JAVA - HARJOITUSTYÖ

11.10.2013

Yleistä :

Tämä dokumentti sisältää 2 otsaketta, joista ensimmäinen kuvaa Java - harjoitustyöni aiheen ja toinen teknisen ratkaisuni pääpiirteet. Kaaviot löytyvät erillisistä dokumenteista, samoin ohjelman käyttöohjeet ja tuntikirjanpito.

1. Aiheen kuvaus :

Ohjelma toteuttaa MasterMind pelin (Säännöt). Pelin tarkoituksena on arvata piilossa oleva koodi. Arvattavan koodin pituus on kiinteästi määritelty 4 merkin pituiseksi. Käyttäjä määrittelee koodiin sisältyvien eri merkkien määrän (2-10). Oletuksena on 6 merkkiä. Tekstiversiossa koodimerkit ovat väliltä 0 - 9. Myös graafisessa käyttöliittymässä käytetään samaa tapaa.

Pelin aluksi ohjelma arpoo koodin. Käyttäjä ei näe arvottua koodia ja koittaa arvata sen syöttämällä koodin näppäimistöltä. Graafisessa liittymässä tämä tapahtuu valitsemalla merkki alasvetovalikosta. Käyttäjän arvattua koodin tai arvauskertojen loputtua näytetään piilossa ollut koodi ja peli loppuu. Käyttäjällä on käytössään 10 arvauskertaa.

Käyttäjällä on mahdollisuus säätää tapaa, jolla ohjelma ilmoittaa arvauksen tuloksen :

- Helppo - pelitapa : Ohjelma ilmoittaa jokaisen arvatun merkin yhteydessä onko merkki oikeassa paikassa (punainen reunus) ja onko merkki mukana koodissa, mutta väärällä paikalla (valkoinen reunus).
- Haastava pelitapa : Ohjelma ei osoita oikeiden merkkien paikkaa vaan ilmoittaa ainoaistaan tuloksen. Esim “2 oikein ja 1 oikeaa merkkiä”

Peliin on toteutettu toiminto, jonka avulla pelitilanteen voi tallentaa tiedostoon. Näin peliä voi jatkaa myöhemmin lukemalla pelitilanne tiedostosta uudelleen ohjelman käyttöön. Pelin lataus on mahdollista vain ohjelman käynnistyken yhteydessä. Latauksen jälkeen pelitilanteen sisältävä tiedosto poistetaan levyltä.

Peliin on toteutettu kaksinpeli variaatio, jossa toinen pelaajista syöttää arvattavan koodin ja toinen pelaaja ratkaisee syötettyä koodia. Pelitapa sisältää pistelaskutoiminnon, jossa jokainen käytetty arvaus lisää vastapelaajan pistesaldoa.

2. Tekninen ratkaisu

Ohjelmaan on toteutettu tekstipohjainen käyttöliittymä testailutarkoitukseen ja graafinen liittymä varsinaisen harjoitustyön ilmiasuksi.

Olen pyrkinyt irroittamaan käyttöliittymän sovelluslogiikasta Komento - abstraktiluokasta perittyjen luokkien avulla. Ajatuksena on ollut, että jokainen Komento - aliluokan ilmentymä toteuttaa yhden käyttötapauksen. Työn edetessä olen havainnut, että en tätä periaatetta ole pystynyt aivan noudattamaan.

Käyttöliittymässä olen ohjelmoinut pnlRivi - luokan, joka on periytetty JPanel - luokasta. Oma paneli kuvaa yhtä arvauskertaa ja sisältää alasvetovalikot arvauksen toteuttamiseen ja tekstikentän tuloksen ilmoittamiseen. Ohjelman käynnistyksen yhteydessä käyttöliittymään asetetaan kymmenen pnlRivi - luokan ilmentymää päällekkäin ja saadaan näin kokonainen pelilauta näkyviin. Alkuperäinen ajatukseni oli, että pnlRivi - paneleita olisi luotu jollain dynaamisella menetelmällä, joka olisi mahdollistanut eri suuruisten pelilautojen käsittelyn käyttöliittymässä. Nyt pelilauta on kiinteästi 4 x 10 kokoinen. Panelit talletetaan hajautustauluun käyttöliittymän jäsenmuuttujaksi. Tämä menettely mahdollistaa yksittäiseen paneeliin viittaamisen panelin nimellä ja toistaalta paneelien käsittelyn kokonaisuutena(Panelien alustus).

Pelin logiikka koostuu lähinnä kahdesta luokasta : Pelilauta ja Rivi. Rivi - luokka kuvaa yhden arvauskerran. Se sisältää kaksi taulukkoa, joista toiseen talletetaan ratkaisuyritys ja toiseen ratkaisuyrityksen tarkistuksen tulos. Rivi - luokan ilmentymät talletetaan Pelilauta - olioon. Pelilauta - olio luodaan kun peli aloitetaan. Käyttöliittymä pyytää pelilaudalta Rivi - oliota, josta käyttöliittymään luetaan tiedot.

Serialisoimalla logiikka - luokat (implements java.io.Serializable) on mahdollista tallettaa oliot levylle ja myöhemmin lukea ne takaisin siten, että olioiden tallatettu tila säilyy. Tätä ominaisuutta käytän pelitilanteen talletustoiminnossa.