

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 suussa 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 asematiedot. Logiikkaluokat ottavat kontrollerilta vastaan käyttäjän tekemiä toimintoja käyttöliittymässä ja sen mukaan muokkaa tietoa ja näyttää sitä käyttäjälle.

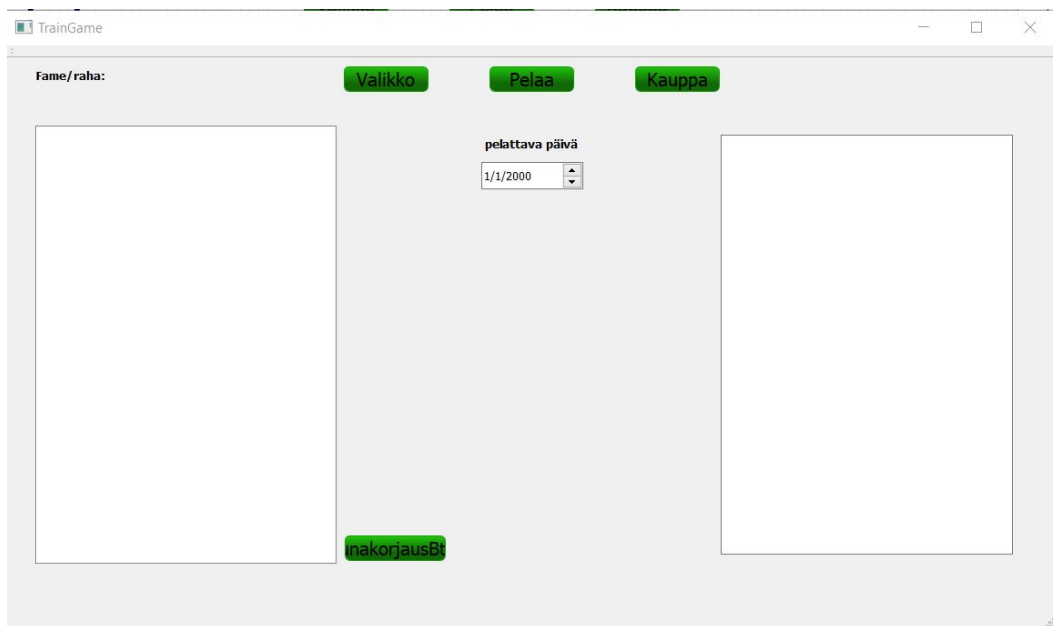
Pelin muut junat ovat reaaliaikaisia, ja niiden tiedot, kuten tämänhetkinen raide, sekä x- ja y-koordinaatit luetaan VR:n tietokannasta. Sovellus vaatii näin ollen toimivan internet-yhteyden.

Peliä voi ohjata joko widgetejä klikkaamalla tai näppäimistön avulla.

## Käyttöliittymä

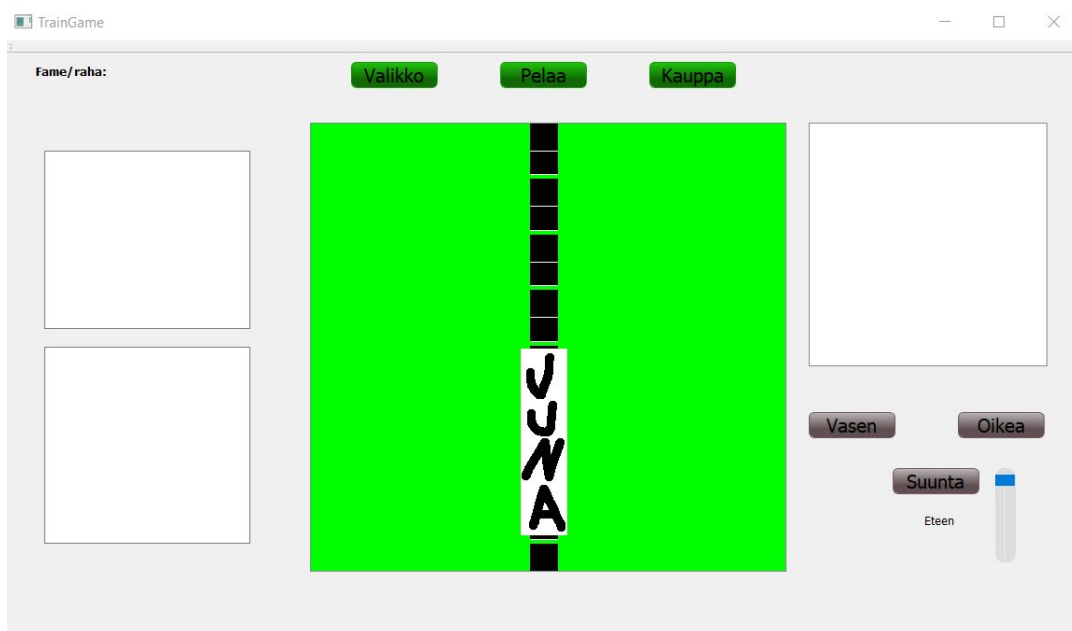
Käyttöliittymä on jaettu kolmeen eri osaan, joiden välillä liikutaan sovelluksen yläosassa sijaitsevilla napeilla. Eri käyttöliittymän osissa pelaaja voi ostaa tai myydä erilaisia junia, korjata hajonneita juniaan sekä pelata omistamillaan junilla. Kaikissa välilehdissä pelaaja näkee sen hetkisen raha-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

Peli aukeaa käynnistettäessä valikko-näkymään. (kuva 1). Valikko-näkymässä voi pelaaja korjata juniaan ja asettaa rata.digitraffic.fi-APIsta haettavan päivän.



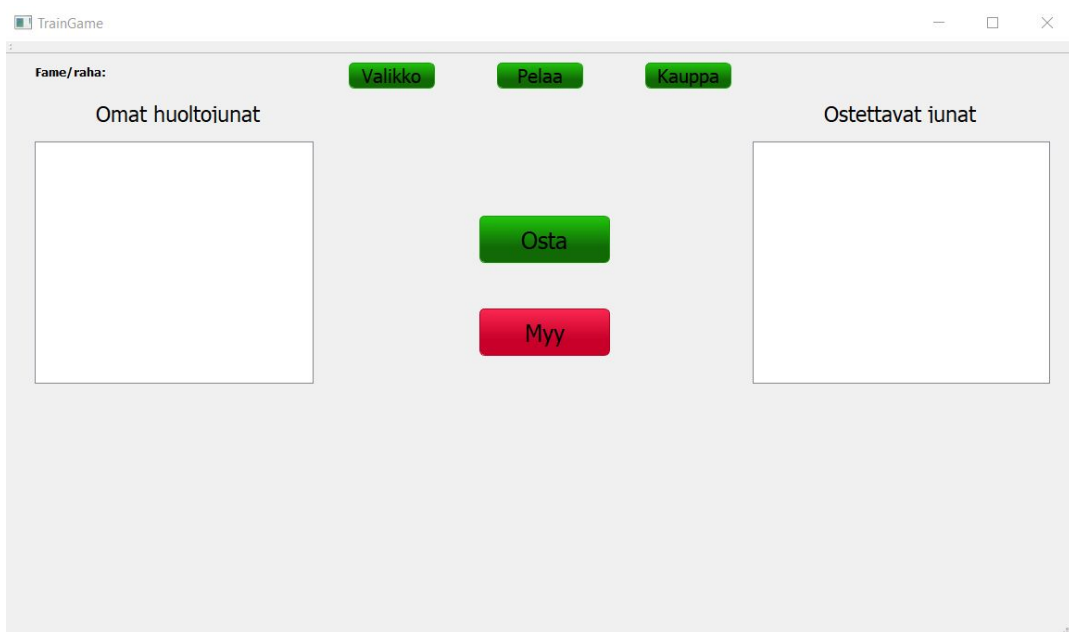
Kuva 1: Valikko-näkymä

Sen jälkeen kun pelaaja on ostanut junan itselleen, hän voi siirtyä “Pela”-painikkeella peliin. Peli-näkymä on kuvassa 2. Keskellä näkymää on alue jossa pelaajan juna kulkee raidetta pitkin. Oikealla puolella ylhäällä on alue jossa näkyy Suomen rautatiet, ja niillä olevat esteet. Alaalla oikealla on kaasun ja suunnan vaihtamiseen nappulat. Vasemmalla ylhäällä tekstiboksi johon tulee lista tämän hetkisistä esteistä. Alhaalla vasemmalla on infoboksi johon tuodaan infoa esimerkiksi läheisistä junista.



Kuva 2: Peli-näkymä

Painamalla “Kauppa”-painiketta pelaaja siirtyy kauppa-näkymään (Kuva 3) josta hän voi keräämällä rahallaan ostaa uusia junia ja myydä nykyisiä.



Kuva 3: Kauppa-näkymä

## Pelin kulku

Pelin alussa pelaajalla on x määrä rahaa, joilla hän voi ostaa itselleen junan. Ilman junaa pelaaja ei voi pelata peliä. Alussa myös pelaajalla on täysi karma maan asukkaisiin.

Pelatessa pelaaja liikkuu rataverkossa ja pyrkii putsaamaan eteen tulevat esteet. Esteitä putsatessaan pelaajan fame kasvaa ja saa työstään rahaa, jolla voi ostaa parempia junia. Rataverkossa vastaan voi tulla liikennöiviä junia, joihin pelaaja ei saa törmätä. Törmätessään pelaajan juna hajoaa osittain. Mikäli koko juna hajoaa ei pelaaja voi enää pelata ko. junalla. Tällöin pelaajan on siirryttävä korjaamaan junansa, ostaa uusi, tai vaihtaa toiseen omistuksessa jo olevaan junaan.

Raiteiden risteämäkohdissa raide haarautuu myös pelinäköymässä. Pelaajan juna valitsee risteyskohdissa suuntavalinnan mukaisen raiteen.

## 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.

**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**.

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

**RailInterface** kuvaa rajapinnan yksittäiselle radan pätkälle. Tästä luokasta periytetään **OneSideRailTile** ja **TwoSideRailTile**

**TrainInterface** kuvaa rajapinnan yksittäiselle junalle. Tästä luokasta periytetään **PlayerTrain** ja **AiTrain**. **AiTrain** kuvaa VR:n apista luettavaa junaa

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

## **Ensimmäisen välinäytön sisältö tavoite**

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.