

2017

# TrainGame 2017

**SUUNNITTELUDOKUMENTTI**

SEVERI PELTONEN MILAN MÄKIPÄÄ JUUSO  
NUIKKA

## Johdanto

TrainGame on Ohjelmistojen suunnittelu-kurssin harjoitustyö. TrainGame on junapeli, jossa käyttäjä ohjaa omaa huoltojunaansa VR:n rautateillä. Yhteyskatkoksen vuoksi vain huoltojunaan saadaan tieto radalla olevista esteistä ja junat jatkavat matkaansa niistä tietämättä, kunnes osuvat niihin. Junien osuessa esteisiin junat vahingoittuvat. Pelissä on tarkoitus parantaa asiakkaiden mielipidettä eli kerryttää "famea" poistaen esteitä ja väistellen muita junia. Esteitä keräämällä saa rahaa, jolla voi korjata tai ostaa uusia junia.

Tämä suunnitteludokumentti selventää pelin tärkeimmät komponentit ja toiminnan. Tämä suunnitteludokumentti on tarkoitus esittää assistenttitapaamisessa ja käyttää sopimuksena kehittäjien välillä peliin toteutettavista asioista.

## Yleiskuvaus

Peli toteutetaan vain pc:lle erillisenä sovelluksena ja sitä voi pelata Windows-, Linux- ja OS X-käyttöjärjestelmissä.

Ohjelma jaetaan kolmeen osakokonaisuuteen MVC-mallin mukaan. Pelin rakenteen tärkeimmät komponentit ovat käyttöliittymä, kontrolleri ja logiikka-osa. Tällä saadaan erotettua käyttöliittymä ohjelman logiikasta. Käyttöliittymä toteutetaan Qt:n widgeteillä ja logiikka C++-kielellä.

Kontrolleri toimii välikätenä pääikkunan ja logiikan välillä. Logiikan puolella on useita luokkia, jotka sisältävät pelin toiminnallisuuden kuten pelaajan, kaupan, rata- ja asemaverkkoon liittyvät osat. Logiikkaluokat ottavat kontrollerilta vastaan käyttäjän tekemiä toimintoja käyttöliittymässä ja sen mukaan muuttavat tilaansa, käyttöliittymään tehtävät muutokset välitetään tämän jälkeen logiikasta käyttöliittymään.

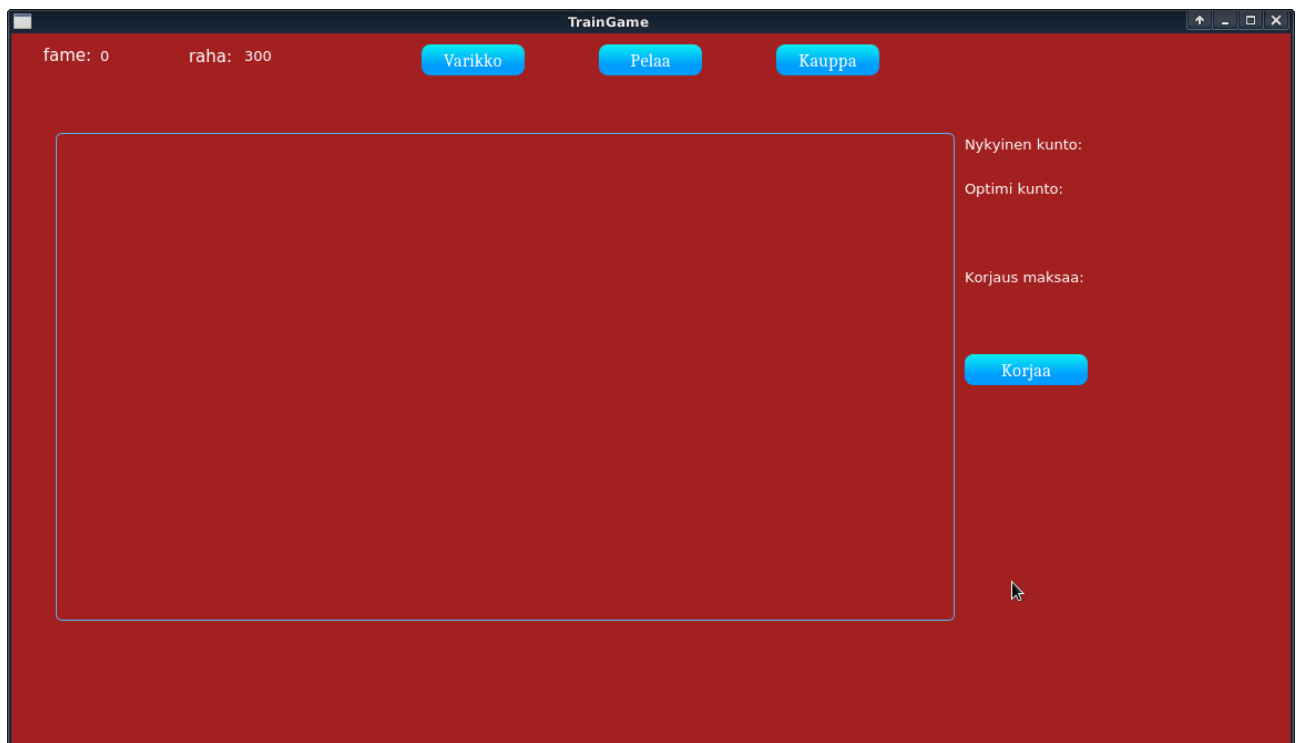
Pelin muut junat liikkuvat rataverkossa reaaliaikaisesti. Niihin liittyvät aikataulu- ja reittitiedot haetaan VR:n verkkopalvelusta. Sovellus vaatii näin ollen toimivan internet-yhteyden pelin käynnistymisen yhteydessä.

Peliä voi ohjata widgetejä klikkailemalla.

## Käyttöliittymä

Käyttöliittymä on jaettu kolmeen eri välilehteen, joiden välillä liikutaan käyttöliittymän yläosassa sijaitsevilla napeilla. Eri välilehdillä pelaaja voi ostaa tai myydä erilaisia junia, korjata hajonneita juniaan sekä pelata omistamillaan junilla. Kaikissa välilehdissä pelaaja näkee sen hetkisen raha- ja famemääränsä käyttöliittymän vasemmassa yläkulmassa.

Peli aukeaa käynnistettäessä varikko-välilehdelle. (kuva 1). Siellä pelaaja voi korjata omistamiaan junia..



Kuva 1: Varikko-näkymä

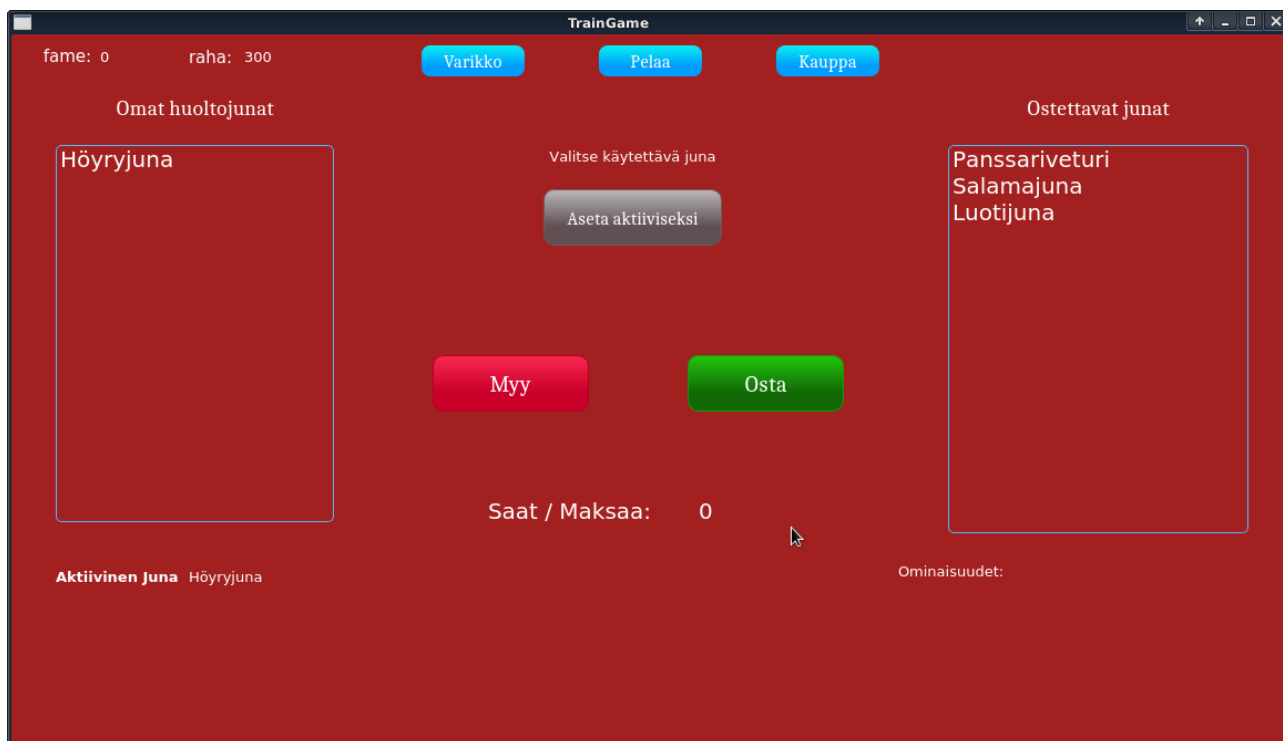
Pela -välilehti on esitelty kuvassa 2. Keskellä näkymää on alue jossa pelaajan juna kulkee raidetta pitkin. Vasemmalla on karttakuva Suomen rataverkosta, josta pelaaja näkee nykyisen sijaintinsa punaisena pisteenä ja esteen sijainnin vihreänä. Ikkunan oikeassa reunassa näkyvät suunnanvalintaan liittyvät listat, joista pelaaja voi valita jatkoyhteyden seuraavalta tai edelliseltä asemalta. Pelaajan juna jatkaa kohdeasemalleen saapuessaan listasta valitun aseman suuntaan. Jos valintaa ei tehdä, jatkaa juna automaattisesti listan ensimmäisen alkion suuntaan.

Muita maininnanarvoisia kohtia käyttöliittymässä ovat vasemman yläkulman tieto edellisestä ja seuraavasta asemasta, oikean yläkulman tieto esteen sijainnista ja vaarallisuudesta, sekä navigaattori, joka ilmoittaa seuraavan aseman, jonka kautta kulkiessa pelaaja on oikealla reitillä etstittävää estettä kohden.



Kuva 2: Peli-näkymä

Painamalla “Kauppa”-painiketta pelaaja siirtyy kauppa -välilehdelle (Kuva 3), josta hän voi ansaitsemallaan rahallaan ostaa uusia junia ja myydä omistamiaan junia. Kuvassa vasemmalla näkyvät pelaajan omistamat junat ja oikealla junat, jotka voi ostaa kaupasta. Mikäli juna on pelaajan omistama, voi hän klikata junan nimeä ja tämän jälkeen painaa “Aseta aktiiviseksi” nappulaa, jolloin ko. juna asetetaan aktiiviseksi junaksi. Pelaa -välilehdellä pelaaja pelaa aktiivisella junalla.



Kuva 3: Kauppa-näkymä

## Pelin kulku

Pelin alussa pelaajalle annetaan yksi juna ja jonkin verran rahaa, joilla hän voi ostaa itselleen parempia junia tai korjata ensimmäistä junaansa tarpeen mukaan. Ilman junaa pelaaja ei voi pelata peliä. Alussa myös pelaajalla on myös jonkin verran famea.

Pelatessaan pelaaja liikkuu rataverkossa ja pyrkii poistamaan sinne ilmestyvät esteet. Esteet poistetaan ajamalla niiden lähelle ja painamalla välilyöntyä. Esteitä putsatessaan pelaajan fame kasvaa ja pelaaja saa työstään rahaa, jolla ostaa parempia junia. Törmäämällä esteeseen pelaaja saa rahallisen palkkion esteen poistamisesta, mutta ei famea. Rataverkossa vastaan voi tulla VR:n aikataulutietojen mukaan liikennöiviä junia, joihin pelaaja ei saa törmätä. Törmätessään pelaajan juna vahingoittuu ja pelaajan famea vähennetään. Mikäli koko juna hajoaa ei pelaaja voi enää pelata ko. junalla. Tällöin pelaajan on siirryttävä korjaamaan junansa, ostaa uuden, tai vaihtaa toiseen omistuksessa jo olevaan junaan. Jos VR:n juna päättyy samalle raiteelle esteen kanssa se tuhoutuu ja pelaajalta vähennetään rahaa ja famea.

Raiteiden risteämäkohdissa pelinäkömässä tulee vastaan asema, johon törmättäessä raide vaihtuu pelinäkömän oikeassa laidassa olevan suuntavalintalistan mukaiseksi.

## Osakokonaisuudet

Game, eli controller -luokka välittää käyttöliittymämuutokset pelilogiikalle.

RailLogic vastaa rataverkon sijaintitietojen ylläpidosta, radan piirtämisestä käyttöliittymään, sekä liikkumisesta rataverkossa.

PlayerLogic sisältää pelaajan ominaisuudet ja ylläpitää pelaajan junia.

Shop toimii kauppana, josta pelaaja voi ostaa junia. PlayerLogic luokka pitää omistajuuden shop-luokkaan

PlayerTrain kuvaa pelaajan junan ja sen ominaisuudet, näitä voi olla useita.

ObstacleLogic vastaa esteiden lisäämisestä käyttöliittymään ja esteiden hallinnasta.

ObstacleFactory luo mahdollisista esteluokista olioita obstacleLogicille.

dataReader-luokka lukee tarvittavat tiedot, kuten asema- ja ratadatan .json tiedostoista ja tallentaa ne edelleen tietorakenteeseen Game-luokkaan. Lukee myös erilaisten junien tiedot tiedostosta.

Boulder ja LeafPile kuvaavat esteiden grafiikkaoliot.

RailInterface kuvaa rajapinnan yksittäiselle radan pätkälle. Tästä luokasta periytetään OneSideRailTile.

PlayerTrain kuvaa yksittäistä junaa, jonka pelaaja voi ostaa kaupsta.

ObstacleInterface kuvaa rajapinnan yksittäiselle esteelle. Näin on helppo hyödyntää esim. Factory suunnitteluperiaatetta. Tästä luokasta periytetään Person, LeafPile ja Rocks esteet. Mahdollisesti myös muita esteitä.

Backgroundlogic vastaa taustan siirtämisestä ja hallinnasta.

Background kuvaa taustan grafiikkaolion.

Pathfinder vastaa reitin etsinnästä estettä kohti.

VrTrain kuvaa VR:n junaan liittyvän grafiikkaolion

VrTrainManager hallitsee VrTrain olioita.

Station kuvaa aseman grafiikkaolion.

## Ensimmäisen välinäytön tavoitesisältö

Ensimmäiseen välinäyttöön pyrimme saamaan valmiiksi tietorakenteen, johon luetaan asemadata ja ratadata. Myös kauppa ominaisuus pyritään saamaan valmiiksi. Junan liikuttaminen raiteella ja risteämäkohdissa toiselle raiteelle siirtyminen toteutetaan ensimmäiseen välinäyttöön mennessä, jos rataverkon tietorakenne saadaan toteutettua järkevässä aikataulussa.

## Ensimmäiseen välinäyttöön mennessä toteutuneet ominaisuudet

Saimme luettua rataverkon sopivaan tietorakenteeseen. Kauppa saatiin toteutettua. Samoin aktiivisen junan valinta. Junan liikkuminen rataverkossa toteutettiin valmiiksi. Junien korjaaminen on toteutettu, mutta junat eivät vastaanota vahinkoa törmätessään esteisiin.

## Viimeisen palautuksen tavoitesisältö

Viimeiseen palautukseen mennessä peliin on toteutettava VR:n sijaintitietojen perusteella liikkuvien junien lisäys. Esteiden luomiseen on lisättävä järkevä tapa. Peliin on lisättävä keskeytys, kun siirrytään kauppaan tai varikolle korjaamaan junaa.

## Viimeisen palautuksen toteutuneet ominaisuudet

Kaikki suunnitellut ominaisuudet saatiin toteutettua, sekä näiden lisäksi myös pelaajalle ohjeita antava navigaattori saatiin toimimaan.

## Keskeneräiset ominaisuudet

Kaikki suunnitellut ominaisuudet saatiin toteutettua viimeiseen palautukseen mennessä