

Házi feladat specifikáció

Sógi - Domokos Erik Zsolt – I5A3X5

Feladat ismertetése

A házi feladat célja a japán sakk, azaz a **sógi** számítógépes megvalósítása **Java** nyelven, grafikus felülettel (**Swing** keretrendszer segítségével). A program a klasszikus sógi szabályait követi, és lehetőséget biztosít három játékmódra :

- **0-játékos (AI vs AI)** – automatikus demonstrációs mód,
- **1-játékos (Player vs AI)** – ember a gép ellen,
- **2-játékos (Player vs Player)** – helyi játék két ember között.

A játék 9×9-es táblán zajlik, ahol mindkét játékos azonos figurakészlettel rendelkezik. A figurák előre meghatározott módon léphetnek, és a leütött figurák különlegessége, hogy bizonyos szabályok szerint visszahelyezhetők a táblára, így a játékmenet dinamikusabb, mint a hagyományos sakkban.

A program grafikus felületen jeleníti meg a táblát és a figurákat. A felhasználó egérrel választhatja ki a lépni kívánt figurát, majd a célmezőt. A játék automatikusan ellenőrzi, hogy a lépés szabályos-e, végrehajtja azt, és átvált a másik játékosra.

Az alkalmazásban külön menü található, amelyen keresztül új játék indítható, mentett állás betölthető, illetve a játék elmenthető fájlba. A játékállapot elmentése **JSON** formátumban történik, a tábla és a játékosok aktuális adataival együtt.

A házi feladatot teljesen önállóan elkülönített modulokban valósítom meg: a játékszabályokat kezelő logika, a grafikus megjelenítés és a fájlkezelés külön osztályokban lesznek kezelve.

A program képes lesz felismerni a játék végét (pl. király leütése esetén), és a felhasználó számára üzenetet jelenít meg a győztesről.

Use-case leírások

- **Új játék indítása** : A felhasználó a menüből kiválasztja az „**Új játék**” opciót, majd megadja a játékmódot (0, 1 vagy 2 játékos). A tábla alaphelyzetbe kerül, és elindul a játék.
- **Lépés végrehajtása** : A felhasználó rákattint egy saját figurájára, majd egy célmezőre. A program ellenőrzi, hogy a lépés szabályos-e, és ha igen, végrehajtja. Ezután a kör a másik játékoshoz kerül.
- **Figura leütése és visszahelyezése** : Ha egy figura leütésre kerül, bekerül az adott játékos „kézben tartott” figurái közé. Később a játékos dönthet úgy, hogy visszahelyezi valamelyiket a táblára, a sógi szabályai szerint.
- **AI lépése** : Egyjátékos módban a gép automatikusan végrehajt egy véletlenszerű, de szabályos lépést. A későbbi verzióban fejleszthető egyszerű értékelő algoritmussal.
- **Játék mentése** : A játékos a menüből kiválaszthatja a „**Mentés**” opciót, amely **JSON** formátumban elmenti a tábla, a figurák és a játékosok aktuális állapotát.
- **Játék betöltése** : Korábban elmentett játék betölthető, amely visszaállítja a tábla és a játékosok állapotát.

- **Játék vége felismerése** : A program automatikusan figyeli, ha valamelyik király (Ōshō vagy Gyokushō) leütésre kerül, és befejezi a játékot, megjelenítve a győztes nevét.
- **Kilépés** : A felhasználó bezárhatja az alkalmazást vagy kiválaszthatja a „**Kilépés**” menüpontot. A program ilyenkor mentési lehetőséget ajánl fel.

Megoldási ötlet, technológiák

A program **Java** nyelven készül, **Swing** grafikus felhasználói felülettel.

A tábla megjelenítése a **Graphics** osztály segítségével történik, minden mező és figura kirajzolása külön metódusban valósul meg.

A figurák mozgási szabályait absztrakt **Piece** őssztály határozza meg, a konkrét figurák (**King**, **Rook**, **Bishop**, stb.) ebből származnak le.

A tábla (**Board**) és a játékvezérlő (**ShogiGame**) osztály felel a lépések, leütések és körváltások kezeléséért.

A fájlkezelést a **SaveManager** osztály végzi, amely **JSON** formátumban menti és tölti vissza a játékot (pl. **Gson** könyvtár használatával).

A játékosok adatait a **Player** és **AIPlayer** osztályok reprezentálják.

A tesztelés **JUnit 5**-tel történik, legalább három osztály (pl. **Board**, **Piece**, **SaveManager**) metódusaira kiterjedően.

A program fő komponensei :

- **ShogiGame** – játéklogika, körváltás, állapotkezelés
- **Board** – tábla reprezentációja és lépésellenőrzés
- **Piece / leszármazottak** – figurák viselkedése és mozgása
- **ShogiGUI** – grafikus felület és eseménykezelés
- **SaveManager** – mentés és betöltés **JSON** formátumban
- **Player / AIPlayer** – játékoskezelés
- **Test classes (JUnit)** – logikai és szabálytesztek, pl. lépésellenőrzés, mentés visszatöltés

Grafikus megjelenítés

- A 9×9-es tábla kirajzolása **paintComponent(Graphics g)** segítségével.
- A figurák rajzolása japán kanjikkal vagy ikonokkal.
- A játékosok keze (fogva tartott figurák) külön panelen látható.
- A menüsor (**JMenuBar**) tartalmazza: **Fájl** → **Új játék**, **Mentés**, **Betöltés**, **Kilépés**.

Tesztelés

- **Piece.canMoveTo()** (különböző figurák esetén)
- **Board.isOccupied()**, **Board.movePiece()**
- **SaveManager.save()**, **SaveManager.load()**
- **ShogiGame.isCheckmate()**, stb.