

# Toteutusdokumentti

---

## Ohjelman yleisrakenne

Ohjelma muodostuu kuudesta paketista. Paketit ovat nimetty selkeästi, jotta on helposti pääteltävissä, mitä luokkia kukin paketti sisältää.

## Peli

### pacman

Paketti pacman sisältää vain Pelin aloittavan Main luokan.

### pacman.alusta

pacman.alusta paketti sisältää pelialustaan olennaisesti liittyvät luokat; Peliruudun ja Pelialustan. Pelialusta on yksinkertaisuudessaan kaksiulotteinen taulukko, joka muodostuu peliruuduista. Jokainen peliruutu tietää muun muassa omat koordinaattinsa, tiedon siitä onko ruudussa haamu tai Man, pistepallo tai ekstrapallo ja tiedon siitä mitä tyyppiä ruutu on (seinä, käytävä).

Parhaiden reittien selvittämistä varten ruutu tietää myös etäisyysarvion maaliin ja lähtöruutuun sekä sen ruudun, josta ruutuun on siirrytty lyhimmissä reitissä. Pelialusta luokan tarkoitus on luoda pelialusta ja peliruudun tarkoitus on kertoa pelille tietoja itsestään. pacman.alusta paketti sisältää myös tekstitiedoston, johon on kirjattu pelialusta numeroin.

### pacman.gui

Pakkaus pacman.gui sisältää pelin graafiseen liittymään ja näppäimistön toimintaan liittyvät toiminnot. Käyttöliittymä luokka toteuttaa graafisen liittymän ja huolehtii mahdollisista virheilmoituksista. Näppäimistönkuuntelija tulkitsee näppäimistön käyttöä ja piirtoalusta piirtää pelitilanteen jokaisen pelin actionperformed metodin päätteeksi. Rajapinta päivitettävän avulla voidaan toteuttaa metodi päivitä, jota käyttää esimerkiksi piirtoalusta uudestaan piirtämisessä. Pakkaus sisältää myös kaikki olennaiset kuvat (haamut, Man, kirsikka) .png-tyyppisenä.

### pacman.hahmot

Pakkauksessa pacman.hahmot ovat kaikkien peliin liittyvien hahmojen luokat. Niin sanotusti ylimpänä on abstrakti luokka Hahmo, jonka kummatkin Man ja Haamu luokka toteuttavat. Luokkaan on kasattu kaikki metodit, joita kaikki haamut ja Man käyttävät. Man luokkaan on sisällytetty metodit, jotka käsittelevät pääosin Manin liikkumista pelialustalla. Man myös huolehtii itse omista elämästään.

Haamuja on pelissä neljä ja kaikki toimivat alustalla omalla tavallaan. Tämän takia tein jokaisesta haamusta oman luokkansa, joka perii ylemmän Haamu luokan. Yksittäisissä haamuluokissa, Red, Green, Cyan ja Magenta, huolehditaan haamujen erikoistarpeista. Esimerkiksi luokassa Magenta etsitään haamulle sopivaa ruutua maaliksi ja luokassa Green toteutetaan kokonaisuudessaan haamun liikkuminen, koska tämä haamu ei käytä muiden tavoin hakualgoritmia, vaan alkuperäisessä pelissä toteuttamaani liikkumistapaa.

Pakkaus pacman.hahmot sisältää myös enum luokan Suunta, jonka avulla on toteutettu suunnat ylös, alas, oikealle ja vasemmalle helpommalla tavalla, joka myös siistii koodia.

## pacman.peli

Pakkaus pacman.peli sisältää luokat, jotka käsittelevät pelin logiikkaa ja peliin olennaisesti liittyvät komponentit, pistelaskuri ja highscore. Pacman luokka sisältää kaikki metodit, jotka koskevat pelin logiikkaa ja HaamujenKasittelija huolehtii kaikesta haamujen liikuttamiseen liittyvästä toiminnasta. Pistelaskuri luokka huolehtii pelaajan pisteiden laskemisesta ja säilyttämisestä ja Highscore luokka huolehtii pelaajan pisteiden mahdollisesta tallentamisesta ennätys tiedostoon.

## pacman.tietorakenteet

Pakkaus sisältää kaikki itse toteutetut tietorakenteet, järjestämisen sekä hakualgoritmin. Luokka Lista simuloi tavallisen listan toimintaa. Toteutettuja metodeja ovat esimerkiksi alkion hakeminen tietystä indeksistä, listaan lisääminen ja tarkistaminen kuuluuko annettu alkio listaan. Olen toteuttanut vain tarvitsemani perustoiminnot.

Järjestäjä luokka on toteutettu toimimaan lomituserjästyksen mukaisesti ja osaa järjestää yksiulotteisen taulukon johon on tallennettu peliruutuja. Luokka AStar toteuttaa hakualgoritmin astar. Hakualgoritmin avulla etsitään parhaat reitit haamuille liikua.

## Testit

Testit muodostuvat pakkauksista, jotka on nimetty samaan tapaan pelin pakkausten mukaisesti, mutta pakkausten nimen päätteeksi on testi (esim. pacman.alusta.test). Jokaisessa pakkauksessa on jokaista pakkauksen luokkaa vastaava testiluokka, jossa on pyritty testaamaan pelin koodiluokkaa mahdollisimman kattavasti.

## Suorituskykytestaus

Suorituskyky testauksessa on tarkasteltu kuinka kauan tärkeimpien osa-alueiden, kuten hakualgoritmin, järjestämisen ja pelin kokonaisvaltainen suorittaminen, kuluttaa aikaa.

## Haamut

Ensimmäiseksi tarkasteltavana on hakualgoritmin toiminta eri haamujen liikuttamisessa. Haamuista Red, Magenta ja Cyan käyttävät liikkumiseensa hakualgoritmia.

## Red

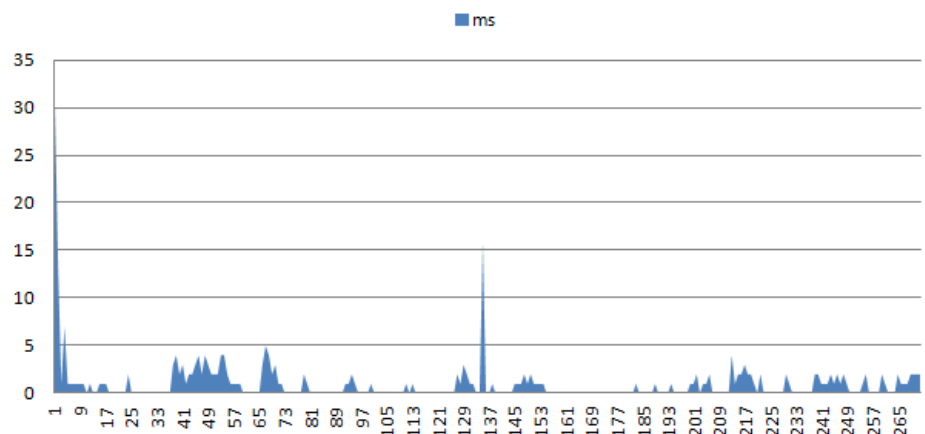
Ensimmäinen kaavio kuvaa kuinka paljon haamua Red liikuttava metodi, joka käyttää hakualgoritmia AStar, kuluttaa aikaa satunnaisella pelin pelaamiskerralla. Red liikkuu siten, että se jahtaa Mania kohteenaan Man itse.

Kaaviosta pystyy

havaitsemaan selkeää

heittelyä, jotka käyvät ilmi pelin kulun lähemmässä tarkastelussa. Alussa oleva korkea piikki selittyy sillä, että tässä vaiheessa punaisen haamun etäisyys Manista on todennäköisesti suurimmillaan.

Haamun Red liikkuminen



Suurempi selkeä heilahdus metodin suorituskerralla 38 selittyy sillä, että haamu heikennetään. Haamun maali ruutu, joka on tähän asti ollut Man itse, muuttuu. Heikennetyn punaisen haamun maali on ruutu, joka pelialustan vastakkaisella puolella peilattuna pystyakselin suhteen. Aikaa kuluu enemmän siten siksi, että uusi ruutu sijaitsee kauempana. Kun haamut ovat heikennettyjä, Man liikkuu nopeammin, joten saa otettua paremmin etäisyyttä haamuihin. Jo kohdassa 43 Red muuttuu takaisin vahvaksi, koska Man syö tämän. Haamu palaa alkuun, joten matka taas kohteeseen, eli Maniin, on sen verran pidempi, että aikaa kuluu liikuttamiseen. Tällainen noin 4ms kuluttava heilahtelu johtuu siitä, että haamu vaihtelee vahvasta heikoksi ja heikosta vahvaksi.

Suuri heilahdus kohdassa 127 johtuu siitä, että tässä vaiheessa haamu oli heikkona ja Man liikkui nopeasti kauas haamusta, joten seuraavan liikahtuksen selvittämiseen meni huomattavasti aikaa.

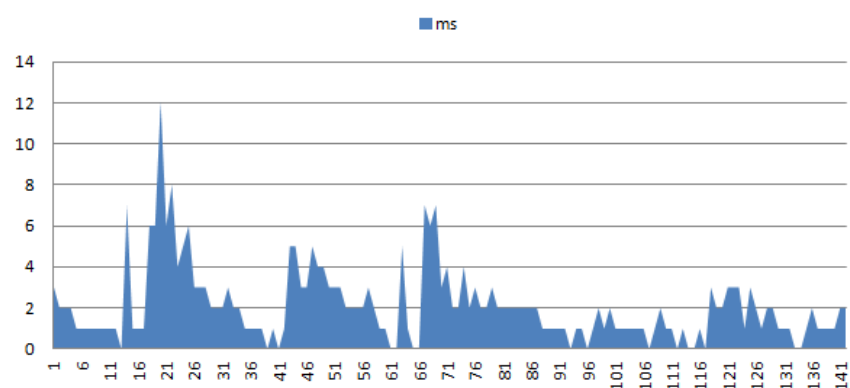
Pääosin kuitenkin haamun liikuttamismetodi toimii nopeudella 0ms tai 1ms. Tämä johtuu siitä, että pääosan pelistä haamu kulkee vain muutaman ruudun Manin perässä, joten lyhin reitti haamun ruudusta Manin ruutuun ei kuluta juurikaan aikaa.

## Magenta

Magenta liikkuu siten, että se arpoo alussa itselleen kohteen johon se pyrkii. Kun haamu saavuttaa maalinsa, se arpoo itselleen uudelleen maalin. Tällä tavoin haamu jatkaa koko pelin ajan, Magenta ei liiku erilailla ollessaan heikko.

Kaaviosta näkee selkeästi, että kun haamu arpoo itselleen uuden maalin ja liikahtaa ensimmäisen kerran aikaa kuluu eniten reitin selvittämiseen. Tämän jälkeen aika vähenee tasaisesti mitä lähemmäksi haamu pääsee kohdettaan. Aika on noin 0ms tai 1ms aivan maalin vieressä tai kun haamu saavuttaa maalin. Jos uusi maali, jonka haamu arpoo itselleen, on lähellä, ei ajassa näy niin suurta piikkiä.

Haamun Magenta liikkuminen

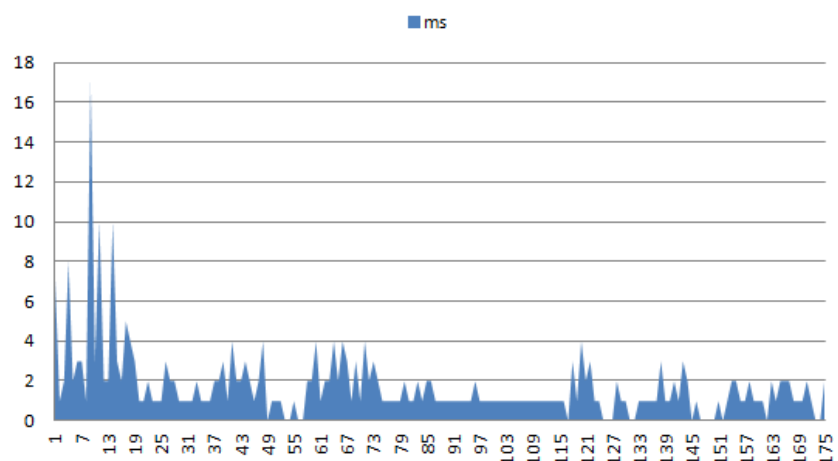


## Cyan

Cyan liikkuu siten, että sen kohteena on joko muutama ruutu Manin edellä oleva ruutu tai, jos tämä ruutu ei ole mahdollinen kohteena on Man itse.

Cyanin liikettä kuuvavasta kaaviosta voi huomata sen, että alussa aikaa haamun liikuttamiseen kuluu enemmän. Tämä johtune siitä, että haamu ei ole saanut Mania tässä vaiheessa vielä kiinni, joten on mahdollista, että Man voi niin

Haamun Cyan liikkuminen



sanotusti huijata haamua, siten että haamu säntäilee ympäriinsä yrittäen ennakoida Manin liikkeitä.

Myöhemmin, kun haamu on saanut Manin kiinni, on liike samantyyppistä Redin liikkeen kanssa. Tämä johtuu siitä, että Cyan samalla lailla ollessaan heikko, muuttaa suuntaansa kuten Red; alkuperäisen kohteensa peilikuvaksi vastakkaiselle puolelle alustaa. Jos haamun ollessa heikkona, haamu on siinä ruudussa, jonka kuuluisi olla sen kohde, kohde on tällöin Manin ruutu.

Erona kuitenkin Redin liikkumisen kaavioon on se, että yleisemmin aikaa kuluu noin 1ms, kun taas Red oli niin nopea, että yleensä kului aikaa vain 0ms. Tämä ero johtuu todennäköisesti siitä, että kohde ruutu on Cyanilla aina hieman kauempani kuin Redillä haamujen kulkiessa Manin perässä.

## **Puutteet & Parannus ehdotukset**

Peliä pystyy vielä kehittämään lisäämällä toiminnallisuuden, että sivuseinistä, samasta kohdasta, jossa alkuperäisessä pelissä, pystyisi pääsemään seinän läpi toiselle puolelle pelialustaa.