

# Shakki

---

## OOP harjoitustyö

**Paavo Pärssinen 507551, Teppo Huhtala 505002**

**31.3.2014**

# Tehtävän kuvaus ja analysointi

Harjoitustyömme aiheena on shakkipeli, tarkemmin kuvailtuna tehtävänanto on seuraavanlainen:

” Harjoitustyössä tulee mallintaa shakkilauta ja pelin eri nappulat. Tekoälypelaajaa ei tarvitse toteuttaa, mutta työn tulisi mahdollistaa ihmispelaajien väliset ottelut. Graafisesti pelipöydän ja pelin tilan voi esittää komentorivillä ascii-grafiikalla tai graafisella käyttöliittymällä. Toteutuksessa pitää myös huomioida eri pelinappuloiden sallitut siirrot ja hoitaa automaattisesti nappuloiden syönnit.

Pelin logiikkaa ei muuten tarvitse toteuttaa, eli esimerkiksi pelin voittajaa ei tarvitse selvittää. Lisäksi pelin tila tulee olla mahdollista tallentaa, jotta peliä voi tarvittaessa jatkaa myöhemmin.”

Pelissä tulee siis toteuttaa luoda eri oliot shakkilaudalle, sekä eri nappuloille. Nappuloilla tulee olla metodeja, joilla niitä siirretään, ja joilla syödään vastustajan nappuloita sääntöjen rajoissa. Lisäksi tulee toteuttaa tekstitiedostoon tallennettava pelitiedosto, eli pelin tallentaminen.

## Ratkaisuperiaate

Päätimme alkaa ratkaisemaan tehtävänantoa seuraavalla tavalla. Pelin toteutuksena toimii komentorivillä toimiva shakkipeli, peliä ohjataan antamalla aluksi haluamansa nappulan koordinaatit ja sen jälkeen koordinaatit sinne mihin haluaa nappulan siirtyvän. Päätimme luoda olion jokaiselle nappulalle erikseen, ja niille yhteisen abstraktin yläluokan. Lisäksi päätimme luoda laudalle oman luokan, joka sisältää 64 Square-oliota, eli myös joka ruutu on oma olionsa. Game-luokka taas pyörittää itse peliä.

Tavoitteena oli että jokainen nappula voi siirtyä vain sääntöjen sallimalla tavalla, mutta ongelmaksi jäi vielä kuninkaan siirtyminen uhattuun ruutuun. Valitettavasti emme saaneet kuningasta toimimaan kunnolla ennen deadlinea, joten shakkipelimme on yksinkertaistettu versio shakista. Kuningas pystyy liikkumaan uhattuun ruutuun, ja peli voitetaan syömällä vastustajan kuningas. Pelin muut nappulat liikkuvat normaalisti lukuunottamatta ohestasyöntiä, ylennystä ja linnoittautumista.

Loimme seuraavat luokat:

- **Game**
  - Game-luokka mallintaa itse peliä, hoitaa siis kierrokset ja vuorot, pelintallennuksen sekä pelin lataamisen.
  - Metodi `inBounds(String coord)` tarkistaa onko käyttäjän syöttämät koordinaatit laudan sisällä
  - Metodi `makeX(String coord)` muuttaa käyttäjän syöttämät koordinaatit (esim. A1) ohjelman käyttämään muotoon (esim 11)
  - Metodit `saveGame()` ja `loadGame()` tiedostoon kirjoittamiseen ja tiedostosta lukemiseen
- **Board**
  - Board sisältää kahdessa ArrayListissä valkoiset, sekä mustat Piece-alkiot.
  - Lisäksi sisältää 8x8 taulukon Square-alkioita
  - Metodit `moveBlackPiece()` ja `moveWhitePiece()` nappuloiden siirtämiseen sekä `killPiece()` syömiseen liikkumisen yhteydessä.
  - Metodi `toPrint()` tulostamiseen, käy läpi Square-taulukon ja tulostaa niistä tiedot.
  - Metodi `toSave()` tallennusdatan luomiseen, palauttaa tulosteen yhdellä rivillä.
  - Metodit `checkIfWhiteWins()` ja `checkIfWhiteWins()` voittajan määrittämiseen
- **Square**
  - Sisältää oman sijaintinsa pelilaudalla ja tulosteen sisällöstään (mikä nappula)  
§ Metodi `whatPiece()` määrittää tulosteen oliolle
  - Metodi `whatColor()` sisältämänsä nappulan värin palauttamiseen
- **Piece**
  - Abstrakti Piece-luokka mallintaa yhtä pelinappulaa, aliluokkia ovat King, Queen, Bishop, Knight, Rook ja Pawn
  - Nappula tietää aina oman sijaintinsa, sekä sillä on enumina värinsä ja tulosteena oma alkukirjaimensa (Knightilla "H"-niinkuin Horse)  
§ King-olio tuntee lisäksi ArrayListit valkoisista ja mustista nappuloista, liittyy uhkien tunnistamiseen
  - Tietää myös aina mihin voi siirtyä. Tieto ArrayList-muodossa.  
§ Metodi `possibleSquares()` määrittää mihin voi siirtyä. Käy systemaattisesti läpi lautta selvittääkseen tämän
  - Metodit `getX()` ja `getY()` koordinaattien palauttamiseen

## Testausjärjestely

Peli toimii käyttäjän syöttämillä koordinaateilla, joten tärkeimmäksi testatessa muodostui syötteiden huomioiminen. Ohjelma tarkastaa aluksi että syötteet ovat oikean pituisia, sen jälkeen onko koordinaattien kirjain ja numero oikeassa järjestyksessä. Tämän jälkeen ohjelma testaa onko siirtymäkoordinaatit laudalla, ja sen jälkeen voiko kyseinen nappula siirtyä siihen. Tämän järjestys todettiin parhaaksi testausten avulla.

# Käyttöohje

1. Käynnistä peli ajamalla Game
2. Valitse uusi peli valitsemalla 1, tai lataa vanha peli valitsemalla 2. Valitsemalla 3 voit lopettaa pelin.
3. Valkoinen aloittaa. Siirrä nappuloita kirjoittamalla aluksi nappulan koordinaatit (esim. A7) paina enter, ja sen jälkeen kirjoita mihin koordinaattiin haluat siirtää kyseisen nappulan. Voit lopettaa pelin milloin vain kirjoittamalla 'end'.
4. Toisen pelaajan vuoro toimii samalla periaatteella. Peli tallentaa aina mustan vuoron jälkeen.
5. Peli päättyy kun toisen kuningas on syöty.