

Riskinolla – dokumentaatio

Aarni Kivelä & Jaakko Nurminen

Työnaihe ja kuvaus:

Riskinolla on Asteriski ry -henkinen kahden pelaajan ristinolla peli, jossa toinen pelaajista pelaa ainejärjestön tutulla nimikko ASCII-merkillä. Ohjelma on toteutettu Pythonista löytyvällä tkinter-kirjastolla, jonka avulla on mahdollistettu ohjelman graafisenkäyttöliittymä. Perinteisen pelaamisen lisäksi ohjelma tarjoaa mahdollisuuden seurata omaa pelihistoriaa erilliseltä ”Historia”-ikkunalta.

Työn ratkaisuperiaate:

Riskinollan ratkaisuperiaate pohjautuu listamatriisiin, jonne kirjataan pelaajien tekemät valinnat. Peli tarkastetetaan ensin iteroimalla läpi ristikon vaaka ja pystyriivit sekä lopuksi diagonaalit voittoyhdistelmät. Kaiken kaikkiaan mahdollisia voittoyhdistelmiä on siis vain 8. Työn historia-ominaisuus ratkaistiin ohjelman sisäisen sanakirjan sekä ulkoisen tekstitiedoston avulla. Kun ohjelma käynnistetään hakee se tekstitiedostosta aiemmat pelit sanakirjaansa ja tämän jälkeen kirjaa ne ohjelman historia-ikkunan tekstikenttään.

Työn rakenne:

Työ on jaettu kahdeksaan funktioon, jotka ovat seuraavanlaiset:

1. `click()`
Hallitsee pelilaudan klikkauksia ristikon ruutuihin, kutsuu `paattyiko_peli()` -funktia
2. `lopetus()`
Kertoo pelaajalle, että peli on ohi ja nostaa näytölle viestin pelaajan voitosta. Kutsuu `tallennus()`-funktia ja tallettaa pelitilanteen sanakirjaan muistiin.
3. `paattyiko_peli()`
Tarkistaa onko ristikossa voittoyhdistelmää ja kutsuu voiton yhteydessä `lopetus()`-funktia
4. `peli_historia()`
Luo pelihistoria-ikkunan ja kirjaa siinä olevaan tekstikenttään sanakirjan sisällön.
5. `nollaa_peli()`
Aloittaa uuden peli ja nollaa laudan nykyisen tilanteen.
6. `hae()`
Hakee tekstitiedostosta aiemman pelihistorian ja kirjaa sen valimuisti-sanakirjaan.
7. `tallennus()`
Tallentaa pelitilanteet sanakirjaan, jossa avaimena on voittoajankohta ja sisältönä pelin voittaja.
8. `luo_lauta()`
Luo graafisenkäyttöliittymän ja siinä olevat näppäimet.

Ulkoistenkirjastojen käyttö:

Ohjelmassa käytetään kahta ulkoista kirjastoa. Näistä toinen on aiemmin mainittu tkinter-kirjasto, joka on mahdollistanut ohjelman graafisen käyttöliittymän luonnin. Toinen kirjasto, jota työn toteutuksessa on hyödynnetty on datetime-kirjasto, jonka avulla pelihistorian ajankohdat voidaan kirjata.

Vastuualueiden jakautuminen:

Työ toteutettiin varsin yhdenvertaisesti. Valtaosan koodista kirjoitimme kahdestaan mieltien ongelmatilanteita yhdessä. Historia-ominaisuutta kirjoittaessa päätimme jakaa toteutuksen karkeasti front- ja backendiin. Jolloin toinen kirjoitti koodin, joka vastaa ohjelman historia-sivun luomisesta ja toinen koodin joka vastaa sen sisällöstä sen tuottamisesta.