

Sakk HF-Specifikáció

Feladat

A Sakk játékot fogom megvalósítani, normális szabályokkal. Lehesen játszani két emberi játékoskal, egy ember egy gép ellen vagy teszteléshez akár két gép egymás ellen is. A játék pillanatnyi állását el lehessen menteni fájlba is.

Sakk szabályai és a játék menete

A Sakkot egy 8x8-as négyzet alakú pályán játsszák. A képen látható alapállással. 6 fajta különböző bábu van, ami mind máshogyan mozog. A **király** bábugól csak egy van mindkét félnek és ha a játékos nem tud úgy lépni hogy a királya ne legyen leüthető a következő körben, akkor veszít. Bármely irányban (vízszintesen, függőlegesen, átlósan) léphet, de csak egy mezőt és csak oda ahol nincs saját színű bábuja (mert egyszerre csak egy bábu lehet egy mezőn). A **vezér** mozgása olyan, hogy egyenesen vagy átlósan bármely irányban, bármennyi mezőt léphet, mindaddig, amíg a tábla széléhez nem ér, vagy



egy másik figura nem kerül az útjába (ha ez ellenséges figura, kiütheti, ha saját bábu, meg kell állnia egy mezővel előtte). Ez egyébként a huszár kivételével minden figurára igaz. A **bástya** bármennyi mezőt léphet, de csak függőleges és vízszintes irányban (amíg egy másik bábu nem kerül útjába), átlósan nem. A **futók** átlós irányban léphetnek, bármennyi mezőt, amíg egy másik bábu nem kerül útjába. A **huszár** tud lépni: vízszintesen jobbra vagy balra két mezőt, majd függőlegesen fel vagy le egyet (vagy fordítva: függőlegesen fel vagy le kettőt és vízszintesen jobbra vagy balra egyet). A **gyalog** kizárólag előre léphet. A kiindulási helyéről mind a nyolc gyalog tetszés szerint egy vagy két mezőt léphet előre, de a továbbiakban lépésenként mindig csak egy mezőt haladhat előre. Ütni azonban csak jobbra vagy balra átlósan tud, szintén csak egy mezőnyi távolságra. A játékot mindig világos kezdi, és tetszése szerint – de a szabályoknak adta kereteken belül – valamelyik figuráját áthelyezi egy másik mezőre. Ezt lépésnek hívjuk. Ezt követően sötét lép egyet. A játékosok felváltva lépnek, passzolásra nincs lehetőség („lépéskényszer”). Ha egy mezőn az ellenfél bábuja áll, azt egy szabályos lépéssel ki lehet ütni: az ellenfél figuráját levesszük a tábláról, és saját, odalépő bábunkat tesszük a helyére. Saját figura kiütésére nincs lehetőség. Ha egy játékos nem tud lépni, de a másik játékos nem tudná leütni a királyát egyből, akkor döntetlen az eredmény.

Különleges szabályok

Ha egy gyalog eléri a tábla másik végét, akkor átalakulhat egy bármilyen másik bábuvá (kivéve király), általában vezért szoktak választani az emberek, de néha a huszárnak is van értelme.

Van még a sáncolás, ahol a király egyszerre két mezőt lép oldalra, és a bástya a király másik oldalára kerül. Ehhez ennek kell a király és a bástya első lépésének is lennie és köztük nem állhat más bábu.

Funkciók

A pályát és a bábukat vizuálisan, alacsonyszintű grafikai rutinokkal (a Graphics osztály felhasználásával) fogom megjeleníteni. A program kiírja épp melyik játékos jön, és az rá tud kattintani a saját bábuira. Ha az a bábu tud lépni, akkor megjeleíti a lehetséges lépéseit a program, ha nem, akkor a felhasználó újra tud választani. A játék állását egy erre kijelölt gomb megnyomásával le lehet menteni, és azt későbbi indításkor újratölteni egy másik gombbal. A játék előtt ki lehet választani a módot(0, 1 vagy 2 emberi játékos). A játék végét észreveszi a program és kiírja a nyertest. Opcionálisan időzítőt is lehet indítani, ha időre szertenének játszani a felhasználók.

Use Case-ek

A játékos kiválasztja a játékmódot.(2 emberi játékos felváltva lép, egy ember gép ellen vagy két gép játszik egymás ellen)

A játékos rákattint egy bábura, ami megjeleníti a lépési lehetőségeit a táblán.

A játékos rákattint egy megjelenített lépési lehetőség mezőre(színes kör van a négyzetben) és ennek hatására oda lép a előzőleg kijelölt bábu.

A játékos rákattint a mentés gombra, ami kimentti a játék jelenlegi állását.

A játékos rákattint a visszatöltés gombra, ami visszatölti az előzőleg elmentett játék állását.

Ha az egyik játékos veszít(vagy döntetlen), akkor azt jelzi a program a játékosnak.

Megvalósítás

A bábukat egy abstract osztályból fogom származtatni. Mindegyik bábu máshogy fog reagálni, hogyha rákattintanak. A szabályok alapján ellenőrzik, hogy hova léphetnének és azt tovább küldik a fő játék loopnak, ami megjeleníti a felhasználónak. Lépéseket embertől vagy géptől is kaphat a program. A gépi játékos látja a táblát és az alapján generál legális lépéseket. A bábukat egy tábla osztályban fogom tárolni. A játék állását a Serializeable implementálásával fogom fájlba menthetővé tenni. A tábla megjelenítéséhez a Graphics osztályt használom, a bábukat képfájlból olvasom be és azt jelenítem meg. A sáncolást és a parasztok átalakulását is a bábukban oldom meg.

Fontosabb függvények leírása

Játék

A run függvény kér lépéseket a RobotJatekosoktól, ha éppen olyan játékmód van beállítva és megy még a játék(nincs vége).

Az ElfogyottAzIdo függvényt az órák hívják meg a Jatekon, ha lejártak, aminek hatására a Jatek leállítja a játékot és kiírja hogy melyik játékos nyert.

A VegeVanMar függvény ellenőrzi a Tabla segítségével, hogy vége van-e a játéknak, ha igen, akkor kiírja hogy ki nyert(vagy azt hogy döntetlen).

Tabla

A Tabla a konstruktorában a megadott fájlnevből olvassa ki a bábuk megjelenítésére használt képeket.

A BabukAlaphelyzetbe függvény beállítja a Tabla két listájában a bábukat úgy, hogy a sakk szabályai szerint a kezdőállapotban legyenek.

A Get függvények visszaadják a kért változót.

A LepesLegalizalas függvény a megadott bábu összes lépését ellenőrzi és csak azokat rakja vissza a lépések közé, amelyek után nem lehet leütni a saját királyát.

A VesztettE függvény megnézi, hogy tud-e úgy lépni a játékos úgy, hogy ne lehessen utána leütni a királyát, ha nem akkor vesztett, ha nem tud lépni, de nem lehet leütni a következő körben a királyát akkor döntetlen.

A readObject függvény a Serializable függvényét definiálja felül, azért, mert a képeket nem szerializálom csak a fájlnevet és onnan olvasom ki beolvasásnál. (mert az Image nem szerializálható)

Babu

A konstruktorban a megadott x és y értékekből csinálja a pozícióját, beállítja a színét és megkapja melyik táblán van.

A TablanVanE statikus függvény megnézi egy pontról, hogy egy sakktáblán van-e. Ezt főleg a lehetséges lépések számolásánál használok, hogy ne kelljen mindig kiírni a feltételekben.

A Get függvények visszaadják a kért változót.

A HovaLephet függvény abstract, mert a leszármazottak különböző módon valósítják meg. Vissztérési értéke igaz, ha van legalább egy lépése, hamis, ha nem tud sehova se lépni. Beállítja a lépési és ütési lehetőségek listáját.

A Lepes függvény egy x, y paraméterként megadott pozícióra próbálja léptetni a bábút, amin meghívták ezt a függvényt. Ha sikerült, akkor igazzal tér vissza, ha nem, akkor hamissal. Először megnézi, hogy a lépési vagy ütési lehetőségeiben benne van-e ez a pont, aztán azt hogy azon a mezőn már van-e másik bábu, ha saját színű, akkor nem léphet oda, ha ellenfél, akkor leüti.

Ora

A konstruktor beállítja, hogy melyik szín óraja, mennyi időről indul, melyik játékhoz tartozik és hogy melyik JTextField-be írja ki az időt.

A Szunet függvény leállítja az órát ideiglenesen.

Az Indit függvény elindítja az óra számolását.

Az IdoMentes elmenti a pillanatnyi időt.

Az IdoVisszaallitas visszatölti az elmentett időt.

A run függvény másodpercenként csökkenti az időt, ha megy az óra és kiírja a JTextField-be a jelenlegi időt. Ha 0 lesz az idő akkor meghívja a játékon az ElfogyottAzIdo függvényt a saját színével.

RobotJatekos

A konstruktor beállítja robot színét és hogy melyik táblán játszik.

A SetTabla függvény átállítja a robot tábláját.

A LepesGeneralas függvény végigmegy a saját bábuin és megnézi hogy hova tudnak lépni, ha tud ütni a bábu akkor az ütései, ha csak lépni tud, akkor a lépései közül választ egy lépést, amit aztán meg is valósít a táblán.

Kiraly

A konstruktor meghívja a Babu ős konstruktorát.

A HovaLephet függvény megnézi az összes irányba az 1 távolságra lévő mezőn hogy van-e bábu, ha nincs akkor jó lépésnek, ha ellenfél akkor jó ütésnek. Aztán ha még nem lépett és a sáncolás összes feltétele teljesül akkor azt a mezőt is hozzáadja a lépésekhez.

A Lepes függvény nagyon hasonlít a Babu ős Lepes függvényére, de a sáncolás esetén még a bástyát is mozgatja.

Gyalog

A konstruktor meghívja a Babu ős konstruktorát.

A HovaLephet megnézi előre az első mezőt(első kettőt, ha még nem lépett) és ha üres, akkor jó lépésnek. Átlósan előre is megnézi az első mezőket és ha ellenfél van rajta, akkor jó ütésnek.

A Lepes függvény nagyon hasonlít a Babu ős Lepes függvényére, de ha az ellenkező oldalra ér a Gyalog, akkor a helyére egy királynőt rak(a RobotJatekos kedvéért csak simán királynő, választás helyett).

Futo

A konstruktor meghívja a Babu ős konstruktorát.

A HovaLephet függvény megnézi az átlós irányokat a pálya széléig, amíg nem talál másik bábut addig jó lépésnek, ha talál, akkor ha azonos színű, akkor simán leáll, ha ellenfél, akkor jó ütésnek és leáll.

Basty

A konstruktor meghívja a Babu ős konstruktorát.

A HovaLephet függvény megnézi a vízszintes és függőleges irányokat a pálya széléig, amíg nem talál másik bábut addig jó lépésnek, ha talál, akkor ha azonos színű, akkor simán leáll, ha ellenfél, akkor jó ütésnek és leáll.

Huszar

A konstruktor meghívja a Babu ős konstruktorát.

A HovaLephet függvény megnézi az olyan pozíciókat, amik az egyik koordináta szerint kettővel a másik szerinte eggyel különböznek a jelenlegi helyétől és ha üres akkor jó lépésnek, ha ellenfél van rajta akkor jó ütésnek.

Vezer

A konstruktor meghívja a Babu ős konstruktorát.

A HovaLephet függvény megnézi a vízszintes, a függőleges és az átlós irányokat a pálya széléig, amíg nem talál másik bábut addig jó lépésnek, ha talál, akkor ha azonos színű, akkor simán leáll, ha ellenfél, akkor jó ütésnek és leáll.

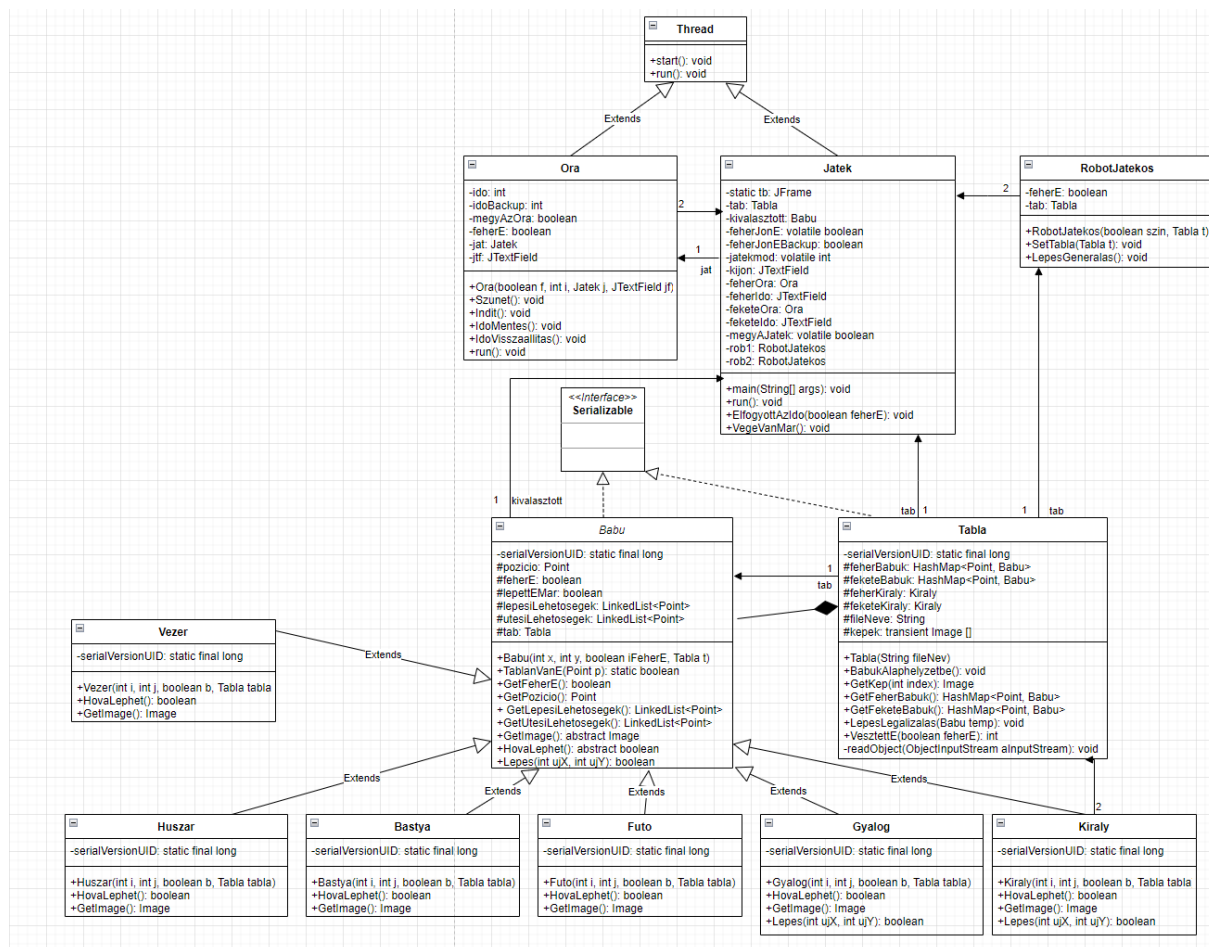
Megoldási vázlat

A feladatot úgy kezdtem hogy először az üres pályát rajzoltam ki, aztán megcsináltam a bábuakat hogy különböző módon nézzenek ki és lépjenek. Majd hozzáadtam a RobotJatekos osztályt hogy tudjon maga ellen is játszani, aztán megcsináltam a sakkórát és a menürendszert a szerializálással. Közben használtam a JUnit könyvtárat a teszteléshez.

Tesztelési dokumentáció

A tesztesetek a Babu leszármazottokon azt nézi hogy szabályosan mozognak-e. Az órát teszteli hogy megáll-e ha megállítják, és hogy rendesen számol-e. A RobotJatekoson megnézi hogy lép-e. A Tabla tesztjében nézem meg hogy a LepesLegalizalas tényleg működik-e, azaz nem enged, hogy a királyt leüssék.

Osztálydiagram



Felhasználói kézikönyv

A felugró ablakon a felhasználó kiválaszthatja a játékmódot. (2 emberi játékos felváltva lép, egy ember gép ellen vagy két gép játszik egymás ellen) Aztán rákattinthat egy bábura, ami megjeleníti a lépési lehetőségeit a táblán. Ha rákattint egy megjelenített lépési lehetőség mezőre (színes kör van a négyzetben), akkor ennek hatására oda lép a előzőleg kijelölt bábú. Ezután a másik szín jön (amit lehet hogy a számítógép irányít). Ha a játékos rákattint az opciók gombra a menüsávban, akkor a kinyíló menüben a gombok megnyomásával tud menteni, vissztölteni vagy átállítani a játékmódot ami kimentti a játék jelenlegi állását. Az ablak alján jelzi a program, hogy éppen ki jön, és hogy melyik játékosnak mennyi ideje van. Ha az egyik játékos veszít (vagy döntetlen), akkor azt jelzi a program a játékosnak.

Források

Forrás a szabályokhoz: <https://hu.wikipedia.org/wiki/Sakk> ; <https://www.chess.com/>
 Bábuk kinézete: https://commons.wikimedia.org/wiki/Template:SVG_chess_pieces