

# Miinaharava

## Työn aihe ja kuvaus:

Työn aiheena on peli, jota pystyy pelaamaan esim. Visual Studio Code:ssa sovelluksen terminaali -kentässä. Peli on idealtaan jo monelle tuttu miinaharava, jossa on tarkoituksena liputtaa miinoja kentän antamien arvojen perusteella.

## Työn ratkaisuperiaate:

Ratkaisuperiaatteena se, että ohjelma luo ensin kentän. Kentällä on "ruutuja" eli tiettyjä arvoja, joista pelaaja valitsee jonkin ruudun. Mikäli pelaaja liputtaa ruudun, ja ruutu on saanut backend -kentässä pommia vastaavan arvon (-1), pommi liputetaan. Mikäli taas pelaaja haluaa avata ruudun, joka on "pommi", peli päättyy. Pelin voittaa, kun saa liputettua kaikki pommit kentässä.

## Työn rakenne: miten se on jaettu funktioihin?

Työ on jaettu funktioihin sen perusteella, mitä operaatioita halutaan tehdä samanaikaisesti, sekä näyttääkö koodi siistiltä ja onko sitä helppo lukea. Esimerkiksi yhdellä funktiolla voidaan luoda kenttä, ja toisella vain kertoa ohjeet.

## Käytetyt funktiot ja niiden keskinäiset suhteet:

Funktiot:

`tyhjenna_terminaali()`:

Siisteyttä edistävä funktio, joka pelin pelaajan käyttöjärjestelmän huomioiden, tyhjentää terminaalin.

`kysy_koko()`:

Kertoo lyhyen esittelyn pelistä, ja kysyy käyttäjältä pelilaudan kokoa.

alusta\_listat\_ja\_matriisit():

Alustaa sekä mat\_back, että mat\_front matriisit sen mukaan, minkä kokoisen pelilaudan käyttäjä on valinnut funktiossa kysy\_koko().

Alustaa myös listat valmiiksi liputettujen ja avattujen ruutujen, sekä pommien koordinaateille.

asetta\_pommit():

Asettaa pommit satunnaisesti backend matriisiin. Pommien lukumäärän määrittelee funktio kysy\_koko().

numerointi():

Numeroi ruutuja sen mukaan moneenko pommiin, eli backend -matriisin arvoon -1 avattu ruutu on kosketuksissa. Käyttää pommien koordinaatteja, jotka määräytyvät funktiossa aseta\_pommit().

tulosta\_ruudukko() -> str:

Funktio muodostaa ruudukon, jota on helppo lukea ja yksinkertainen pelata. Ruudukko määräytyy sen mukaan, mitä pelaaja on syöttänyt funktioon kysy\_koko().

pelaajan\_valinta():

Funktio ensin kysyy pelaajalta, minkä toiminnon hän haluaa suorittaa, ja sen jälkeen sen ruudun koordinaatit, jossa tämä toiminto suoritetaan.

ruutujen\_muutos\_frontissa():

Tuottaa näkyvän kuvan terminaaliin. Reagoi funktioihin pelaajan\_valinta() ottamalla sieltä halutun toiminnon ja ruudun, sekä alusta\_listat\_ja\_matriisit() tarkistamalla mikä arvo kyseisellä valitulla ruudulla on. Mat\_back arvon perusteella tuottaa halutun ruudukon pelaajalle terminaaliin.

ohjeet\_str() -> str:

Tulostaa ohjeet pelin alkuun ja kertoo, mitä toimintoja on mahdollista käyttää edetäkseen pelissä.

voitto\_str() -> str:

Tulostaa tiedon pelin voitosta ja antaa vaihtoehdot mahdollista pelin jatkamista varten.

tappio\_str() -> str:

Antaa hieman henkistä tappiosta ja antaa vaihtoehdot toiminnoille uutta peliä varten.

tallenna\_pelin\_tiedot(nakyvat, liput):

Tallentaa pelaajan avaamien ja liputtamien ruutujen määrän tiedostoon "pelin\_tiedot.txt". Käyttää funktion alusta\_listat\_ja\_matriisit() listoja oikeiden arvojen saamiseen.

tulosta\_pelin\_tiedot() -> str:

Tulostaa pelin tiedot tiedostosta "pelin\_tiedot.txt".

kysy\_lopetus():

Funktio, joka kysyy pelaajalta tämän halukkuutta tarkastella omia tuloksia pelistä, ja/tai lopettaa pelaaminen. Vaihtoehtoina on k tai e.

## Mahdollisten ulkoisten kirjastojen käyttö:

random: satunnaisten arvojen generoimiseen

os: käyttöjärjestelmätason suorittamiseen

## Vastuualueiden jakautuminen:

Työn vastuualueita tai rooleja ei ole jaettu, sillä molemmat ryhmän jäsenistä ovat tehneet työtä hyvin pitkälti palvelukseensa omaa innokkuutta ja tietotaitoa. Lähtökohtana oli tehdä työ vain yhdessä. Tärkeimpänä se, että molemmat ymmärtävät kokonaisuuden, ja kun toinen tekee muutoksia silloin, kun toinen ei ole paikalla, ne ovat hyvä dokumentoida ohjelmaan itseensä esim. # -merkkejä käyttäen.

Alku meni käsi kädessä yhtä matkaa. Tämän jälkeen:

Funktioiden hahmottelu/suunnittelu: Eetu

Pelin kannalta merkittävimpien funktioiden toteuttaminen: Marius

Muita funktioita: Eetu ja Marius

Ohjelman läpikäynti ja testailu, vikojen korjaaminen ja pikkuasioiden parantelu: Marius ja Eetu

Dokumentointi: Eetu ja Marius

## Ohjeet ohjelman käyttämiseksi:

1. Suorita ohjelma terminaalissa
2. Syötä terminaaliin kentän koko (p/i)
3. Kohtaan 'toiminto:' joko "å" (poistu), "f" (liputa ruutu) tai "" (avaa ruutu)
4. Pyri etsimään ruutuja avaamalla mahdolliset miinat ja liputa ne. Ruudun numero kertoo, moneenko miinaan kyseinen ruutu on kosketuksissa.
5. Kun kaikki miinat on löydetty ---> kohtaan 'Pelitietojen tulostus:' joko "k" (kyllä) tai "e" (Ei).
6. Haluatko jatkaa pelaamista? Vastaa kohtaan 'Pelin lopetus:' joko "k" (kyllä) tai "e" (Ei).

## Chatgpt keskustelu:

Myös googlea itsessään on käytetty ahkerasti komentojen etsimiseen.

Chatgpt keskustelu:

<https://chat.openai.com/share/556805c3-c6f9-47d0-b1d3-e9bb33296c3f>