

- *public abstract void leutve()*

Ha leütik, akkor az ellenfél csapatának a leütött bábújaihoz kerül.

- *public Babu eloleptetes()*

Ha a bábu eléri az ellenfél első 3 sorát, akkor előléptetődik egy erősebb bábuvá. Ez a függvény azért nem abstract, mert vannak bábuk pl: akik már elő lettek léptetve, így már nem tudnak mit tenni. A leszármaztatott osztályok, amelyeknél van lehetőség az előléptetésre, azok override-olják.

- *public void paint(Graphics2D g)*

Kirajzolja a bábút a sor és oszlop értéke által meghatározott mezőre a táblán.

- *public void sor_modositas(int uj)*

Átváltoztatja a sor értékét a paraméterként kapott értékre.

- *public void oszlop_modositas(int uj)*

Átváltoztatja az oszlop értékét a paraméterként kapott értékre.

- *public int sor_lekerdezes()*

Lekérdezi a sor értékét.

- *public int oszlop_lekerdezes()*

Lekérdezi az oszlop értékét.

- *public boolean feher_e()*

Lekérdezi, hogy fehér-e az adott bábu.

- *public Kep_beolvaso kep_beolvaso_lekerdezes()*

Lekérdezi a Kep_beolvaso-t.

- *public Tabla tabla_lekerdezes()*

Lekérdezi, hogy melyik táblához tartozik. Fő funkciója, hogy a különböző függvényekben el lehessen érni a Tábla-t.

A package-ben található többi osztály az itt felsorolt *tipus_lekerdezes()*, *lepes_szabaly(int uj_sor, int uj_oszlop)* *leutve()* és esetenként az *eloleptetes()* függvényeket valósítja meg.

Babukhoz_valo package

Ellenorzott_lepes osztály

- *public static boolean ervenyes_lepes(Babu babu,int uj_sor,int uj_oszlop)*

Ezzel a függvénnyel ellenőrzi, hogy ha az adott sorra és oszlopra lép, az megfelel-e a lépésszabálynak és, hogy azzal nem-e teszi veszélybe a királyát. A veszély ellenőrzését a Lepes_ellenorzo osztállyal végzi el egy másik (az eredetiről leklónozott) táblán végzi el.

Leutott_babuk osztály

Struktúra, amivel eltárolja a program, hogy a leütött bábukból mennyi van színenként, típusonként külön szedve.

- *public void hozzasdas(boolean feher,String tipus)*

A paraméterként megadott szín és típus által meghatározott bábuhoz hozzáad 1-t.

- *public void kivonas(boolean feher, String tipus)*

A paraméterként megadott szín és típus által meghatározott bábuhból kivon 1-t.

- *public boolean van_belole(String tipus,boolean feher)*

Megnézi, hogy a paraméterként megadott szín és típus által meghatározott bábuból van-e

Fajl_kezeles package

Beolvaso osztály

A klónozott táblák létrehozásának és a lerakások megvalósításához szükséges osztály.

- *public static Babu atalakitas(String tipus, int sor, int oszlop,boolean feher_e,Kep_beolvaso kep_hash, Tabla tabla)*

Létrehoz a paraméterként megadott értékekkel egy új bábút

Kep_beolvaso osztály

Struktúra, hogy csak egy típust kelljen megadni, és meg is kapja a bábu az ő saját képét.

Mentes osztály

Mentések létrehozásáért, és betöltéséért felelős osztály.

- *public static void mentes(Tabla jatekter)*

Lementi a játék jelenlegi helyzetét a „jatekmenet.txt” fájlba. Ha a mező nem tartalmaz bábut, akkor ’,’ ír ki, ha pedig nem, akkor először leírja a típusát, majd a sorát oszlopát, végül pedig, hogy melyik színű csapatba tartozik. A különböző mezőket egy ’;’-el választja el.

- *public static void betoltes(Tabla jatekter)*

Az előző függvény által létrehozott „jatekmenet.txt” fájlt olvassa be, ha létezik a fájl.

Mentes_gomb osztály

- *public void actionPerformed(ActionEvent e)*

Gomb nyomására lementi a tábla jelenlegi helyzetét.

Betoltes_gomb osztaly

Gomb nyomásra betölti a mentést, ha az létezik.

Jatek package

Tabla osztály

A Tabla a játék fő konténereként funkcionál. Itt tárolja el a játéktáblát, a leütött bábukát, hogy sakkban van-e az egyik szín, vagy hogy éppen mi van kiválasztva vagy, hogy vége van-e a játéknak.

- *public void lepes_modositas(int i)*

Megváltoztatja a jelenlegi kör értékét a paraméterként kapott számra. Az egyetlen hely, ahol ki van használva az a mentés betöltése, mert ott nem csak 1-el kell növelni a jelenlegi kört.

- *public int lepes_lekerdezes()*

Lekérdezi a jelenlegi kör értékét.

- *public void lepes_befejezese()*

Növeli egyel a jelenlegi kört.

- *public boolean foglalt_mezo_e(int sor, int oszlop)*

Ellenőrzi, hogy van-e bábu a paraméterként megadott mezőn.

- *public Babu mezo_lekerdezes(int sor, int oszlop)*

Lekérdezi a paraméterek által meghatározott mező tartalmát.

- *public void mezo_modositas(Babu uj, int sor, int oszlop)*

Megváltoztatja a paraméterként megadott mező tartalmát egy új bábura.

- *public void kivasztott_modositasa(Babu uj)*

Megváltoztatja a kiválasztottat a paraméterként kapott bábura.

- *public Babu kivasztott_lekerdezese()*

Lekérdezi a kiválasztott bábút.

- *public void leutott_kivasztott_modositasa(String tipus)*

Megváltoztatja a leütött kiválasztott értékét.

- *public String leutott_kivasztott_lekerdezese()*

Lekérdezi a leütött kiválasztott típusát.

- *public void leutott_kivasztott_feher_e_modositasa(boolean feher)*

Módosítja, hogy a leütött kiválasztott melyik csapathoz tartozik.

- *public boolean leutott_kivasztott_feher_e_lekerdezese()*

Lekérdezi, hogy a leütött kiválasztott melyik csapathoz tartozik.

- *public Babu kiraly_kereso(boolean feher)*

Megkeresi a paraméterként megadott csapat királyát.

- *public Kep_beolvaso kep_beolvaso_lekerdezese()*

Lekérdezi a Kep_beolvaso-t.

- *public Leutott_babuk leutott_lekerdezese()*

Lekérdezi a leütött bábuk strukturáját.

- *public boolean vege_lekerdezese()*

Lekérdezi, hogy vége van-e a játéknak.

- *public void vege_modositas(boolean vege)*

Megváltoztatja a vege változó értékét.

- *public synchronized void paint(Graphics g)*

Kirajzolja a Tabla-hoz tartozó elemeket.

Rajzoló osztály

A táblán lévő dolgok kirajzolásáért felel. Külön lett szedve a Tabla-tól átláthatóság kedvéért.

- *public static void mezo_rajzolasa(Graphics2D g2d, Tabla jatekter, int sor, int oszlop)*

Kirajzolja a táblát, az azon lévő bábukat. A kiválasztott bábut egy zöld négyzettel jelöli, a veszélyben lévő királyt, pedig pirossal.

- *public static void kivlasztott_rajzolasa(Graphics2D g2d, Tabla jatekter, int sor, int oszlop)*

Zöld kört rajzol azokra a mezőkre, amelyek érvényes lépésnek számítana a kiválasztott bábunak. Piros kört meg azokra, ahova tudna lépni, azonban nem tud, mert azzal sakkba helyezné a királyt.

Leutot_rajzó osztály

- *public static void fekete_leutottek_rajzolasa(Graphics2D g2d, Tabla jatekter)*

Kirajzolja az ablak bal oldalára a feketéknek a leutott bábuikat (amik eredetileg fehérek voltak).

- *public static void feher_leutottek_rajzolasa(Graphics2D g2d, Tabla jatekter)*

Kirajzolja az ablak jobb oldalára a fehéreknek a leutott bábuikat (amik eredetileg feketék voltak).

- *public static void leutott_kivlasztott_rajzolo(Graphics2D g2d, Tabla jatekter, int sor, int oszlop)*

Ha kiválasztott egy adott bábut a felhasználó, akkor megmutatja, hogy hova lehet őket lerakni.

Bemenet osztály

A konstruktor paramétere alapján eldönti, hogy mely körökben tud a felhasználó rákattintani a táblára, a bábukra.

- *private void tablan_kivalasztas(MouseEvent e)*

Az egér gomb nyomását reagálja le, megnézi, hogy az hol történt, az alapján pedig megszerzi, hogy hanyadik sorra és oszlopra kattintott a felhasználó. Ha már ki van választva egy adott bábu, akkor megpróbál a következő kattintás helyére lépni.

- *private void leutott_kivalasztas(MouseEvent e)*

Megnézi, hogy melyik oldalt kattintott a felhasználó, az alapján pedig belövi, hogy bábura kattintott. Ha már ki van választva egy bábu, akkor azt megpróbálja a következő kattintás helyére letenni.

Jatek_vege osztály

- *public void vege(Tabla jatek, My_Frame program)*

Ha vége a játéknak, akkor leveszi a képernyőről a játékot, és újra elérhetővé teszi a menü gombjait.

Jatek_vege_ellenorzo osztály

Egy szál, ami ellenőrzi, hogy megy-e még a játék. Ehhez az előbb említett osztályt veszi segítségül. Ha vége a játéknak, akkor leáll.

Robot_lepes osztály

- *private synchronized List<Mozgas> egy_mezohoz_tartozo_lepesek(int sor, int oszlop, Tabla jatekter, boolean feher)*

Kigyűjti a paraméterek által meghatározott mezőből indítható érvényes lépéseket.

- *private synchronized List<Mozgas> leutottek_felrakasa(String tipus, Tabla jatekter, boolean feher)*

Összegyűjti az összes érvényes lerakásos lépést.

- *public synchronized void lepes(Tabla jatekter, boolean feher)*

Az előző két függ kigyűjti a paraméter által meghatározott színű csapatnak az összes érvényes lépését, majd azt el is végzi.

Robot osztály

Egy szál, ami megvárja a saját körét, amit a paraméterben dönt el, az boolean értéke alapján, majd pedig az előbb ismertetett osztály segítségével lép egyet.

Kepernyo package

My_Frame osztály

Létrehoz egy új ablakot, amelyen elhelyez 3 gombot. Ezekkel a gombokkal lehet navigálni a különböző játékmódok között.

Nulla_jatekos_gomb osztály

- *public void actionPerformed(ActionEvent e)*

Létrehoz egy új játékmenetet 2 bottal. A menüben található gombokat láthatatlanná teszi, és kikapcsolja azokat.

Egy_jatekos_gomb osztály

- *public void actionPerformed(ActionEvent e)*

Létrehoz egy új játékmenetet egy bottal és egy játékosal. A színeket egy random szám alapján dönti el. A menüben található gombokat láthatatlanná teszi, és kikapcsolja azokat.

Ket_jatekos_gomb osztály

- *public void actionPerformed(ActionEvent e)*

Létrehoz egy új játékmenetet két játékosal. A menüben található gombokat láthatatlanná teszi, és kikapcsolja azokat.

Lepesekhez package

Mozgas osztály

Két mozgásfajtát gyűjt össze a lépést és a lerakást.

- *public abstract void lepes()*

Elvégzi a lépést.

- *public abstract void proba_lepes()*

Elvégzi a lépést. Ez a két lépés fajtánál más-más dolgot végez el.

- *public Babu mozgatni_valo_lekerdezese()*

Lekérdezi, hogy melyik bábút akarja mozgatni.

- *public String tipus_lekerdezes()*

Lekérdezi, hogy milyen típusú bábút akar letenni.

- *public Tabla jatekter_lekerdezes()*

Lekérdezi, hogy melyik táblán akarja elvégezni.

- *public int sor_lekerdezes()*

Lekérdezi, hogy melyik sorban akarja elvégezni a műveletet.

- *public int oszlop_lekerdezes()*

Lekérdezi, hogy melyik oszlopban akarja elvégezni a műveletet.

Lepes_ellenorzo osztály

- *public static boolean megmenti(Tabla jatekter,int sor, int oszlop)*

Megnézi, hogy a kiválasztott bábu adott lépése megmenti-e a sakktól a saját királyát.

- *public static boolean megmento_lerakas(Tabla jatekter,int sor, int oszlop)*

Megnézi, hogy a kiválasztott bábu lerakása megmenti-e a sakktól a saját királyát.

Lepes osztály

A táblán lévő bábuk kezelésére szolgáló osztály.

- *private Babu eloleptetes(Babu mozgatni_valo ,int uj_sor, int uj_oszlop)*

Megnézi, hogy a mozgatni való bábu az ellenfelének az első 3 sorába lépett-e be. Ha igen akkor meghívja a bábu eloleptetes függvényét, és azzá változik át.

- *private void babuk_cserelese(Babu mozgatni_valo ,int uj_sor, int uj_oszlop)*

Kicseréli az cél mező tartalmát a mozgatni való bábuval, és befejezi a lépést.

- *public void lepes()*

Megnézi, hogy az egy érvényes lépés lenne-e, csak akkor végzi el a lépést, ha nem foglalt a mező, vagy ha foglalt, akkor nem egy csapattársa.

- *public void proba_lepes()*

Ugyan azt csinálja, mint a *lepes()* függvény, csak itt a *lepes_szabaly()* függvényt használja, hogy a *lepes_ellenorzo()* ellenőrizni tudja, hogy eltűnt a sakk.

Lerakas_szabalyok osztály

A sógiban vannak különböző szabályok a lerakással kapcsolatban, ezeket ellenőrzi ez az osztály.

- *public static boolean lepes_nelkuliek_szabaly(String tipus,int uj_sor,int uj_oszlop,Tabla jatekter)*

Ellenőrzi, hogy a típus az gyalog vagy jari, vagy lovas, mert ezeket nem lehet lerakni az utolsó sorban, vagy a lovas esetében még az utolsó előttibe se, mert onnan nem tudnának elmozdulni.

- *public static boolean ket_gyalog_szabaly(String tipus,int uj_oszlop,Tabla jatekter)*

Ellenőrzi, hogy van-e más gyalog bábu a kiválasztott oszlopban, ugyan olyan színben.

- *public static boolean szabalyos(String tipus,int uj_sor,int uj_oszlop,Tabla jatekter)*

Megnézi, hogy a lerakás betartja-e ezeket a szabályokat, és hogy van-e olyan típusú bábu a paraméter által meghatározott leütöttek Hash-ben.

Lerakas osztály

- *public void lepes()*

Megnézi, hogy szabályos lenne-e a lerakás.

- *public void proba_lepes()*

Megnézi, hogy szabályos lenne-e a lerakás, ez a *lepes_ellenorzo()* miatt szükséges, mert ott el is végzi a lerakást, és az alapján megnézi, hogy változott-e a sakk állapota.

Sakk_matt_ellenorzo osztály

- *private static boolean Sakk_matt_check_belseje(Tabla jatekter,boolean feher, int sor, int oszlop)*

Megnézi, hogy ennek a bábunak van-e olyan lépése, ami megmentené a csapatát.

- *private static boolean Sakk_matt_check_leutottes_belseje(Tabla jatekter,String tipus, boolean feher)*

Megnézi, hogy van-e olyan lerakás, amely megmentené a csapatát.

- *public static boolean Sakk_matt_check(Tabla jatekter,boolean feher)*

Addig fut, amíg nem talál egy olyan lépést/lerakást, ami meg tudná menteni a csapatát. Ha nem talált egy ilyen lépést, akkor visszajelez, hogy sakk-matt van.

- *private static boolean Kiraly_checkelve(Tabla jatekter,boolean feher)*

Megnézi, hogy van-e olyan bábuja a paraméter által meghatározott csapatban, ami veszélyezteti az ellenséges királyt.

- *public static void Kissebb_eroforrasu_check_ellenorzes(Tabla jatekter,Babu mozgatnivalo)*

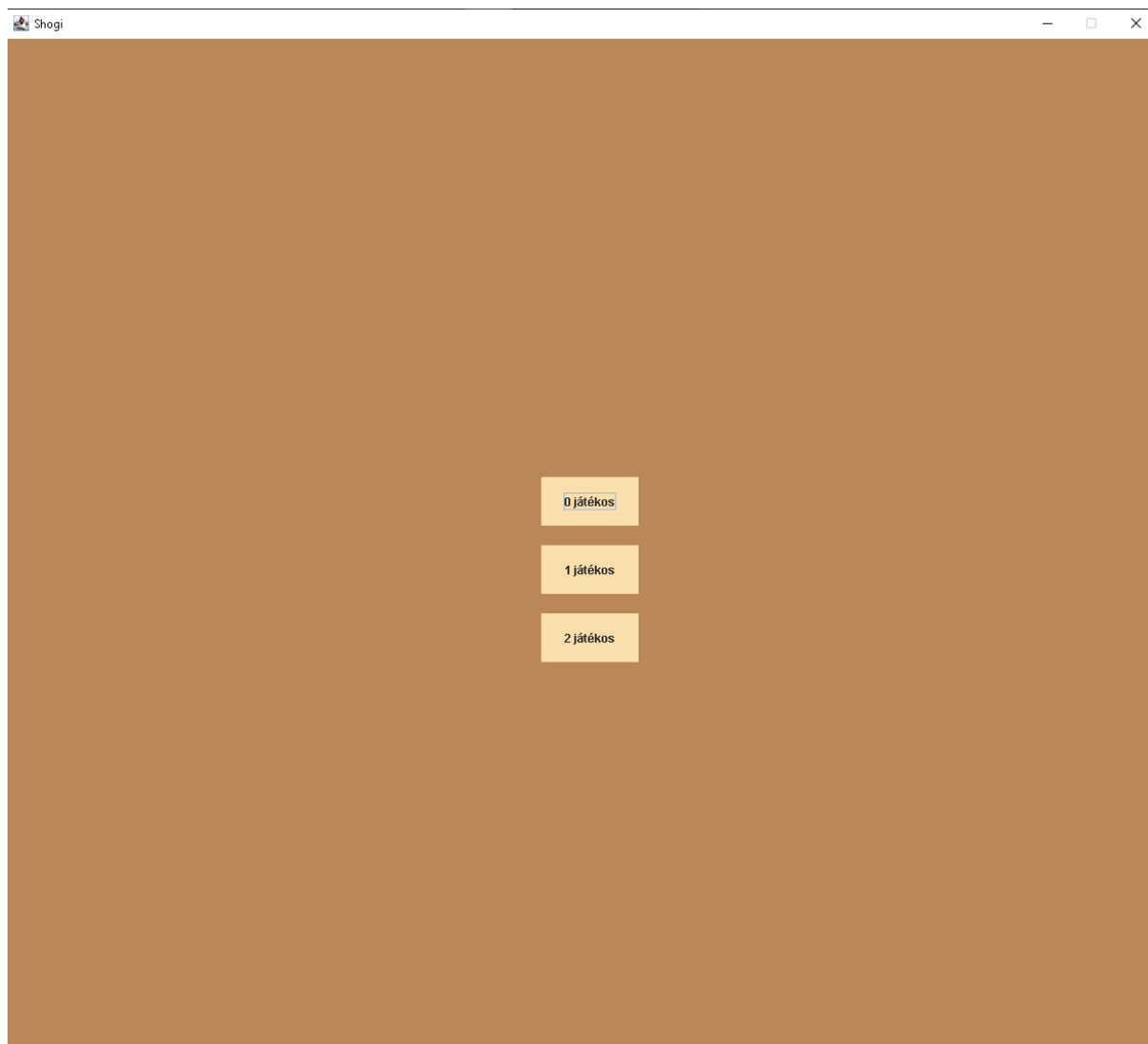
Megnézi, hogy a mozgatni való bábu a lépésével sakk-ba keríti a csapatát, vagy kikeríti magát onnan.

- *public static void Check_ellenorzes(Tabla jatekter)*

Megnézi mind a két színre, hogy sakokban van-e és attól függetlenül mind a két színre futtat egy sakk-matt checket is.

Felhasználói dokumentáció

A felhasználót program elindításakor egy menü várja.



A felhasználó 3 gomb közül tud választani:

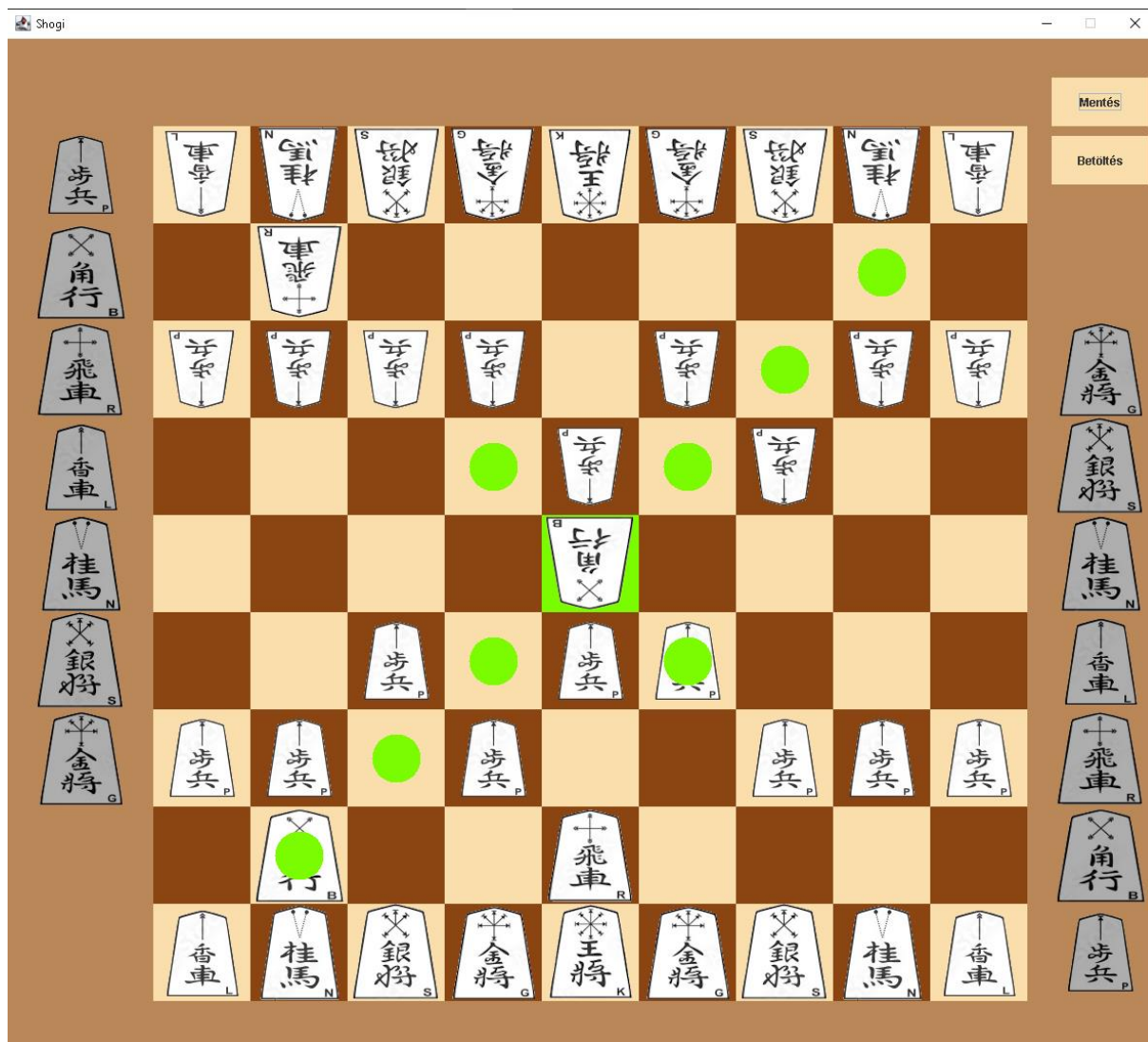
1. A 0 játékos
2. Az 1 játékos
3. A 2 játékos

A 0 játékos esetében a felhasználó 2 botot néz, ahogy játszanak. Ha valamelyikük veszít akkor visszatér a menübe.

Az 1 játékos esetben, a felhasználó egy bot ellen játszik. A kezdési sorrendet egy a játék eldönti magától véletlenszerűen.

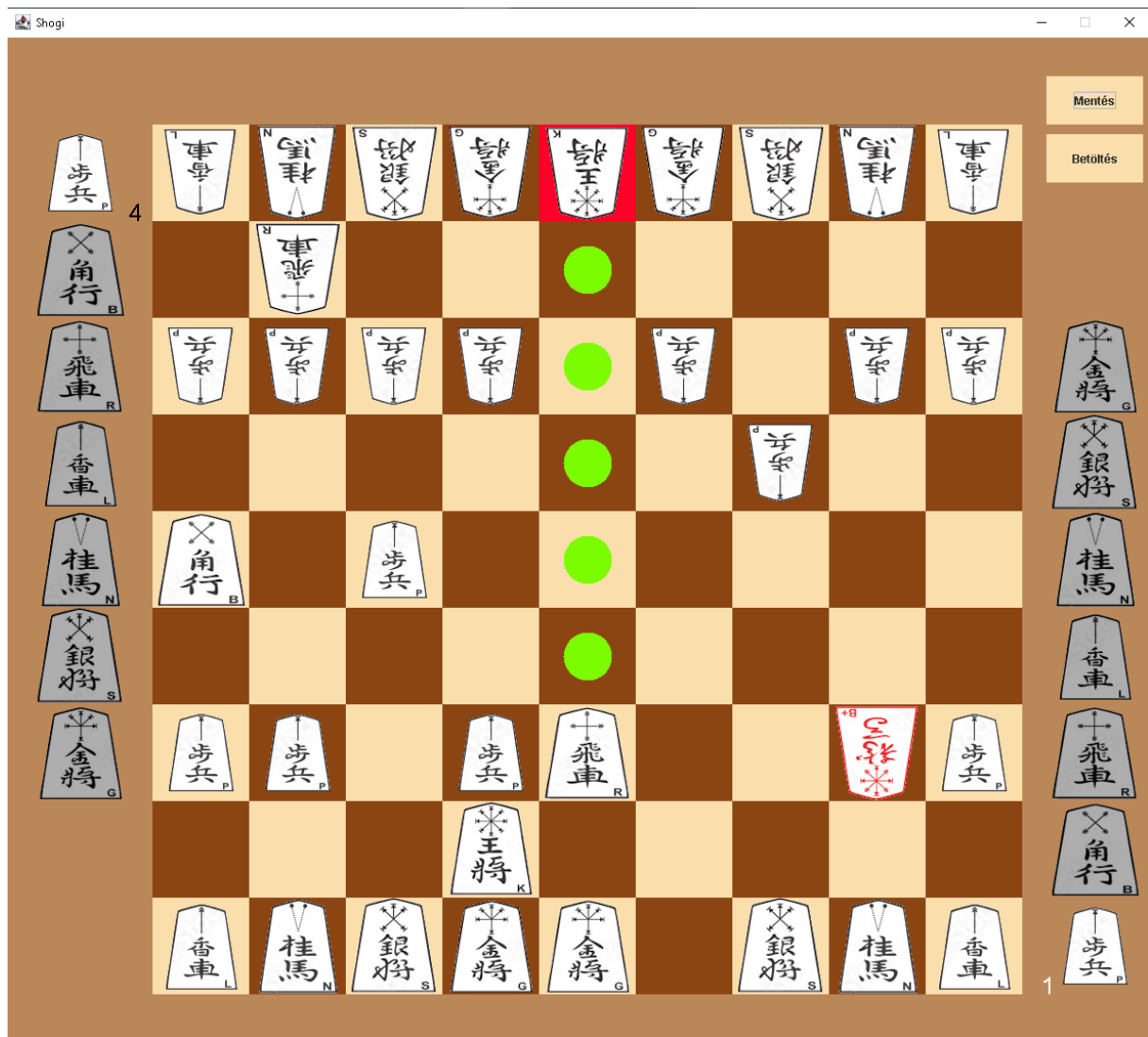
A 2 játékos módban 2 játékos tud egymás ellen játszani.

Azokban az játékmódokban, amelyekben játszik egy játékos a saját körében, ha rákattint egy bábra, akkor megjelenik, hogy hova tud lépni azzal az adott bábuval.



Pirossal pedig azokat a mezőket jelzi, hogy ahova lehetne lépni a bábuval, azonban nem lehet, mert sakkot kapna.

Ha leüt egy bábut, akkor oldalt kifeléredik az adott bábunak a képe. A bábukhoz hasonlóan, ha rákattint, akkor kijelzi, hogy hova tud lépni.



A játékos a játszma végén újra a menüben találja magát, és újra tud kezdeni egy meccset.

Ha egy meccs közben készít egy mentést, akkor az bármikor újra be tudja tölteni, ha rámegy a „Betöltés” gombra.