kyseinen virtuaaliajopolku sekä sen reitin määrittämiseen liittyvät toiminnot. Varsinainen tekoälyn koulutus ja ohjaaminen on tarkoitus toteuttaa vasta myöhemmin. Työ toteutettiin soveltavana tutkimuksena, jossa aiemman tiedon pohjalta toteutetaan käytännön sovellus. Robottiautoja voidaan kouluttaa ajamaan autonomisesti rataa pitkin muun muassa Donkey Car -alustalla. Uutena sovelluksena on radan muuttaminen virtuaaliseksi, jolloin robotin ajoa voidaan ohjata rataa säätämällä.

Toteutukseen sisältyi useita osa-alueita. Polun on tarkoitus piirtyä Laykkan kameranäkymään 3D-projisoi-tuna polkuna, joka pysyy paikoillaan ympäröivään maastoon nähden. Polku pitää voida määrittää asetta-malla kartalle pisteitä, joiden kautta polun halutaan kulkevan. Pisteitä tulee voida lisätä klikkaamalla karttaa tai lisäämällä valmis koordinaatti erilliselle välilehdelle, johon kaikki pisteet ovat listattuina järjestyksessä. Polku piirtyy kyseisten pisteiden välille, ja polun leveys säätyy pisteiden säteiden mukaan. Pisteiden tarkkaa koordinaattia ja sädettä tulee voida muokata välilehden kautta. Yksittäisiä pisteitä sekä itse polku tulee myös voida tarvittaessa poistaa. Lisäksi pisteet, joita kohti Laykka on kulloinkin matkalla, tulee säteineen olla piirrettyinä kameranäkymässä varsinaisen polun lailla. Tällöin pisteet toimivat eräänlaisina ”ruokinta-pisteinä” tekoälylle sekä maamerkkeinä Laykkan operaattorille.

Työ saatiin onnistuneesti päätökseen pitkälti asetettujen tavoitteiden mukaisesti. Polun varsinainen testaa-minen jäi kuitenkin käytännön syistä vähäiseksi. On siis todennäköistä, että tuotos tulee vielä sisältämään haasteita, jotka löydetään vasta myöhemmän testauksen kautta. Kun onnistuneet testit saadaan suoritet-tua, voidaan siirtyä tekoälyn koulutukseen. Ajopolku on oleellinen osa Laykka-AMPGV-tutkimusprojektia ja jää aktiiviseen käyttöön.

Avainsanat (asiasanat)

Tekoäly, robotiikka, automaattiset järjestelmät, koneoppiminen, neuroverkot

Muut tiedot (salassa pidettävät liitteet)

Maaheimo, Kari

Virtual driving path for guiding a robot artificial intelligence

Jyväskylä: JAMK University of Applied Sciences, May 2022, 35 pages.

Information and Communications. Degree Programme in Information and Communication Technology. Bachelor's thesis.

Permission for web publication: Yes

Language of publication: Finnish

Abstract

The Department of Military Technology of the Finnish Defence Forces in The National Defence University of Finland does research on artificial intelligence in the form of Laykka-AMPGV research project. Laykka is an unmanned ground vehicle, which is meant to be able to drive autonomously along a specified path. The autonomous driving is planned to be carried out by using a virtual driving path painted into Laykka's camera view, along which Laykka's AI could then be taught to drive. This way Laykka could be guided to its destination by simply defining where the path should go.

The objective was the creation of the said driving path and the features used to define its route. Training of the AI would be carried out later. The implementation method used is applied research, where a new practical application is produced based on previous well-researched knowledge. Robot cars can be taught to drive autonomously along a track by utilizing for example the Donkey Car platform. The new application would be turning the track virtual. This allows controlling the robot's route by adjusting the track.

The implementation included multiple areas. The path must be drawn into Laykka's camera view as a 3D-projected path that visually remains fixed relative to the surrounding terrain. The route of the path must be defined by placing points on a map, via which the route should go. The points are added by either clicking the map or by giving a full coordinate on a separate tab, where all the points are listed in order. The exact coordinate and radius of each point must be adjustable on the said tab. Individual points as well as the path itself must also be removable when needed. In addition, the points, towards which Laykka is on its way to, must be painted into the camera view along with their radii much like the path itself. This way the points can function both as "feeding points" of sort for the AI as well as landmarks for Laykka's operator.

The task was completed successfully largely according to the objectives. Actual testing of the path remained minimal for practical reasons, which means it is likely that there are still some challenges that will only be found through further testing. Once tests have been successfully completed, we can move on to training the AI. The path is an essential part of the Laykka-AMPGV research project as a whole and will remain in active use.

Keywords/tags (subjects)

Artificial intelligence, robotics, self-acting systems, machine learning, neural networks

Miscellaneous (Confidential information)

Sisältö

1	Johdanto	4
2	Tausta, tarkoitus ja tavoitteet	5
3	Teoreettis-käsitteelliset lähtökohdat	7
3.1	Laykka	7
3.2	Tekoäly ja neuroverkot	8
3.3	Donkey Car	9
4	Käytetyt teknologiat	9
4.1	GitHub	9
4.2	Visual Studio Code.....	10
4.3	Python ja kirjastot	10
5	Toteutus	12
5.1	Virtuaalijopoku kameranäkymään	12
5.2	Polku karttanäkymässä	19
5.3	Maalipisteet kameranäkymään.....	21
5.4	Polun siirtäminen robotille.....	25
5.5	Koordinaatit ja kokonaisuuden toiminta	26
6	Tulokset	29
6.1	Esittely	29
6.2	Testaus	32
7	Pohdinta	33
	Lähteet	35

Kuviot

Kuvio 1 Laykka-AMPGV	8
Kuvio 2 Kuvankaappaus kameranäkymän keinohorisontista	13
Kuvio 3 Laykkan virtuaaliajopolun 3D-projektion luomisprosessi visualisoituna	15
Kuvio 4 Valmis ajopolku kameranäkymässä	16
Kuvio 5 Polun asetusvälilehti	17
Kuvio 6 Polun kaartuminen	18
Kuvio 7 Virtuaaliajopolkuun vaikuttavat keinohorisontin asetukset	18
Kuvio 8 Karttanäkymä sekä karttapistevälilehti	20
Kuvio 9 Uuden pisteen lisääminen karttapistelistan kautta	21
Kuvio 10 Funktio kehän pisteiden laskemiseksi	23
Kuvio 11 Algoritmi kehän 3D-projisoitujen pisteiden yhdistämiseksi viivoilla	23
Kuvio 12 Maalipiste ajopolkunäkymässä	24
Kuvio 13 Maalipisteen asetukset	24
Kuvio 14 Polkunäkymän luominen robotin päähän havainnollistettuna	26
Kuvio 15 Kameranäkymä polun ulkopuolelta	28
Kuvio 16 Lopputuotoksen kokonaisuus käyttöliittymässä	30

Lyhenteet

AI	Artificial Intelligence
AMPGV	Autonomus Multi-Purpose Ground Vehicle
GPS	Global Positioning System
IMU	Inertial Measurement Unit
MGRS	Military Grid Reference System
MPKK	Maanpuolustuskorkeakoulu
STKNL	Sotatekniikan laitos
UGV	Unmanned Ground Vehicle
VS Code	Visual Studio Code
RGB	Red Green Blue

1 Johdanto

Tekoälyn kehittyessä sen käyttökohteiden määrä kasvaa ja koneoppimista hyödynnetään yhä useammalla alalla. Yksi näistä aloista on maanpuolustus. Miehittämättömiä maastoajoneuvoja (UGV, Unmanned Ground Vehicle) on kehitelty eri maiden asevoimissa jo pitkään mutta muun muassa tekoälyn kehittymisen myötä ne ovat nyt ajankohtaisempia kuin koskaan. Aiemmin pääsääntöisesti kauko-ohjattaviin järjestelmiin voidaan nyt helpommin lisätä autonomiaa, eli niistä voidaan esimerkiksi tehdä itsestään ajavia. Autonomiseen ajamiseen sisältyy kuitenkin haasteita, kuten se, miten kertoa tekoälylle, minne sen halutaan ajavan tai kuinka ajaa maastossa ilman selkeää tietä, jota seurata. (Andersson 2021.)

Näitä haasteita pyritään ratkomaan Laykka AMPGV (Autonomous Multi-Purpose Ground Vehicle) - tutkimusprojektissa Suomen Puolustusvoimien Sotatekniikan laitoksella Maanpuolustuskorkeakoulussa. Laykka on miehittämätön maastoajoneuvo, jonka halutaan muun muassa kykenevän ajamaan autonomisesti haluttuun pisteeseen haluttua reittiä pitkin. Tämä pyritään toteuttamaan *virtuaalisella ajopolulla*. Kyseessä on kirjaimellisesti virtuaalinen polku, jonka Laykka näkee piirrettynä kameranäkymäänsä tavallisena tienä, jota pitkin Laykkan tekoäly voidaan kouluttaa ajamaan. Näin Laykkan autonomista ajoa voitaisiin ohjata yksinkertaisesti määrittämällä kyseisen polun reitti, jota Laykka sitten seuraisi.

Tässä työssä keskitytään kyseisen ajopolun sekä sen reitin määrittämiseen vaadittujen ominaisuuksien toteuttamiseen. Varsinainen tekoälyn koulutus suoritetaan myöhemmin, eikä se siis sisälly tähän työhön. Työssä selitetään aluksi tausta sekä tavoitteet ja esitellään käytetyt teknologiat. Tämän jälkeen käydään läpi työn toteutus osa-alueittain, esitellään tulokset ja kerrotaan testauksesta. Työn lopussa on pohdinta. Työssä esitellyt kuvankaappaukset Laykkan kameranäkymästä on selkeyden vuoksi otettu Laykkan ollessa sammuksissa. Sammutettu Laykka ei lähetä videokuva, joten kameranäkymä jää mustaksi. Tällöin esiteltävät asiat erottuvat taustasta selkeämmin.

Työ on toteutettu soveltavana tutkimuksena, jossa pyritään aiemman tutkitun tiedon pohjalta toteuttamaan käytännön sovellus. Työssä kehitellään ja toteutetaan tekoälyn koulutukseen perustuva virtuaalinen ajopolku tekoälyn ohjaamiseksi. Robottiautoja voidaan kouluttaa ajamaan autonomisesti rataa pitkin esimerkiksi Donkey Carin avulla (ks. luku 3.3), jolla Laykkan tekoälyn ajokoulutus pyritään toteuttamaan. Uutena sovelluksena on tässä työssä toteutettu ajopolku, joka on täysin virtuaalinen ja täten muokattavissa sekä säädettävissä.

Aiheesta, joka liittyy tekoälyn ohjaamiseen haluttuun suuntaan manipuloimalla sitä virtuaalisesti luoduilla rajoilla, jotka ovat nähtävissä ja seurattavissa, ei ole aiempaa akateemista julkaisua. Kyseisellä aiheella on siis yleistä tutkimuksellista arvoa tietotekniikan alalle.

2 Tausta, tarkoitus ja tavoitteet

Työ on osa Puolustusvoimien Maanpuolustuskorkeakoulun sotatekniikan laitoksen (MPKK STKNL) Laykka AMPGV -tutkimusprojektia. Laykka on miehittämätön maastoajoneuvo (robotti), jolle on tarkoitus lisätä autonomisia ominaisuuksia. Yksi oleellinen osa Laykkan autonomian kehitystä on virtuaaliajopolku, jonka kehittämiseen tämä työ keskittyy.

Työn tarkoituksena oli luoda virtuaalinen ajopolku, jota voitaisiin myöhemmin hyödyntää robotin tekoälyn kouluttamiseen ja ohjaamiseen. Tämän koulutuksen tavoitteena olisi opettaa robotin tekoäly ajamaan robotin kameranäkymään piirrettyä virtuaaliajopolkua pitkin, jolloin robotti voitaisiin ohjata ajamaan itseksensä haluttuun paikkaan virtuaaliajopolkua seuraamalla. Tämä työ ei kuitenkaan sisällä itse tekoälyn koulutusta, vaan keskittyy nimenomaan itse virtuaaliajopolkuun sekä sen määrittämiseen ja säätämiseen liittyvien toimintojen kehittämiseen ja toteuttamiseen.

Työn tekijä on ollut jo aiemmin mukana Laykka-AMPGV-tutkimusprojektissa kehittämässä muun muassa Laykkan graafista käyttöliittymää. Alusta on tekijälle siis jo ennestään tuttu, ja työ on suoraan jatkumoa aiemmalle kehitystyölle. Edellä mainittuun käyttöliittymään sisältyy kameranäkymä, johon polku piirretään. Käyttöliittymään sisältyy myös karttaikkuna, johon haluttu polku tulee voida määrittää käyttäjän toimesta.

Virtuaaliajopolku on Laykkan autonomian tutkimuksen ja kehityksen keskeisimpiä ominaisuuksia ja siis aivan välttämätön osa koko Laykka-AMPGV-tutkimusprojektia. Työn keskeisimpänä tavoitteena on saada polku ja siihen liittyvät oleelliset toiminnot valmiiksi tekoälyn hyödynnettäväksi.

Yksityiskohtaiset vaatimukset ja tavoitteet esitellään osa-alueittain luvussa 5. Työn kehitysprosessia johdettiin inkrementaalisen kehittämisen menetelmällä, jossa tekijälle annettiin uusia vaatimuksia ja tavoitteita palaverissa sitä mukaa kun työ eteni. Vaatimukset ja tavoitteet kohdistuivat pitkälti suoraan lopputulokseen eikä niinkään työprosessiin, jolloin tekijälle annettiin vapaat kädet toteutuksen suunnitteluun.

Tavoitteiden osalta työn toteutus jakautuu selkeisiin osa-alueisiin. Ensimmäisenä osa-alueena on ajopolun piirtäminen käyttöliittymän kameranäkymään. Polun on tarkoitus olla 3D-projektio, jossa polulla on punainen vasen laita, vihreä oikea laita ja sininen keskiviiva. Polun asennon ja ominaisuuksien on oltava säädettävissä ja polun on voitava kaartua kääntyvän tien lailla käänöksissä.

Toisena osa-alueena on polun lisääminen käyttöliittymän karttanäkymään. Kartalle on voitava määrittää pisteitä, joiden kautta polku kulkee. Kartan polulla on kameranäkymän polun tavoin oltava oikea ja vasen laita sekä keskiviiva. Toiveena oli, että polku saataisiin kaartumaan sulavasti pisteiden kautta, jos mahdollista. Pisteitä on tarkoitus voida määrittää klikkaamalla karttaa tai antamalla suoraan halutun pisteen koordinaatti sotilaskäytössä olevassa MGRS-muodossa. Pisteitä on myös voitava siirtää kartalla raahaamalla. Pisteillä on oltava säde, jonka mukaan polun leveys säätyy. Pisteiden on tarkoitus olla myös listattuina erillisellä karttapistevälilehdellä, jolta käsin pisteiden koordinaatteja ja säteitä voidaan muokata. Kyseisten pisteiden on lisäksi oltava piirrettyinä säteineen kameranäkymään varsinaisen polun tavoin.

Kolmantena ja viimeisenä osa-alueena on kokonaisuuden toiminta ja edellytykset koulutusdatan keräämiselle. Kokonaisuuden toiminnan kannalta on oleellista, että kaikki kokonaisuuden eri osat keskustelevat keskenään. Tämä tarkoittaa muun muassa sitä, että kameranäkymän polun on täsmäyttävä kartalle määritellyn polun kanssa. Pisteitä on myös merkittävä saavutetuiksi sitä mukaa kun robotti ajaa niitä läpi, jotta robotti osaa jatkaa kohti järjestyksessä seuraavana olevaa pistettä. Koulutusdatan keräämisen osalta tavoitteena on saada käyttöliittymän kameranäkymää vastaava

näkymä polkuineen luotua myös robotin päähän, jossa varsinainen tekoälyn koulutus ja ohjaus tulee tapahtumaan.

3 Teoreettis-käsitteelliset lähtökohdat

3.1 Laykka

”Laykka” viittaa sotatekniikanlaitoksen autonomiantutkimusprojekti Laykka-AMPGV:hen, jonka osana tämä työ on toteutettu. Kuviossa 1 esitelty Laykka itsessään on miehittämätön maastoajoneuvo, johon tässä työssä selkeyden vuoksi viitataan sanoilla ”Laykka” tai ”robotti”.

Laykkan kaltaiset sotilaskäyttöön suunnitellut miehittämättömät maastoajoneuvot, eli UGV:t (Unmanned Ground Vehicle) eivät sinänsä ole mikään uusi konsepti. Vastaavia autonomisia sekä kauko-ohjattavia järjestelmiä on viime aikoina kehitetty ja testattu ympäri maailmaa, esimerkiksi Venäjällä ja Israelissa. Tällaisten järjestelmien käyttökohteet voivat vaihdella tarpeen mukaan, toimien muun muassa aselavetteina, evakuointialustoina tai kuormajuhtina. (Andersson 2021.)

Laykka on kuitenkin joiltain osin ainutlaatuinen. Se eroaa selkeästi useimmista vastaavista järjestelmistä esimerkiksi halpuutensa ja uhrattavuutensa puolesta. Laykka kykenee muun muassa toimimaan eräänlaisena älykkäänä miinana. (Andersson 2022.)



Kuvio 1 Laykka-AMPGV (Salonen 2022)

3.2 Tekoäly ja neuroverkot

Tekoälyllä tarkoitetaan yleisesti jonkin ohjelman omatoimista ongelmanratkomiskykyä. Tekoäly kykenee tekemään omia päätöksiä saamansa datan perusteella ihmisten lailla, eli kyseessä on sanan mukaisesti “tekoäly” (IBM Cloud Education 2020). Tässä projektissa tekoälyllä viitataan Laykkan autonomiseen toimintaan. Laykkan tekoälyn on tarkoitus oppia ajamaan omin päin virtuaalijopolkua pitkin sekä tunnistamaan muun muassa erilaisia esteitä, joita Laykka saattaa polun varrella joutua väistämään.

Tekoälyn oppimisella viitataan Laykkan tapauksessa neuroverkkopohjaiseen tekoälyn koulutukseen, jossa tekoäly koulutetaan suorittamaan jokin tehtävä ennalta määritetyn koulutusdatan pohjalta. Tällaista voi olla muun muassa kuvantunnistus (Hardesty 2017).

Kuvantunnistuksessa neuroverkolle annetaan koulutusdatana kuvia tunnistettavasta kappaleesta eri kulmista ja erilaisissa ympäristöissä. Kuvien pohjalta neuroverkko pyrkii löytämään esimerkiksi

toistuvia muotoja tai kaavoja, joiden avulla tunnistaa kyseinen kappale missä tilanteessa tahansa. Lopputuloksena olisi siis tekoäly, joka kykenee kertomaan, löytyykö kyseinen kappale kuvasta vai ei. (Hardesty 2017.)

Koska tämä työ ei itsessään sisällä tekoälyn koulutusta, riittää käytännössä ymmärtää, että tekoälyn koulutukseen tarvitaan koulutusdataa ja että tekoäly voidaan kouluttaa toimimaan halutulla tavalla tämän koulutusdatan pohjalta. Laykkan koulutusdatana toimisivat kuvankaappaukset robotin kameranäkymästä virtuaaliajopolkua pitkin ajettaessa. Kuvankaappauksilla voidaan sekä kouluttaa Laykkan tekoälyä että ohjata Laykkan autonomista ajoa tekoälyn koulutuksen tapahduttua.

3.3 Donkey Car

Donkey Car on Python-pohjainen harrastelijoille suunnattu robottiautoalusta, jonka avulla voidaan kouluttaa robottiauto muun muassa ajamaan itsekseen rataa pitkin edellisessä luvussa mainitulla tavalla. Donkey Carissa koulutus tapahtuu keräämällä koulutusdataa ajamalla autoa rataa pitkin. Koulutusdatana toimivat auton omasta kameranäkymästä otetut kuvat. (Kreuzhuber 2020.)

Laykkan autonomisen ajon koulutus on tarkoitus toteuttaa hyödyntämällä Donkey Caria sekä sen virtuaalista simulaattoriympäristöä Donkey Gymiä. Laykkan koulutusdatana toimisivat kuvankaappaukset robotin kameranäkymästä, joissa robotti ajaisi tässä työssä luotua virtuaaliajopolkua pitkin. Ideana siis olisi se, että Laykka oppisi Donkey Carin tavoin ajamaan rataa pitkin, mutta kyseinen rata olisi täysin virtuaalinen ja muokattavissa. Tässä työssä ei kuitenkaan paneuduta tähän myöhemmin tapahtuvaan Donkey Carin hyödyntämiseen, vaan työ keskittyy nimenomaan edellä mainitun virtuaaliajopolun luomiseen.

4 Käytetyt teknologiat

4.1 GitHub

Git on versionhallintajärjestelmä, jonka avulla projektin koodi saadaan pidettyä tallessa repositoriona (engl. repository). Repositorion avulla koodiin tehtyjä muutoksia voidaan seurata ja hallita. GitHub puolestaan on verkkopohjainen palvelu, jonka kautta Gitin repositorioita voidaan hallita. (An Introduction to GitHub 2020.)

Kun projekti on tallessa Gitissä, voivat projektin osalliset kloonata (engl. clone) koodin mille tietokoneelle tahansa. Kun käyttäjä on tehnyt tietokoneellaan muutoksia koodiin, muutokset vietään versionhallintaan (engl. commit). Projektin GitHubiin tallennettu päärepositorio voidaan sitten päivittää vastaamaan tietokoneen paikallista versiota (engl. push). Jos huomataan, että jokin tietty koodimuutos on rikkonut jotain projektissa, voidaan kyseiset muutokset tarvittaessa perua. Koodi voidaan siis aina palauttaa aiempaan versioon, jossa kaikki on vielä toiminut, eikä projektia siis käytännössä voi vahingossa rikkoa kokonaan. (User Manual 2022.)

Laykkan lähdekoodi ovat tallennettuna GitHubiin. Työn toteutuksessa käytettiin Git Bash -ohjelmaa, jonka avulla Git-komentoja voi suorittaa suoraan terminaalissa.

4.2 Visual Studio Code

Työn koodi on kirjoitettu Microsoftin Visual Studio Code (VS Code) -ohjelmalla. VS Code on Microsoftin kehittämä koodinmuokkausohjelma, jolla ohjelmistokehittäjät voivat kirjoittaa ja muokata ohjelmien lähdekoodia usealla eri ohjelmointikielillä (Getting Started N.d.). Työn tekijän päätös VS Coden käyttämisestä perustui sen aiempaan käyttökokemukseen ja myös ohjelman monipuolisuuteen ja joustavuuteen: VS Codeen on saatavilla suuri määrä laajennuksia, joiden asentaminen käy helposti itse ohjelmasta käsin (Extension Marketplace N.d.), joista Laykkalle oleellimmat ovat Python ja Qt.

4.3 Python ja kirjastot

Python on laajasti käytössä oleva korkean tason ohjelmointikieli. Niin sanottuna ”scriptauskielenä” Pythonia ei tarvitse erikseen kääntää ennen sen suorittamista, mikä nopeuttaa koodin työstämistä. Python on suosittu ohjelmointikieli muun muassa yksinkertaisuutensa ja helppo-oppisuutensa vuoksi. (What is Python? Executive Summary N.d.).

Työ on kokonaisuudessaan karttanäkymää lukuun ottamatta toteutettu Pythonilla. Työn tekijällä ei ollut kokemusta Pythonista ennen Laykka-AMPGV-tutkimusprojektiin osallistumistaan, mikä puolestaan kertoo lisää kielen helppo-oppisuudesta.

Qt-kirjasto

Laykkan kannalta oleellisin Python-kirjasto on Qt, joka on graafisten käyttöliittymien kehittämiseen tarkoitettu kirjasto (About Qt 2022). Laykkan oma graafinen käyttöliittymä on toteutettu Qt:lla, joten suurin osa tästä työstä on toteutettu Qt-kirjastoa käyttäen. Kirjastoon sisältyy myös QPainter, jolla varsinaisen virtuaaliajopolun piirtäminen käyttöliittymän kameranäkymään on toteutettu.

OpenCV-kirjasto

OpenCV on avoimeen lähdekoodiin perustuva konenäkökirjasto, jolla voidaan muun muassa kouluttaa tekoäly tunnistamaan kuvista erilaisia asioita, kuten esimerkiksi muotoja, esineitä tai tekstiä (About N.d.). Laykkalle OpenCV on oleellinen tekoälyn koulutusta varten, jotta Laykka voi muun muassa oppia tunnistamaan esteitä. Tässä työssä OpenCV:n sisäisillä piirtotoiminnoilla on toteutettu virtuaaliajopolun piirtäminen robotin päässä, vaihtoehtona Qt:n vastaavalle toteutukselle.

Leaflet Vue ja OpenStreetMap

Laykkan graafisen käyttöliittymän karttanäkymä on toteutettu Leaflet Vuella, jonka karttapohjana toimii OpenStreetMap. OpenStreetMap on avoimeen lähdekoodiin perustuva karttapalvelu, jota on monipuolisen ja käyttäjäystävällisen rajapintansa vuoksi helppo hyödyntää erilaisten karttapohjaisten ohjelmien alustana (About OpenStreetMap 2022).

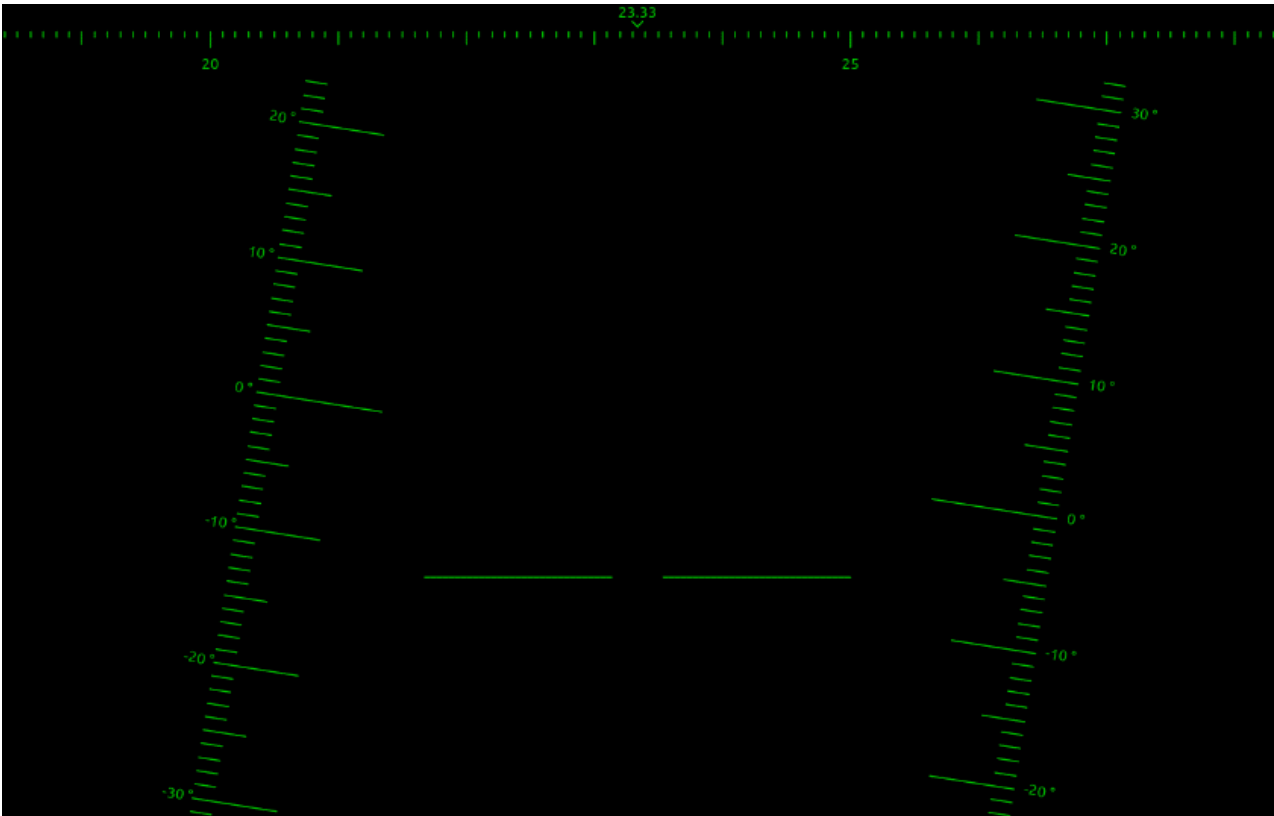
Leaflet on JavaScript-kirjasto interaktiivisten karttojen luomiseen. Sen karttapohjana toimii OpenStreetMap. Vue on puolestaan JavaScript-kehys, jonka tarkoitus on tarjota normaalia JavaScriptiä tehokkaampi ja monipuolisempi kehitysympäristö verkkopohjaisten käyttöliittymien luomiseen. (Abidi 2020.)

5 Toteutus

5.1 Virtuaaliajopolku kameranäkymään

Työn keskeisin osa on kameranäkymään piirrettävä virtuaaliajopolku. Virtuaaliajopolun on oltava näkyvissä sekä robotille että käyttäjälle. Polku tarvitaan robotille, jotta sen tekoälyä voidaan kouluttaa ja ohjata ajamaan kyseistä polkua pitkin. Käyttäjän on puolestaan nähtävä polku, jotta hän voi ajaa sitä pitkin manuaalisesti koulutusdatan tuottamiseksi ja jotta robotin autonomista ajoa voidaan seurata. Tämän haasteena on se, että sama polku pitää saada siis luotua sekä päätelaitteeseen, josta käsin robottia käytetään, että itse robotin päähän, joka tarvitsee näkymän tekoälyä varten. Työtä suunniteltaessa todettiin, että polku olisi parasta luoda käyttöliittymän päässä, mistä polku sitten lähetetään robotille. Tällöin polun luomista ei tarvitse turhaan suorittaa kahdesti eikä polun koodiin tehtyjä muutoksia tarvitse tehdä rinnakkaisesti kahteen eri paikkaan.

Laykkan graafisessa käyttöliittymässä oli jo ennestään kameranäkymä, johon polku oli tarkoitus lisätä. Työn tekijä on ollut jo aiemmin mukana kyseisen graafisen käyttöliittymän kehittämisessä, johon sisältyi muun muassa kuviossa 2 esitellyn keinohorisontin lisääminen edellä mainittuun kameranäkymään visualisoimaan Laykkan asentoa. Keinohorisontin ominaisuudet tarjosivat hyvän pohjan ajopolun kehittämiseksi, sillä polun on tarkoitus keinohorisontin tavoin pysyä ympäröivään maastoon nähden paikoillaan. Keinohorisontti sekä ajopolku hyödyntävät robotin IMUa (Inertial Measurement Unit) robotin asennon seuraamiseksi.

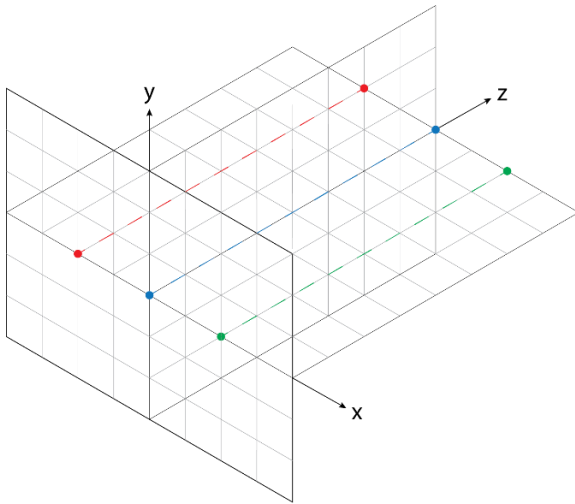


Kuvio 2 Kuvankaappaus kameranäkymän keinohorisontista

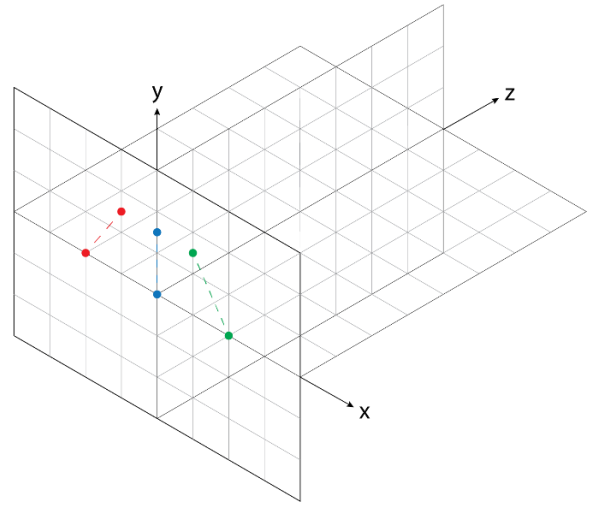
Varsinaisen polkunäkymän kehittäminen voidaan jakaa kahteen osa-alueeseen: polun ulkomuodon luomiseen ja polun käyttäytymiseen. Polun ulkomuodoksi haluttiin 3D-projisoitu polku, jolla olisi selkeä vasen ja oikea laita sekä keskiviiva – eli kolme vierekkäistä viivaa. Veneiden ja lentokoneiden navigointivalojen tavoin vasen laita olisi punainen ja oikea vihreä. Keskiviiva puolestaan olisi sininen. Andersson Laykka -projektin tutkimusryhmän tekemiin havaintoihin perustuen värien haluttiin olevan selkeät ja RGB-pohjaiset, jotta tekoäly tunnistaisi ne mahdollisimman hyvin kameranäkymästä. Kameranäkymässä ei tarvitse näkyä koko polkua kerralla, vaan riittää, että robotti näkee sen osan polkua, jolla se kullakin hetkellä on. Polun ulkomuodoksi siis riittää yksittäinen suora tai kaartuva pätkä, joka ohjaa robotin suoraan kohti seuraava maalipistettä.

Polun 3D-projektio saadaan aikaiseksi asettamalla ensin polun muodostavien janojen alku- ja päätepisteiden matemaattiset koordinaatit kolmiulotteiseen koordinaatistoon siten, että alkupisteet ovat Z-akselin koordinaatissa 0 ja loppupisteet polun haluttua pituutta vastaavalla Z-akselin koordinaatilla. Tämän jälkeen lisätään perspektiivi kertomalla pisteet matriisilla, joka käytännössä kor-

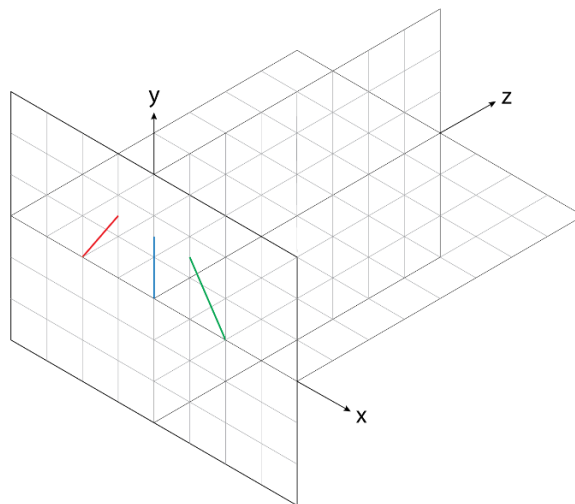
vaa pisteiden Z-koordinaatin perspektiivillä. Ilman perspektiiviä Z-akselin suuntaiset janat näkyisivät käyttäjälle pelkästään kolmena pisteenä. Perspektiivin lisäämisen lopputuloksena pisteillä on vain X- ja Y-koordinaatit, jotka suoraan vastaavat pisteitä käyttöliittymän kameranäkymässä. Kun lopuksi lisätään janat pisteiden välille, on lopputuloksena selkeä 3D-projisoitu kolmen janan muodostama virtuaalipolku. Tämä prosessi on visualisoitu tekijän piirrosten avulla kuviossa 3. Kuvankaappaus lopputuloksesta on nähtävissä kuviossa 4. Polun pisteiden alkuperäisiä X- ja Y-koordinaatteja muokkaamalla voidaan tarvittaessa säätää polun korkeutta, leveyttä ja sivuttaista siirtymää. Pituutta puolestaan voidaan säätää muokkaamalla janojen päätepisteiden Z-koordinaatteja.



1. Alkuperäiset pisteet asetettuna koordinaatistoon

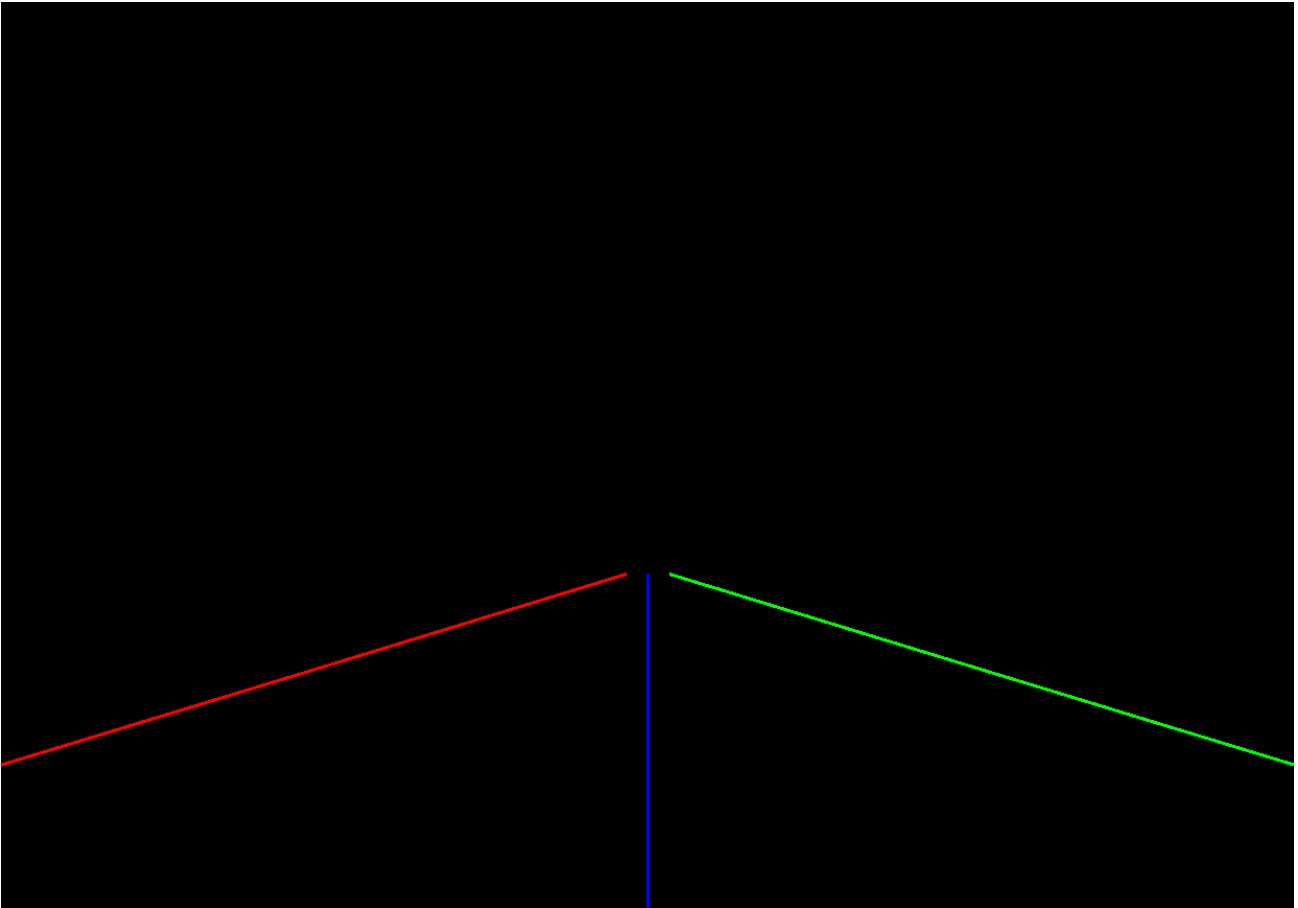


2. Pisteet perspektiivin lisäämisen jälkeen



3. Lopputulos, kun pisteet yhdistetään polun viivoilla

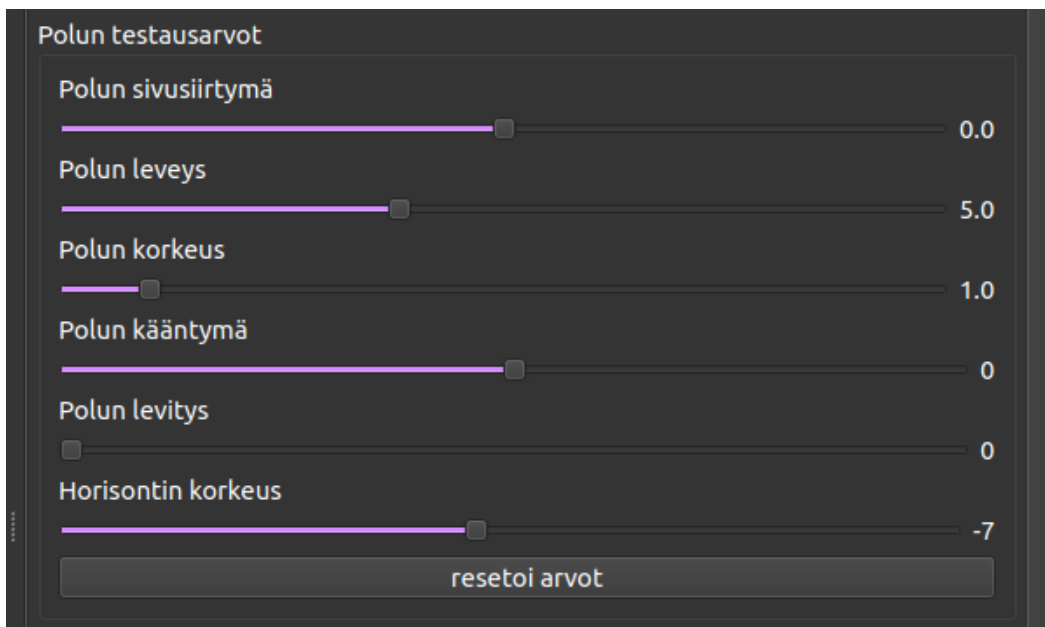
Kuvio 3 Laykkan virtuaaliajopolun 3D-projektion luomisprosessi visualisoituna



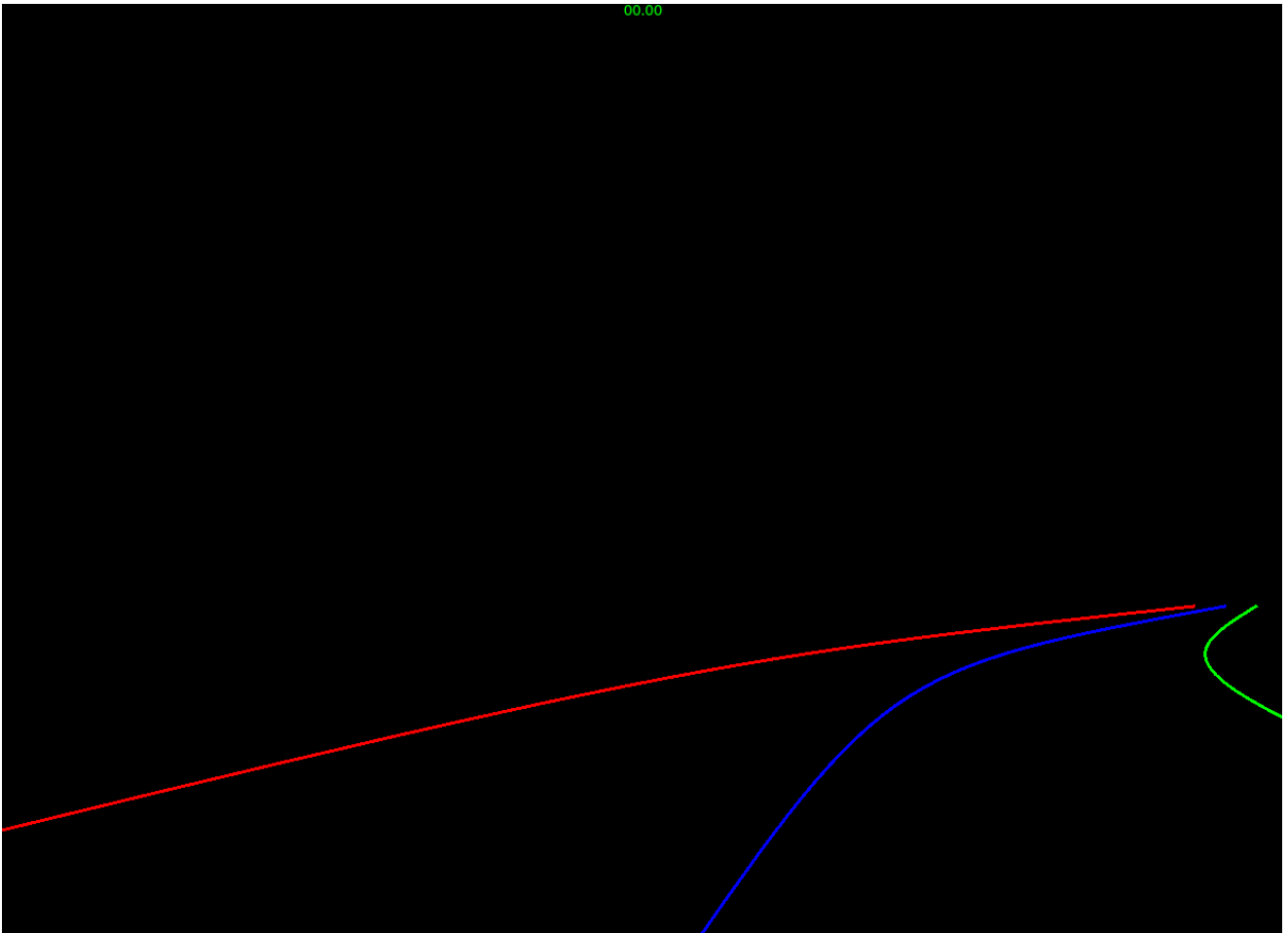
Kuvio 4 Valmis ajopolku kameranäkymässä

Polun käyttäytymiseen sisältyy polun kääntäminen ja siirtäminen siten, että se näyttää pysyvän paikoillaan ympäröivään maastoon nähden robotin liikkuesssa. Polun tulee siis käyttäytyä aivan kuin se olisi oikea fyysinen polku, jota pitkin robotti voi ajaa. Polun pitäminen paikoillaan suhteessa ympäröivään maastoon saavutetaan hyödyntämällä robotin IMU-dataa, joka kertoo muun muassa robotin asennon. Asennosta voidaan päätellä, miten polkua tulee siirtää, jotta robotin asento saadaan ”kumottua” ja polku palautettua alkuperäiseen asentoon. Jos IMU esimerkiksi kertoo robotin olevan kallellaan 20 astetta oikealle, saadaan polku takaisin vaakatasoon ympäröivään maastoon nähden kallistamalla sitä 20 astetta vastakkaiseen suuntaan vasemmalle. Koska robotin rungon asento suhteessa ympäröivään maastoon voi vaihdella ajon aikana, tulee polun asentoa ja kulmaa voida säätää tarvittaessa mihin asentoon tahansa, jotta se saadaan aina vastaamaan maastonmuotoja.

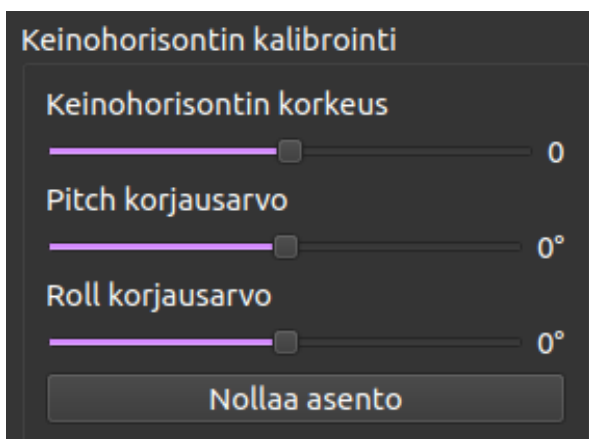
Polun kallistelu tapahtuu perspektiivin lisäämisen tavoin muunnosmatriiseilla kertomalla. Tarvittavia muunnosmatriiseja ovat käännökset X-, Y- ja Z-akselien ympäri. Niillä polku voidaan asettaa mihin asentoon tahansa. Polun kehittämisen ja testaamisen kannalta on olennaista, että polun asentoa ja muotoa voidaan säätää myös manuaalisesti käyttöliittymästä käsin. Käyttöliittymään tarvittiin siis polulle oma välilehti (ks. kuvio 5), jolta polun ominaisuuksia voidaan säätää reaaliajassa. Manuaalisia säätöjä tarvitaan viimeistään sellaisten säätöarvojen löytämiseksi, joilla polun asento näyttää parhaiten täsmäävän ympäröivän maaston kanssa tai joilla polku saadaan optimoitua robottimallille optimaaliseksi. Polun vaatimia säädettäviä ominaisuuksia ovat eri kallistuskulmat, sivusiirtymä, leveys, korkeus, kaartuminen sekä toimeksiantajan lisätoiveena polun levittäytymisen horisonttia kohti. Polun kaartuminen kääntyvän tien tapaisesti on toteutettu jakamalla polku erillisiin osiin, joita erikseen siirtämällä ja kääntämällä saadaan asetettua pisteet siten, että ne yhdistämällä bézier-käyrillä (suorien viivojen sijaan) saadaan aikaiseksi vaikutelma sivulle kaartuvasta polusta (ks. kuvio 6). Aiemmin mainitun keinohorisontin toimintaan perustuvat säädöt (robotin itsensä aiheuttamat sivu- ja etukallistumat, eli "roll" ja "pitch") tulevat polulle suoraan keinohorisontilta, jonka asetukset ovat nähtävissä kuviossa 7, eikä polku siis tarvitse niille erikseen omia asetuksia.



Kuvio 5 Polun asetusvälilehti



Kuvio 6 Polun kaartuminen



Kuvio 7 Virtuaaliajopolkuun vaikuttavat keinohorisontin asetukset

5.2 Polku karttanäkymässä

Virtuaaliajopolun varsinaisen reitin määrittäminen tapahtuu ensisijaisesti käyttöliittymän karttanäkymässä. Karttaikkuna on osa käyttöliittymää, mutta itse kartta on toteutettu erikseen Leaflet Vuella ja tulee erilliseltä karttapalvelimelta karttaikkunaan. Polku rakentuu yksittäisistä karttapisteistä, joilla on sekä koordinaatit MGRS (Military Grid Reference System) -muodossa että säde. Säde määrittää sen, kuinka lähelle pistettä robotin on päästävä, jotta piste lasketaan saavutetuksi ja robotti voi lähteä kohti seuraavaa pistettä. Lisäksi pisteen säde vaikuttaa polun leveyteen. Jos esimerkiksi polun yksittäisen osuuden alku- ja loppupisteet ovat molemmat säteiltään 2.5 metriä (halkaisija 5,0 metriä), on silloin kyseisen osuuden leveys 5,0 metriä. Vastaavasti, jos loppupisteen säteenä olisikin 5,0 metriä (halkaisija 10,0 metriä), levenisi polku tällöin viidestä metristä kymmeneen metriin loppupistettä kohti edettäessä.

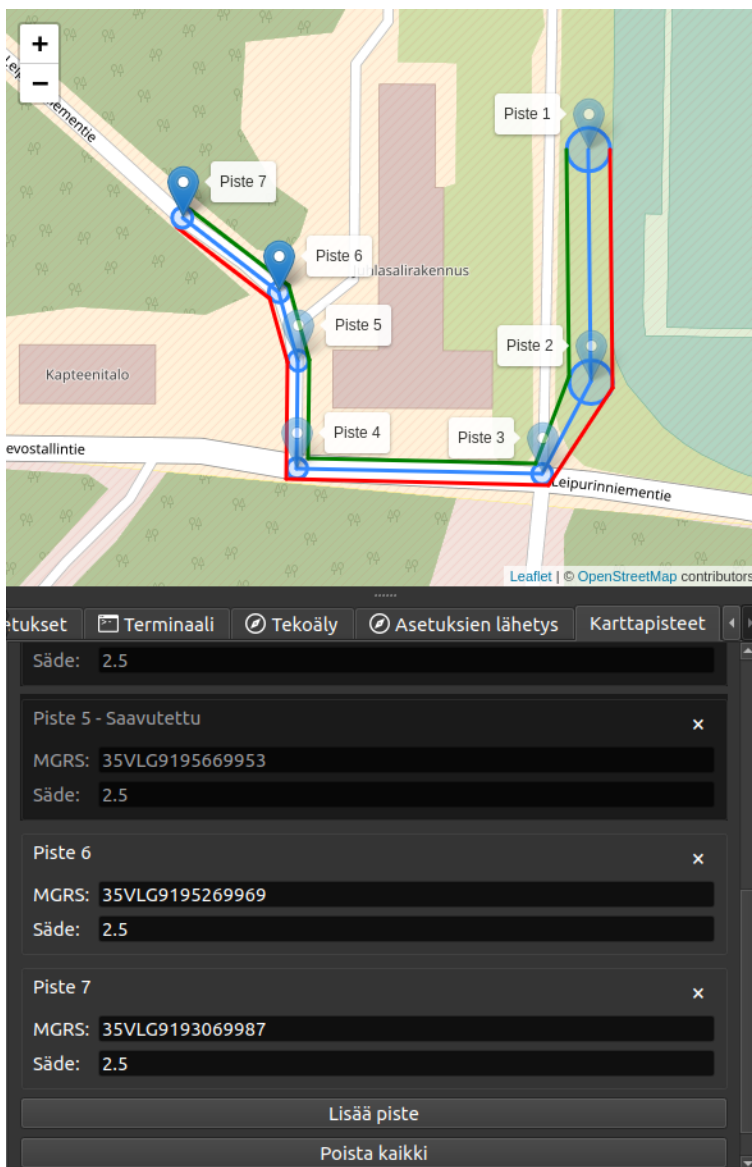
Karttapisteitä on tarkoitus määrittää ensisijaisesti klikkaamalla kartassa sitä kohtaa, josta polun halutaan lähtevän tai jonne polun halutaan seuraavaksi kulkevan. Kartan pisteitä voidaan siirtää *raaha ja pudota (drag and drop)* -menetelmällä sekä poistaa klikkaamalla olemassa olevaa pistettä.

Olemassa olevien pisteiden koordinaattien ja säteiden pitää tarvittaessa olla muokattavissa myös käsin. Karttapisteet piti siis listata omalle välilehdelle käyttöliittymään, joka on yhdessä karttanäkymän kanssa esitelty kuviossa 8. Välilehdeltä käyttäjä näkee järjestyksessä kaikki kartan pisteet koordinaatteineen ja säteineen. Välilehden kautta käyttäjä voi lisätä uusia pisteitä suoraan koordinaatilla (ks. kuvio 9) ja muokata olemassa olevien pisteiden ominaisuuksia. Lisättäessä pisteitä koordinaatilla ainoastaan validi MGRS-muotoinen koordinaatti kelpaa syötteenä. Olennaisena toimintona mukaan kuuluu myös yksittäisten pisteiden poistaminen listalta tai koko listan tyhjentäminen, eli koko polun poistaminen.

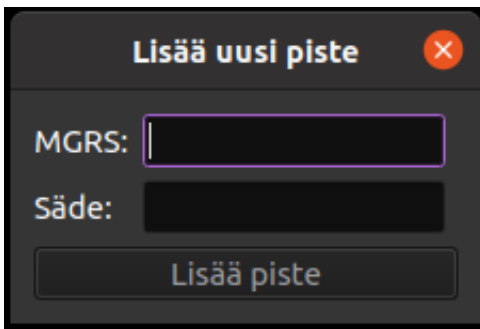
Pisteistä muodostuvan polun on tarkoitus näyttää kartalla suurin piirtein samalta kuin kameranäkymässä, eli sillä on punainen vasen laita, vihreä oikea laita ja sininen keskiviiva. Kameranäkymästä poiketen karttanäkymässä tulee näkyä kerralla koko polku. Lisäksi itse pisteiden tulee olla

selkeästi merkittyinä ja numeroituina järjestyksessä, jotta käyttäjä voi tarvittaessa säätää juuri haluamaansa pistettä. Polun ulkomuoto on edelleen nähtävissä kuviossa 8.

Laykkan toiminnan kannalta on myös oleellista merkitä pisteitä saavutetuiksi sekä kartalla että pistelistalla sitä mukaa kun robotti käy niitä läpi. Näin pistelistalta nähdään suoraan, mitkä pisteet on jo käyty läpi ja mitä pistettä kohti robotti pyrkii seuraavaksi. Ainoastaan vielä saavuttamattomat pisteet ovat muokattavissa.



Kuvio 8 Karttanäkymä sekä karttapistevälilehti



Kuvio 9 Uuden pisteen lisääminen karttapistelistan kautta

Suurimmaksi haasteeksi karttapolun luomisessa osoittautui polun laitojen luominen erityisesti pisteiden kohdalla, eli kun polku muuttaa suuntaa. Jokaisen pisteen ympärille piti trigonometriaa hyödyntäen laskea kaksi pistettä, joiden kautta polun laitojen tulisi kulkea, jotta polun leveys säilyisi oikeana vääristymättä. Lisäksi polun laitojen värit piti saada säilytettyä oikein päin, eli ajosuuntaan nähden oikea laita polun asennosta riippumatta aina vihreänä ja vasen vastaavasti punaisena. Tämä ratkaistiin pitämällä kirjaa siitä, kumpi edellä lasketuista pisteistä on vasen ja kumpi oikea, mikä puolestaan onnistui ottamalla polun menosuunta huomioon laskutoimituksissa: lasketaan ensin oikean, eli vihreän, laidan pisteen sijainti ja kirjataan se oikean laidan pisteiden listalle – punaisen vasemman laidan piste on tästä aina 180 astetta polun toisella puolella, ja se kirjataan omalle listalleen erikseen. Lopuksi yksinkertaisesti yhdistetään näiden erillisten listojen pisteet viivoilla laitojen piirtämiseksi, kukin omalla värillään.

Tällä menetelmällä polku kuitenkin muodostuu pelkistä suorista pätkistä terävillä kulmilla eli polku ei käänny sulavasti missään kohtaa. Tämä tarkoittaa myös sitä, että luvussa 5.1 esitelty polun kaartuminen kameranäkymässä jää toistaiseksi hyödyntämättä. Toimeksiantaja onkin ilmaissut polun sulavan kaartumisen kuuluvan pikemminkin jatkokehityksen puolelle.

5.3 Maalipisteet kameranäkymään

Pisteiden, joiden kautta polku karttanäkymässä kulkee, on tarkoitus samalla toimia yksittäisten polun pätkien maaleina, joiden kautta Laykka pyrkii ajamaan polkua pitkin. Kyseiset ”ruokinta-” tai ”maalipisteet” liittyvät vahvasti tekoälyn kouluttamiseen ja toimivat eräänlaisina maamerkkeinä myös operaattorille. Tätä varten pisteet oli saatava itse polun tavoin näkyviin kameranäkymään.

Kameranäkymässä on tarkoitus näkyä kerrallaan ainoastaan polun sen hetkisen pätkän maalipiste, jolla Laykka kulloinkin on.

Ulkomuodon puolesta pisteen haluttiin muuttavan kokoaan etäisyyden mukaan fyysisen kappaleen tavoin. Pisteen tuli olla selkeä ja sen säteen tuli olla näkyvissä pisteen ympärille piirrettynä kehänä. Kehä muodosti haasteen, sillä jos kehän piirtää suoraan QPainter-kirjaston ellipsityökalulla, siltä jää puuttumaan perspektiivi. Lisäksi kehän siirtely ja säätäminen polun mukana koitui liian haasteelliseksi pelkkänä ellipsinä, sillä ellipsi käyttää pelkkää korkeus- ja leveysarvoa, eikä siis muodostu polun tavoin matriisikerronnalla muunnettavissa olevista matemaattista koordinaattipisteistä.

Ratkaisu haasteeseen oli omien yksinkertaisten algoritmien kehittäminen. Kuviossa 10 esitelty ”draw_circle” -funktio ottaa vastaan ympyrän keskipisteen koordinaatit, ympyrän etäisyyden tasosta, ympyrän säteen ja haluttujen pisteiden lukumäärän (mitä enemmän pisteitä, sitä tasaisempi ympyrän kehä). Funktiosta saadaan ulos lista ympyrän muodostavista pisteistä, jotka voidaan lopuksi yhdistää viivoilla kehäksi toisella algoritmilla, joka on nähtävissä kuviossa 11. Näin saadaan lopputulokseksi kehä, jolle voidaan lisätä perspektiivi ja jonka pisteitä voidaan luomisvaiheessa siirrellä matriiseilla varsinaisen polun lailla. Tämä lopputulos on esiteltynä kuviossa 12.

Omien algoritmien kehittäminen ratkaisi useammankin ongelman, sillä QPainter-kirjaston ellipsityökalulla luodun kehän tarkka haluttu koko oli haasteellista saada täsmäämään halutun säteen kanssa. Uudelle funktiolle annetaan suoraan kehän säde, joka käyttää samaa matemaattista koordinaatistoa (ja täten mittakaavaa) kuin itse polku, eli ympyrän halkaisija saadaan suoraan täsmäämään polun leveyden kanssa. Lisähyötynä on myös se, että tätä erillistä funktiota voidaan hyödyntää jatkossa muun muassa erilaisten esteiden merkitsemiseen robotille.

Kuten virtuaaliajopolun myös pisteen tulee pysyä paikoillaan ympäröivään maastoon nähden, eli pistettä siirrellään yhdessä polun kanssa. Polun keinohorisonttimaiset liikkeet periytyvätkin suoraan pisteelle, mutta pisteen varsinaisen sijainnin (etäisyyden ja suunnan) on oltava irrallaan polusta, sillä pisteen sijainti ei ole polkuun sidoksissa. Polun tapaan pisteelle tarvitaan jo pelkkää kehitystyötä varten sellaiset asetukset, joilla pisteen ominaisuuksia ja sijaintia voidaan säätää. Nämä asetukset (ks. kuvio 13) löytyvät samalta välilehdeltä kuin polun asetuksetkin.

```

def draw_circle(self, center_x, center_y, height, radius, num_points):
    step = 360/num_points
    current_angle = 0

    circle_points = []

    for i in range(num_points):
        x = radius * math.sin(math.radians(current_angle))
        y = radius * math.cos(math.radians(current_angle))
        current_angle = current_angle + step
        circle_points.append([[center_x + x], [height], [-center_y-y]])

    return circle_points

```

Kuvio 10 Funktio kehän pisteiden laskemiseksi

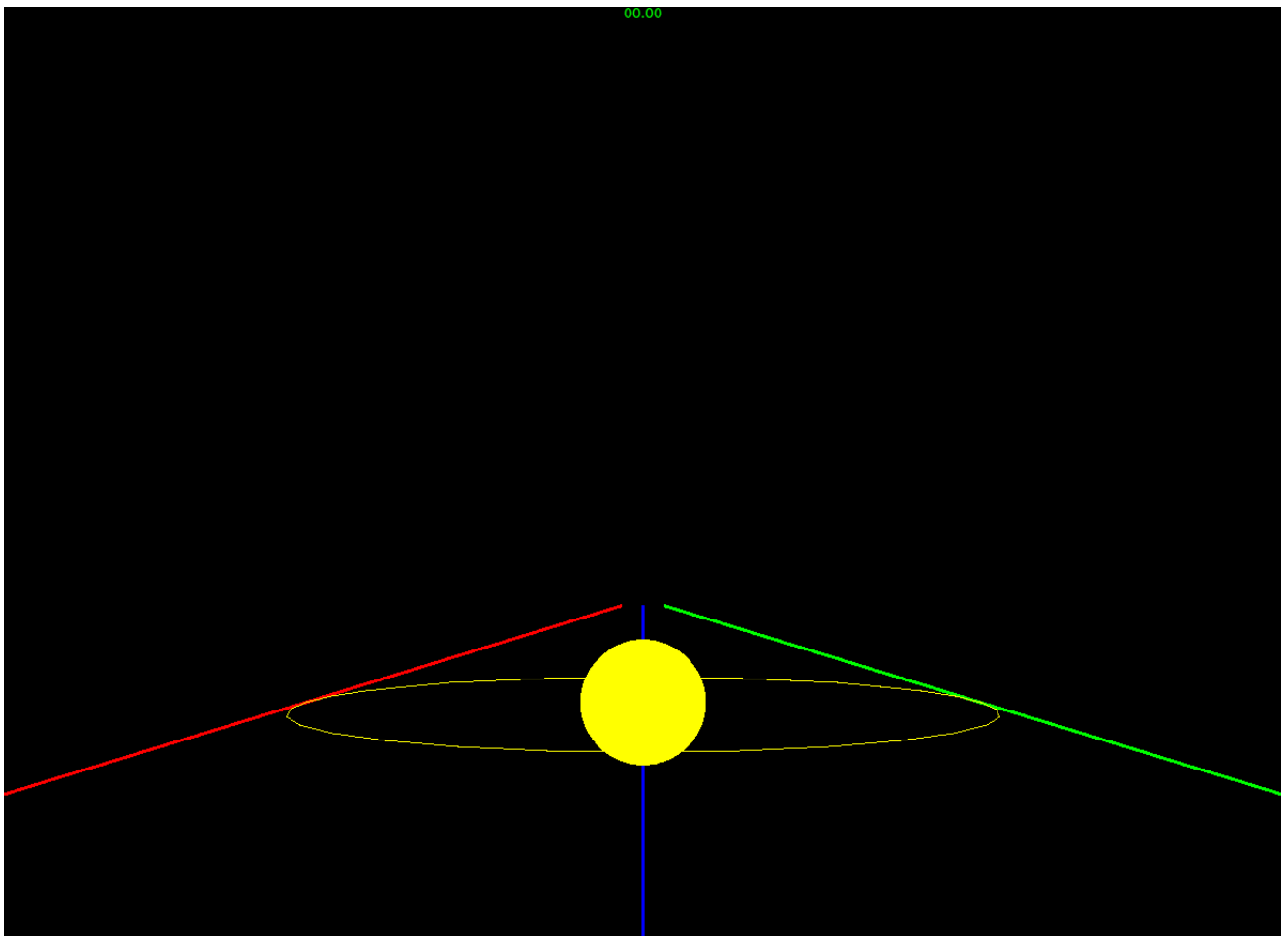
```

p1 = 0
first_p = 0
list_circle_points = []
for circle_point in projected_points_circle:
    p2 = QPointF(circle_point[0], circle_point[1])
    if p1 != 0:
        painter.drawLine(p1, p2)
    else:
        first_p = p2
    if circle_point == projected_points_circle[-1]:
        painter.drawLine(first_p, p2)
    p1 = QPointF(circle_point[0], circle_point[1])
    list_circle_points.append(circle_point[0])
    list_circle_points.append(circle_point[1])

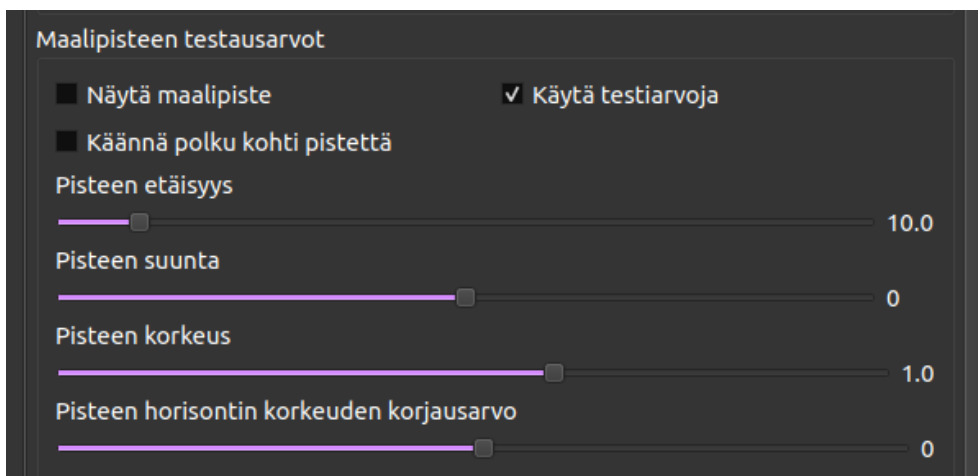
self.output_points.append(list_circle_points)

```

Kuvio 11 Algoritmi kehän 3D-projisoitujen pisteiden yhdistämiseksi viivoilla



Kuvio 12 Maalipiste ajopolkunäkymässä



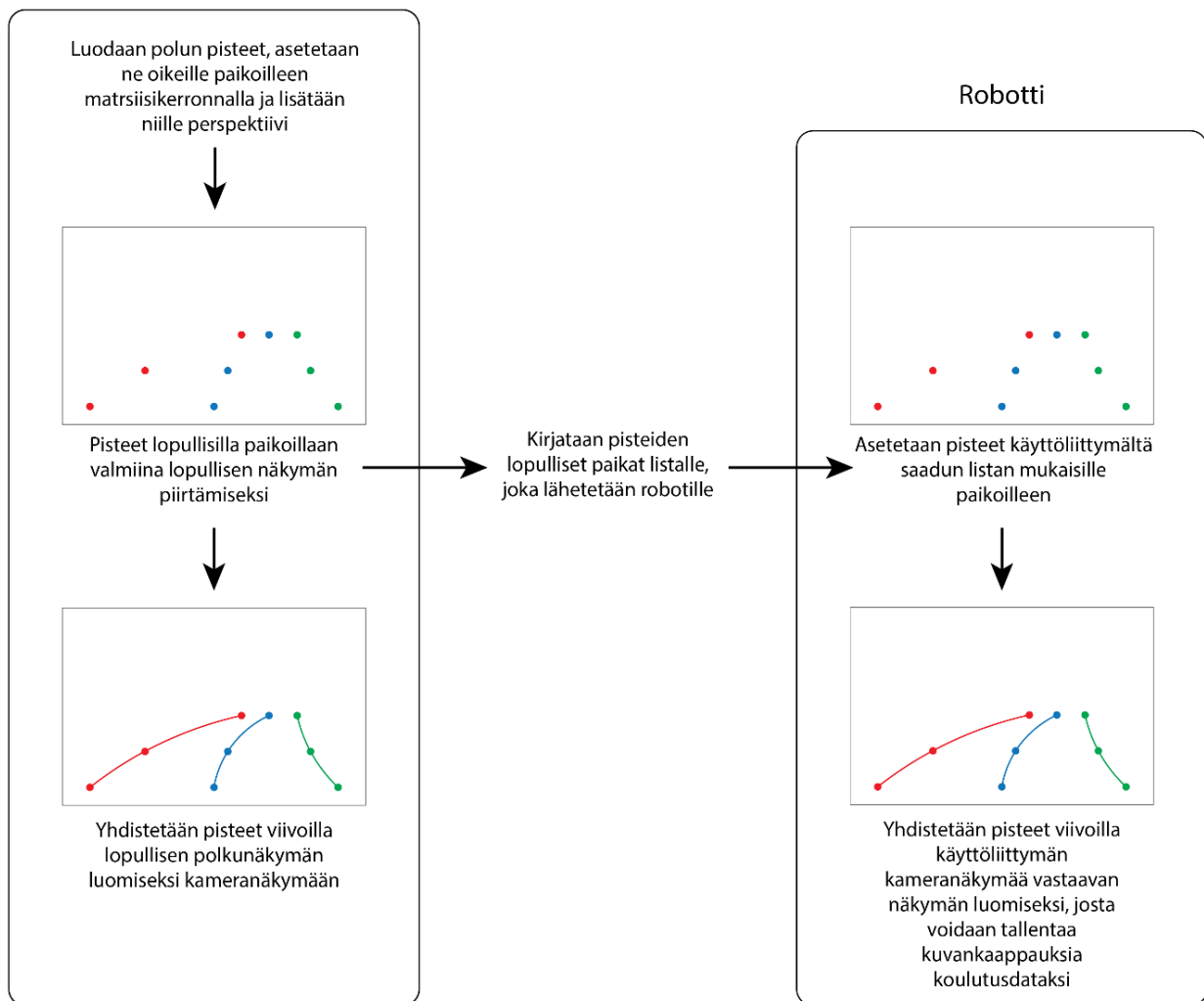
Kuvio 13 Maalipisteen asetukset

5.4 Polun siirtäminen robotille

Polkunäkymä tarvitaan myös itse robotin päähän, jotta sitä voidaan hyödyntää tekoälyn kouluttamiseen ja ohjaamiseen. Haastavaa tässä on se, että tekoälyä varten tarvitaan kokonaisia kuvia, joissa ajopolku näkyy. Kuvien lähettäminen robotille jatkuvana virtana on kuitenkin suhteellisen raskasta, joten lähetettävän datan minimoimiseksi haluttiin kehittää ratkaisu, joka ei vaatisi kokonaisten kuvien lähettämistä robotille.

Polun lähettäminen päätettiin suorittaa kirjaamalla käyttöliittymän kameranäkymän leveys ja korkeus sekä piirretyn virtuaaliajopolun pisteiden koordinaatit (eli polun varsinaiseen piirtämiseen käytetyt kameranäkymän pikseleiden X- ja Y-arvot) listaan, joka lähetetään Laykkalle. Tämän listan arvojen perusteella voidaan Laykkan päässä piirtää polku uudestaan sellaisena kuin se käyttöliittymässä näkyy, ilman että polun piirtämiseen vaadittujen pisteiden X- ja Y-arvojen laskemista tarvitsee suorittaa uudestaan robotin päässä. Robotti siis ainoastaan suorittaa polun luomisprosessin aivan viimeisen askeleen, eli varsinaisen polun piirtämisen, yhdistämällä jo valmiiksi lasketut (eli listalle kirjatut ja robotille lähetetyt) pisteet polun viivoilla lopullisen polkunäkymän luomiseksi. Tämä prosessi on pyritty havainnollistamaan tekijän piirroksella kuviossa 14. Kamerakuvaa ei tarvitse erikseen lähettää robotille, sillä käyttöliittymän kamerakuva tulee itse robotilta, eli polku voidaan robotin päässä piirtää suoraan robotin kamerasta saatavan kuvan päälle. Robotin päässä piirretystä lopputuloksesta voidaan robotille tallentaa jatkuvana sarjana kuvia, jotka voidaan siirtää edelleen Donkey Carille tekoälyn hyödynnettäviksi.

Päätelaitteen graafinen käyttöliittymä



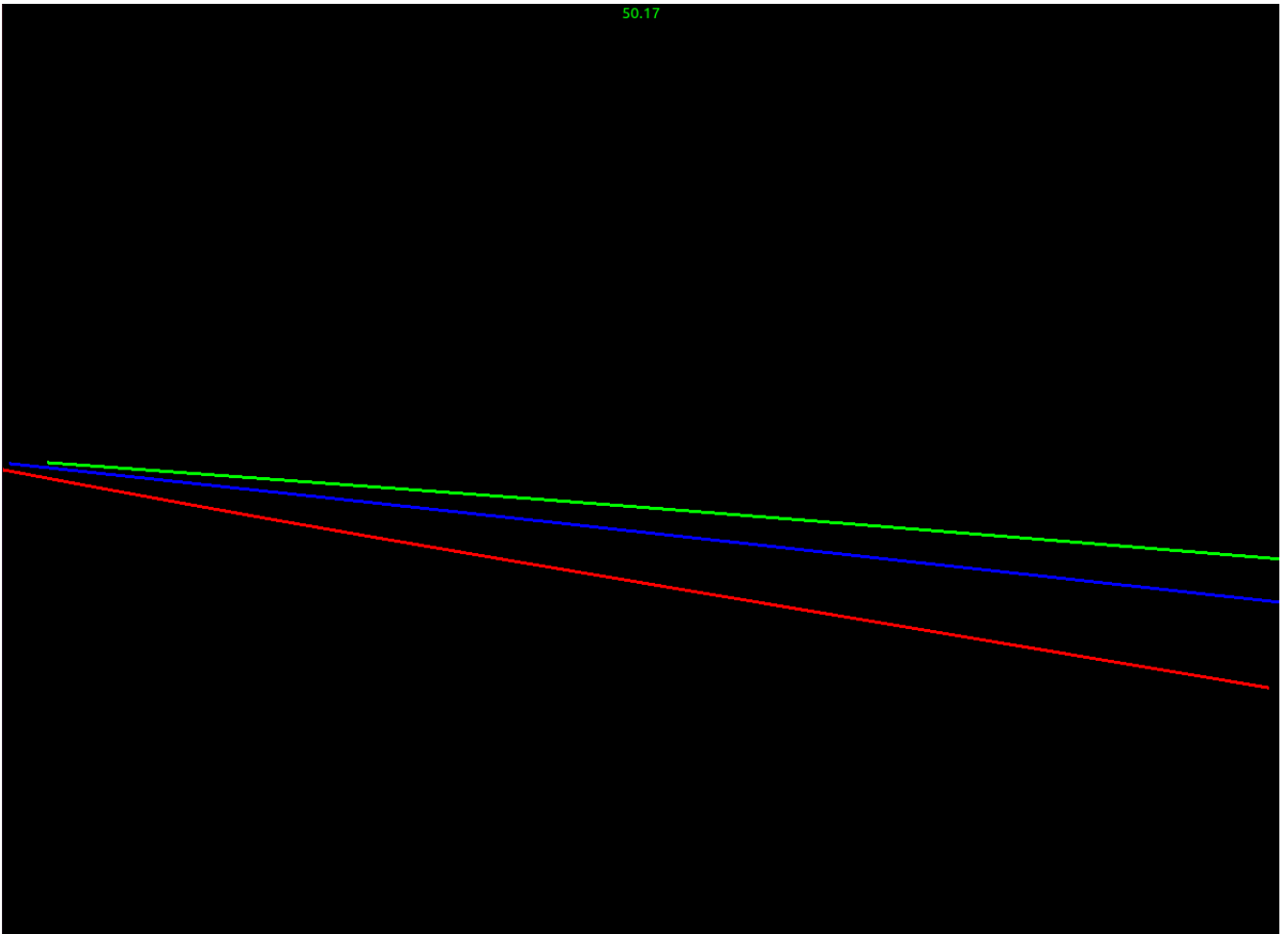
Kuvio 14 Polkunäkymän luominen robotin päähän havainnollistettuna

5.5 Koordinaatit ja kokonaisuuden toiminta

Kaikki edellä mainitut osa-alueet ovat osa samaa kokonaisuutta, jonka eri osat toimivat yhdessä. Keskeisimpänä osana kokonaisuuden toiminnassa ovat koordinaatit ja robotin kompassisuunta, joihin koko ajopolun varsinainen toiminta perustuu. Robotin suunnan ja sijainnin sekä seuraavien maalipisteiden koordinaattien tulee olla tiedossa, jotta polku voidaan kääntää oikeaan suuntaan. Lisäksi robotin etäisyyden maalipisteestä tulee olla tiedossa, jotta voidaan määrittää, onko robotti saavuttanut maalipisteen vai ei. Polun pisteiden koordinaatit tulevat karttapalvelimelta ja käyttöliittymän karttapistelista, jotka keskustelelevat keskenään. Listan ja kartan on keskusteltava kes-

kenään, jotta niiden sisältö täsmäisi, eli molemmilta löytyisivät samat pisteet samoilla koordinaateilla ja säteillä. Jos esimerkiksi pistelistalta poistetaan piste tai muutetaan pisteen koordinaattia, on muutos välitettävä myös kartalle. Vastaavasti jos karttaan lisätään piste, tulee pisteen myös ilmestyä pistelistalle.

Käytännössä koordinaatteja tarvitaan siis sekä polun pisteitä että itse robottia varten. Seuraavan maalipisteen suunta ja etäisyys voidaan laskea, kun tiedetään sekä pisteen että robotin sijainnit. Kun tiedetään etäisyys, voidaan määrittää, onko robotti saavuttanut maalipisteen vai ei. Jos robotin etäisyys pisteestä on pienempi kuin pisteen säde, voidaan piste julistaa saavutetuksi, ja robotti voi jatkaa matkaa mahdolliselle seuraavalle pisteelle. Lisäksi koordinaattien avulla voidaan laskea Laykkan etäisyys polulta. Tämän lasketun etäisyyden perusteella voidaan siirtää polkua siten, että se pysyy alkuperäisellä paikallaan maastoon nähden. Näin robotin on siis mahdollista ajaa polulta ulos ja myös tarkastella sitä sivusta, eli polku saadaan säilymään robotista irrallisena (ks. kuvio 15).



Kuvio 15 Kameranäkymä polun ulkopuolelta

Koordinaattien muodon osalta toimeksiantajan toiveena oli käyttää yleisesti sotilaskäytössä olevaa MGRS (Military Grid Reference System) -koordinaatistoa, jossa koordinaatit annetaan yhtenä tekstipätkänä, johon sisältyvät karttaruutu ja sen sisällä olevat pisteen koordinaatit. Esimerkiksi Santa-haminan pääportin koordinaatti on MGRS-muodossa *35V LG 9175 70408*. Käytännön syistä itse koodissa käytetään ensisijaisesti erillisiä pituus- ja leveysasteita, jotka on käännetty MGRS-muotoon pelkästään käyttöliittymää varten. Vastaavasti kun käyttäjä syöttää uuden pisteen MGRS-koordinaatit, muunnetaan tämä koodin puolella tarvittaessa pituus- ja leveysasteiksi. Koordinaattien muuntaminen on toteutettu "mgrs"-kirjastolla, joka testien perusteella säilyttää koordinaatin riittävän tarkkana muunnoksen jälkeen (heittoa enintään noin metrin verran). Jos koordinaatin muuntaa jatkuvasti yhä uudelleen, heittoa alkaa tosin kertyä sen verran, että käyttäjä saattaa joutua korjaamaan pisteen sijaintia. Käytännössä jatkuvaa muuntamista ei kuitenkaan pitäisi tapahtua, eikä se tapahtuessaan ole suuri ongelma, joten tämän työn kannalta toiminta on etenkin paremman kirjaston puutteessa riittävän hyvä.

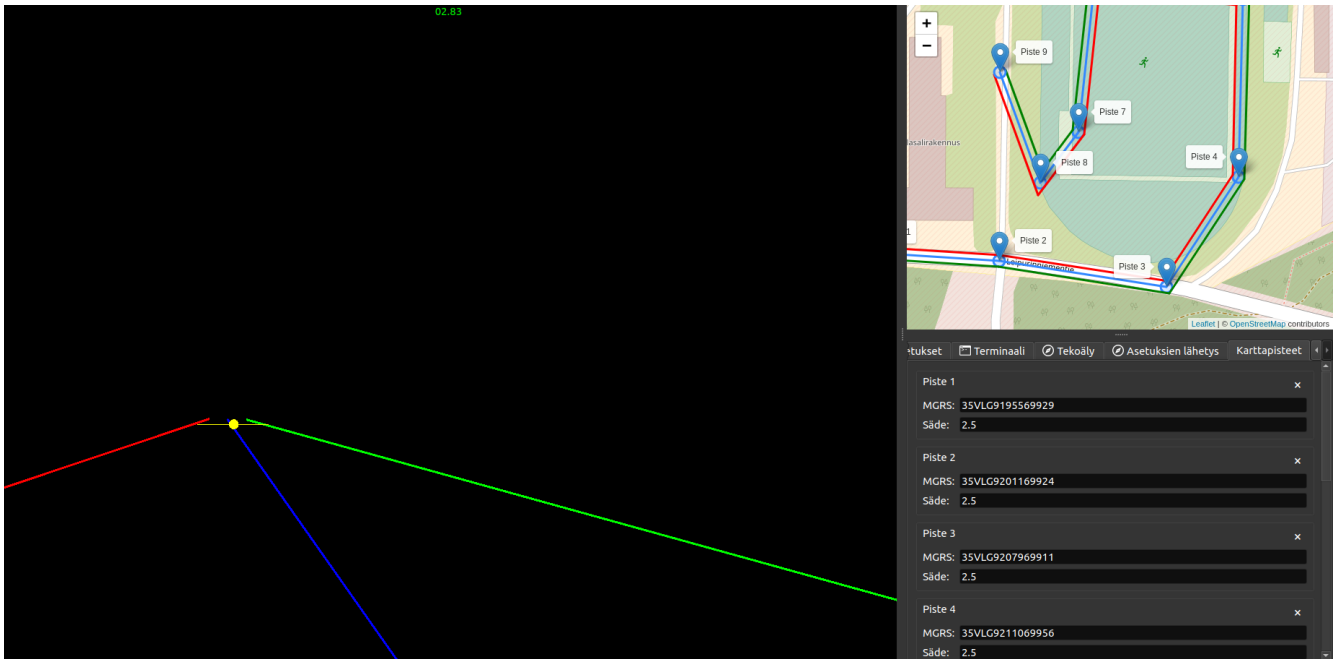
6 Tulokset

6.1 Esittely

Työn keskeisimpinä tavoitteina oli saada

- ajopolku lisättyä kameranäkymään
- polun ulkomuodoksi 3D-projektio, jossa polulla on selkeät laidat ja keskiviiva
- polun asento ja ominaisuudet säädettäviksi
- polku kaartumaan käännöksissä
- polku piirrettyä karttanäkymään, mukaan lukien laidat sekä keskiviiva
- polku kulkemaan kartalle määriteltyjen pisteiden kautta
- polku kiemurtelemaan loivin käännöksin pisteiden kautta
- pisteet määriteltäviksi klikkaamalla karttaa
- pisteet määriteltäviksi antamalla valmis koordinaatti
- pisteille säädettävä säde
- pisteille erillinen välilehti, jolta käsin niiden ominaisuuksia voi muokata
- pisteet siirtymään raahaamalla
- karttapisteet ajopolun lailla piirretyiksi säteineen kameranäkymään
- polun leveys säätymään karttapisteiden säteiden mukaan
- luotua edellytykset koulutusdatan keräämiselle virtuaaliajopolkunäkymästä robotin päähän
- pisteet merkittyä saavutetuiksi sitä mukaa kun robotti ajaa niitä läpi
- koordinaatit MGRS-muotoon.

Tavoitteista saavutettiin yhtä lukuun ottamatta kaikki. Saavuttamatta jäänyt tavoite on polun kiemurtelu loivin käännöksin pisteiden kautta, minkä todettiin kuuluvan pikemminkin jatkokehityksen puolelle. Muiden tavoitteiden osalta lopputulos on pitkälti toiveiden mukainen. Kokonaisuus on nähtävissä kuviossa 16. Kuviossa vasemmalla nähtävän polkunäkymän musta tausta johtuu yksinkertaisesti siitä, ettei robotti ole käynnissä, eikä siis lähetä videokuvaa päätelaitteelle.



Kuvio 16 Lopputuotoksen kokonaisuus käyttöliittymässä

Ajopolku on piirretty kameranäkymään toiveiden mukaisena 3D-projektiona, jota voidaan siirrellä sekä pysty että vaakasuunnassa ja kääntää X-, Y- ja Z-akselien ympäri. Polun asentoa ja ominaisuuksia voidaan säätää ja korjata manuaalisesti polun asetusvälilehdeltä. Myös polun kaartumista voidaan säätää tarvittaessa kääntyvän tien lailla. Polku pysyy vaakatasossa ympäröivään maastoon nähden kolmiulotteisen keinohorisontin tavoin ja johtaa aina kohti seuraavaa pistettä. Jos polun seuraava piste on polun ensimmäinen, piirtyy polku suoraan Laykkalta kohti pistettä. Jos seuraava piste puolestaan ei ole ensimmäinen, eli polku piirtyy Laykkasta irrallisena kahden pisteen välille, polku siirretään ja käännetään asettumaan näiden pisteiden välille Laykkan kameranäkymässä.

Polun reitin voi käyttäjä määrittää joko suoraan karttanäkymässä tai pistelistan kautta. Karttanäkymässä pisteitä lisätään klikkaamalla kartassa kohtaa, jonka kautta polun halutaan kulkevan. Pisteitä voi siirtää kartalla *raahaa ja pudota* -menetelmällä ja yksittäisiä pisteitä voidaan poistaa niitä klikkaamalla. Polku piirtyy karttanäkymään näiden pisteiden välille, säilyttäen leveytensä tasaisena sekä laitojen värit oikeina (vasen punainen, oikea vihreä). Karttapistelistan kautta pisteitä voi lisätä klikkaamalla "Lisää piste"-painiketta, joka avaa ikkunan (ks. kuvio 8 luvussa 5.2), johon syötetään uuden pisteen haluttu MGRS-koordinaatti sekä säde. Polun leveys mukautuu kartalla pisteiden säteiden mukaan, ja kameranäkymän polun leveys säätyy sen mukaan, kuinka leveä polku milloinkin

on siinä kohtaa polkua, jolla Laykka kullakin hetkellä on. Pisteiden koordinaatteja ja säteitä voidaan muokata tarvittaessa pistelistan kautta, jolloin karttanäkymän polku mukautuu muutoksiin. Pisteitä voidaan poistaa myös pistelistan kautta klikkaamalla valkoista ruksia halutun pisteen kohdalla. Pistelistan kautta voidaan myös poistaa koko polku kerralla.

Polun seuraava piste näkyy kameranäkymässä keltaisena pallona, jonka ympärillä on pisteen sädettä vastaava kehä. Piste piirtyy sen hetkisen polun osan loppupisteen kohdalle ja muuttaa kokoaan etäisyyden mukaan luoden vaikutelman fyysisestä kappaleesta. Kun Laykka saavuttaa pisteen kehän, eli saapuu pisteen sädettä vastaavalle etäisyydelle pisteestä, piste julistetaan saavutetuksi sekä karttanäkymään että pistelistalle ja polku ohjataan näyttämään kohti mahdollista seuraavaa pistettä.

Lisäksi polun lähettäminen robotille onnistuttiin toteuttamaan ilman, että robotille tarvitsi lähettää kokonaisia kuvia kameranäkymästä. Tämä toteutettiin lähettämällä robotille käytännössä polunpiirto-ohjeet, eli kameranäkymän leveys- ja korkeusarvot sekä kameranäkymässä olevat polun piirtopisteiden sijainnit, jotka yhdistämällä viivoilla saatiin piirrettyä lopullinen polkunäkymä. Maalipisteen lähettäminen on toteutettu samalla periaatteella ja sisältyy samaan robotille lähetettävään pakettiin kuin itse polku. Tästä robotin päähän luodusta näkymästä voidaan jatkossa tallentaa kuvankaappauksia tekoälyn koulutusdataksi. Polun piirtäminen robotin päässä toteutettiin erikseen sekä Qt- että OpenCV-kirjastoilla, jotta jatkossa voi valita, kummalla kirjastolla koulutusdatan tuottaminen halutaan toteuttaa.

Toimeksiantajan palaute työstä on ollut kiitettävää, ja tuloksiin sekä niiden pohjalta Laykka-AMPGV-tutkimusprojektin yleiseen etenemiseen on oltu tyytyväisiä. Virtuaaliajopolku on erittäin oleellinen osa Laykkan suunnitellun autonomian tutkimusta ja käytännön kehitystä ja jää siis aktiiviseen käyttöön ja jatkokehitykseen välttämättömänä osana tutkimusprojektin kokonaisuutta.

Työ on toteutettu enimmäkseen työharjoittelun aikana Maanpuolustuskorkeakoulun Sotateknikan laitoksella, mutta työn tekijä on toimeksiantajan toiveesta osallistunut projektin kehittämiseen myös harjoittelujakson jälkeen muun muassa sotaharjoituksessa Pudasjärvellä ja on lisäksi kutsuttu toiselle harjoittelujaksolle jatkamaan projektin parissa. Tulosten laatua ja jatkokehitystä pohditaan tarkemmin luvussa 7.

6.2 Testaus

Suurin osa työstä toteutettiin työharjoittelussa Maanpuolustuskorkeakoulun sotatekniikan laitoksella, jolloin testaaminen suoritettiin paikan päällä työn ohessa. Muutamien suunnitelmallisten testien lisäksi projektissa oli mahdollisuus tehdä vapaasti testejä tarpeen mukaan miten ja koska tahansa. Virallisemmissa testikerroissa ajopolkua käsiteltiin lähinnä ohimennen, muun muassa tarkistamalla millä asetuksilla polun saisi parhaiten vastaamaan maaston asentoa Laykkan kameranäkymässä. Polku oli näiden testien aikana vielä niin keskeneräinen, ettei varsinaisia kenttäkokeita olisi voitukaan vielä suorittaa. Varsinaisen lopputuloksen testaamisen estävänä tekijänä puolestaan on ollut työstä riippumaton Laykkan GPS:n toimimattomuus.

Polun toimintaa koordinaattien kanssa oli tarkoitus testata vuoden 2021 joulukuussa järjestetyssä sotaharjoituksessa Pudasjärvellä. Laykkan GPS:ää ei kuitenkaan edelleenkään saatu toimimaan, joten polun testaus jäi tekemättä. Sotaharjoituksen aikana itse polkua saatiin kuitenkin kehitettyä eteenpäin.

Helmikuuhun 2022 mennessä Laykkaan oli saatu uusi GPS-moduuli ja Pudasjärvellä testaamatta jääneitä toimintoja pyrittiin jälleen testaamaan Santahaminassa. Uuden GPS:n sekä IMU:n integrointi Laykkaan osoittautui kuitenkin toistaiseksi puutteelliseksi, eli polun toiminnot jäivät jälleen testaamatta. Uuden GPS:n toiminta saatiin kuitenkin varmistettua ja ongelmakohtat tunnistettua.

Huhtikuussa 2022 aloitettiin uusi työharjoittelujakso, jonka aikana otettiin käyttöön erillinen testipenkkialusta, jossa on kiinni muun muassa Raspberry Pi -pienoistietokone sekä GPS- ja IMU-moduulit. Tällä alustalla Laykkan GPS- ja IMU-moduuleita voitiin testata ilman robottia. Alustan idea on vastata täysin Laykkan toimintaa GPS- ja IMU-arvojen lähettämisen osalta. Pi saa arvoja siihen kytketyiltä GPS- ja IMU-moduuleilta, jotka se lähettää Laykkan tavoin päätelaitteelle käyttöliittymään. Virtuaaliajopolun toiminta perustuu Laykkan GPS-sijaintiin sekä IMU:ltä saataviin kallis- ja kompassiarvoihin. Pi:llä, johon kyseiset moduulit ovat kytkettyinä, voidaan siis täysin simuloida Laykkan ajoa virtuaaliajopolun kannalta oleellisten toimintojen kannalta. Tämä helpottaa testaamista suuresti, sillä itse robotin ajo valmisteluineen on huomattavasti työläämpi toteuttaa kuin yksinkertaisesti Pi:n kanssa käveleminen. Kameran videokuva ei ole polun varsinaisen toiminnan kannalta oleellinen, joten sen puuttuminen Pi:ltä ei vaikuta polun testaamiseen.

7 Pohdinta

Tavoitteena oli toteuttaa virtuaaliajopolku, joka näkyy robotin kameranäkymässä ja jonka reitin ja leveyden käyttäjä voi itse määrittää. Polun tuli käyttäytyä aidon fyysisen polun tavoin robotin kameranäkymässä, mukautuen maaston sekä robotin asennon ja sijainnin mukaan. Polun reitti tuli voida määrittää joko karttanäkymään piirtämällä tai suoraan antamalla koordinaattipisteet, joiden kautta polun halutaan kulkevan. Kameranäkymään haluttiin näkyviin myös maalipisteet, joiden kautta polku kulkee ja joiden sädettä säätämällä voidaan määrittää polun leveys. Lisäksi polkunäkymästä oli tavoitteena voida tuottaa koulutusdataa robotin päähän kuvankaappauksina.

Työn tulokset olivat pitkälti tavoitteiden mukaiset ja toimeksiantajan palaute on ollut todella myönteistä. Polun ulkoasu sekä käyttäytyminen vastaavat toiveita ja kaikki polun luomiseen ja säätämiseen liittyvät oleelliset ominaisuudet saatiin toteutettua onnistuneesti. Polun kaartumisen karttanäkymässä osoittautui kuitenkin ominaisuutena toistaiseksi liian haasteelliseksi toteuttaa, etenkin suhteessa siitä saatavaan lähinnä ulkonäölliseen hyötyyn. Suurin haaste varsinaisen kaartumisen määrittämisen ohella olisi kirjan pitäminen siitä, missä tämä kaartuva polku tarkalleen kulkee, eli millä logiikalla kameranäkymän polku saataisiin vastaamaan täysin kartan polun kaarretta. Tämä ei ole ongelma, jos polku muodostuu toteutuksen lailla pisteiden välisistä suorista pätkistä, sillä Laykkan sijainti suhteessa kahden pisteen väliseen suoraan on helppo laskea. Tämä kuitenkin tarkoittaa myös sitä, että kameranäkymän polun kaarto-ominaisuutta ei toistaiseksi ole voitu hyödyntää. Ominaisuus on kuitenkin olemassa ja hyödynnettävissä, mikäli polun kaartuminen saadaan jatkossa toteutettua. Toimeksiantaja onkin todennut polun kaartumisen olevan pikemminkin jatkokehittämisen kohde, minkä vuoksi sitä ei vielä tässä työssä ole täysin toteutettu.

Laykka-AMPGV on prototyyppi ja tutkimusalusta, jota kehitetään aktiivisesti. Tästä johtuen on ymmärrettävää, että kaikki ominaisuudet eivät vielä toimi moitteettomasti. Laykka ei ole valmis paketti, jonka olemassa oleviin toimintoihin perustuvia uusia ominaisuuksia voidaan vain yksinkertaisesti lisätä ja odottaa asioiden toimivan. Työn osalta tästä seurannut suurin rajoitus oli Laykkan GPS:n sekä IMU:n kompassin toimimattomuus työn aikana. Niiden toimintaan vahvasti nojautuvan virtuaaliajopolun testaamista ei ehditty toteuttaa itse robotilla.

Testaamisen vähäisyyden vuoksi on todennäköistä, että toteutukseen sisältyy vielä haasteita, jotka ilmaantuvat vasta myöhempien testien aikana. Työhön sisältyy myös ominaisuuksia, joiden hiominen muutenkin vaatii enemmän testaamista, muun muassa polun mittasuhteiden määrittäminen vastaamaan oikeaa maailmaa kameranäkymässä esimerkiksi polun leveyden ja näennäisten etäisyyksien osalta. Työ on kuitenkin testejä vaille valmis. Kun onnistuneet testaukset on saatu suoritettua, voidaan siirtyä tekoälyn kouluttamiseen ja käytännön testaamiseen.

Työ on hyvin oleellinen osa Laykkan tekoälyn kehitystä, tutkimusta ja toimintaa ja jää siis aktiiviseen käyttöön osana projektia, jonka kehitys jatkuu edelleen. Työ on ollut laaja, ja siihen on sisällynyt monta eri osa-aluetta. Työtä voidaan hyödyntää pohjana Laykkan muille myöhemmille, muun muassa kameranäkymää hyödyntäville tekoälyn toiminnoille, kuten esteiden merkitsemiselle polun maali pisteitä vastaavalla tavalla. Lisäksi tämä työ käytännössä tarjoaa yhden mahdollisen toteutuskeinon autonomisen ajon ohjaamiseen.

Työn jatkokehittämiseen sisältyy todennäköisesti polun ulkoasun hiomista niin kamera- kuin karttanäkymässä ja mahdollisten myöhemmin ilmaantuvien ongelmien ratkomista tai korjaamista. Laykka-AMPGV itsessään kehittyy projektina ja uusia ideoita tai tarpeita voi ajan mittaan ilmaantua myös polkuun liittyen.

Työn tekijä on toteutuksen aikana syventänyt osaamistaan Pythonin ja erityisesti Qt-kirjaston osalta. Tekijä on jo aiemmin ollut mukana Laykkan graafisen käyttöliittymän kehittämisessä, minkä johdosta tekijällä oli jo aiempaa kokemusta sekä Pythonista että Qt-kirjastosta. Tekijälle täysin uusia osa-alueita työn kannalta olivat 3D-projektion luominen sekä karttapolun määrittäminen ja piirtäminen Leaflet Vue:lla. Jonkin verran kokemusta kertyi myös OpenCV-kirjaston käytöstä. Lisäksi tekijälle on kertynyt kokemusta Laykkan varsinaisesta kehittämisestä ja myös työskentelestä maanpuolustusosalalla. Työn tekijä jatkaa Laykkan kehitystä mahdollisuuksien mukaan ja on jo ollut ajoittain mukana Laykkan kehitystyössä myös työharjoittelun ulkopuolella.

Lähteet

Abidi, S. 2020. Building an interactive map with Vue and Leaflet. Artikkele Leaflet Vue:n käytöstä interaktiivisen kartan luomiseen LogRocket-sivustolla. Viitattu 16.3.2022. <https://blog.logrocket.com/building-an-interactive-map-with-vue-and-leaflet/>.

About. N.d. Esittelysivu OpenCV:n kotisivulla. Viitattu 14.4.2022. <https://opencv.org/about/>.

About OpenStreetMap. 2022. Esittely OpenStreetMap-karttapalvelun wikisivustolla. Viitattu 6.4.2022. https://wiki.openstreetmap.org/wiki/About_OpenStreetMap.

About Qt. 2022. Esittely Qt:n wikisivustolla. Viitattu 6.4.2022. https://wiki.qt.io/About_Qt.

An Introduction to GitHub. 2020. Esittelyartikkeli GitHub-palvelusta Yhdysvaltojen hallituksen sivustolla. Viitattu 4.4.2022. <https://digital.gov/resources/an-introduction-github/>.

Andersson, C. 2021. The unmanned ground vehicles to be used in future military operations. Tiede ja Ase No79 s.85–106. Viitattu 14.4.2022.

Andersson, C. 2022. Laykka-AMPGV:n inkrementaalinen kehitysprosessi runkoversio X.2:sta X.3:een sekä kehityksen seuranta kenttätesteillä ja -kokeella. Pro Gradu, Maanpuolustuskorkeakoulu. Viitattu 14.4.2022.

Extension Marketplace. N.d. Laajennusten esittelysivu Visual Studio Coden dokumentaationsivustolla. Viitattu 19.3.2022. <https://code.visualstudio.com/docs/editor/extension-marketplace>.

Getting Started. N.d. Aloitusohjeet Visual Studio Coden dokumentaationsivulla. Viitattu 19.3.2022. <https://code.visualstudio.com/docs>.

Hardesty, L. 2017. Explained: Neural networks. Artikkele MIT:n uutissivustolla. Viitattu 18.3.2022. <https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>.

IBM Cloud Education, 2020. Artificial Intelligence (AI). Artikkele IBM:n sivustolla. Viitattu 18.3.2022. <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>.

Kreuzhuber, K. 2020. How to convert a remote controlled car into a Donkey Car. Blogikirjoitus Ars Electronica -sivustolla. Viitattu 1.4.2022. <https://ars.electronica.art/aeblog/en/2020/01/21/donkey-training/>.

Salonen, I. 2022. Data ui aseisiin. Tekniikan Maailman artikkeli. Viitattu 21.4.2022. <https://tekniikanmaailma.fi/lehti/8b-2022/data-ui-aseisiin/>.

User Manual. 2022. Gitin käyttöohjeet Gitin kotisivulla. Viitattu 12.4.2022. <https://git-scm.com/docs/user-manual>.

What is Python? Executive Summary. N.d. Yhteenveto Python-ohjelmointikielen esittelystä Pythonin dokumentaationsivustolla. Viitattu 7.4.2022. <https://www.python.org/doc/essays/blurb/>.