

Sakkprogram

Működési leírás

Struktúrák

Babu struktúra

Ezzel definiáljuk mind az egységeket mind az üres mezőket a táblán. Elsődleges jellemzőjük a megjelenés, ez alapján különböztetjük meg a bábu típusokat (ez általában a bábu nevének első karaktere pl.: gyalog -> g, huszar -> h, de az üres mező csak egy sima szóköz). Másodlagos megkülönböztető jel a szín integer, ami 1, hogyha fehér- és 0 ha fekete bábúról beszélünk (azonban egy üres mezőnek nincs színe, ezért neki ez az érték -1 lesz). Speciális eseteknél használatos még a megmozdult és az en_passant integerek. Az előbbi az jelzi, ha az adott bábu már megtette első lépését, ilyenkor az értéke 1, egyébként 0. Az utóbbi csak akkor vesz fel 1-et értékeként, ha az adott gyalog kettőt lépett első lépésénél (azaz amikor lehetőség van az en_passant-ra), minden más esetben (egy kör elteltével is) ez az érték 0.

Vektor struktúra

Csupán két integerből áll, x-ből és y-ból. Arra használatos, hogy egy pozíciót (mezőt) ki tudjunk vele jelölni a táblán. Az x a sorokat a z y az oszlopokat jelöli (0-7).

main.c - A FŐMENÜ

Innen érhető el a többi modul. Új játék, Játék betöltése és Utolsó játék visszajátszása opciók közül választhatunk. Hogy melyik modulba megyünk azt a bemenetet megvizsgáló switch szerkezet határozza meg.

rossz_bemenet függvény

Ez a void függvény csupán egy hibaüzenetet takar. Valahányszor hibás bemenetet adunk meg, a gép válaszul ezt a függvényt indítja el, aminek következtében az íródik ki a képernyőnkre, hogy "Hibas bemenet" 2 másodperc erejéig. Nem ad vissza semmilyen értéket.

initialize.c - LÉTREHOZÓ ÉS BETÖLTŐ EGYSÉG

Feladata a bábuk és a tábla létrehozása vagy éppen egy meglévő tábla (játékállás) betöltése fájlból. Akármilyen is csináljunk a főmenüben, ha éppen nem kilépünk a programból, akkor ez a következő modul, ahová eljutunk.

tabla_letrehozasa függvény

Feladata egy 8x8-as Babu-kat tartalmazó mátrix létrehozása, amit feltölt 16 bábuval és 48 üres mezővel (az alap felállítás rendje és módja szerint). A procedúra befejeztével a paramétereként kapott mod integer alapján vezérel minket tovább a game.c vagy replay.c modulokba. Nem ad vissza semmilyen értéket.

babu_letrehozasa függvény

Létrehoz egy Babu-t bemeneti paraméterei alapján, majd vissza is adja. A tabla_letrehozasa egy segédfüggvénye.

tabla_betoltese függvény

Fájlból betölti a táblát (játékállást) majd továbbítja minket a game.c modulba

szam_dekod függvény

A tabla_betoltese egy segédfüggvénye. Feladata, hogy a paramétereként kapott char típusú számot integerben adja vissza.

[game.c - A JÁTÉKVEZÉRLŐ](#)

A modul feladata a sakk szabályszerű lebonyolítása két ember között.

jatek függvény

A függvény feladata, hogy egy switch segítségével beolvassa mit szeretnénk csinálni, majd a választásunkkal kapcsolatos műveleteket levezényelje és betartassa a szabályokat. Paraméterként egy Babu-kat tartalmazó mátrixot (innenről tabla) és egy integert (kor) kap. Van Lepes, Feladas és Kilepes opció, a Lepes opciót leszámítva visszajuttat minket a főmenübe. Ez utóbbi kettő opció esetén a játékállás mentése is megtörténik a saves.c modul meghívásával, viszont Feladas esetén csak a játék visszajátszása lesz elérhető annak betöltése nem. A Lepes opció választásával a program két Vektor-ba tölti a bemeneteket (ez a hagyományos mező jelölés pl.: A2 dekódolásával történik : A -> 7, 2 -> 1). Az első vektor azt a mezőt jelöli, amin az a bábu áll, amivel lépni akarunk, a második pedig azt a mezőt ahová lépnénk vele. Ezután a lépés helyességének ellenőrzése következik. Ha véletlenül nem lenne elfogadható a lépésünk egy hibaüzenetet kapunk minden esetben. A helyes lépéseket viszont a program elvégzi (meghívja rá a lepes függvényt) és a visuals.c modul meghívásával ki is rajzolja az állást a konzolablakba. Ezen kívül ez a függvény kezeli a különleges lépéseket is. Sánc esetén a bástyát áthelyezi a király utáni első mezőre, en-passant esetén törli azt a gyalogot, amit "leütöttünk". Az összes olyan gyalognak pedig, amivel éppen nem léptünk, beállítja az en-passant értékét 0-ra (ugyanis az en-passant lehetősége csak egy körig él). A függvény még ellenőrzi a sakk, matt, patt lehetőségét is. Amennyiben egy látszólag teljesen szabályos lépés megtörténik, de a saját királyunk sakkba kerül illetve abban marad, akkor ezt az utolsó lépést visszavonjuk és nem is kerül mentésre. Matt és patt esetén a program kijelenti a játék végét és visszaküld minket a főmenübe.

sort_beolvas függvény

Egy rekurzív beolvasó függvény dinamikus memóriakezeléssel, ami egy sort elemenként olvas be majd visszatér string-ként. Egyetlen bemenete egy integer, ami a visszajövő string méretét határozza meg, de a függvény meghívásánál ez mindig a 0 értéket kell képviselje (ugyanis ekkor még nem tudjuk milyen hosszú a sor).

lepes függvény

Két különböző Babu pointert kap meg, feladata az "a" pointer által mutatott Babu minden értékét átírni a "b" pointer által mutatott Babu-ba, majd az előző értékeit felülírni az üres mező értékeivel.

szabalytalan_lepes függvény

Egy hibaüzenetet ír ki a képernyőre 2 másodperc erejéig, ha szabálytalanul léptünk.

kiraly_sakkban_van függvény

Egy üzenetet ír ki a képernyőre 2 másodperc erejéig, ha a királyunk sakkban van (általában csak akkor, ha úgy próbálunk meg lépést elkövetni, hogy figyelmen kívül hagyjuk a tény, hogy királyunk sakkban van).

mezo_letezik függvény

Paraméterként egy sztringet kap és megállapítja, hogy a sztring harmadik karaktere a lezáró 0-e, és első és második karaktere szerepel-e a mezőket jellemző koordináták halmazának (A-H és 1-8). Ha a fentiek közül bármelyik nem igaz, akkor 0-t ad vissza és hibaüzenetet ír ki, egyébként 1-et ad vissza.

mezo_ures_vagy_rossz_babu függvény

A függvény paraméterként megkapja egy mező koordinátáit (Vektor), a táblát (Babu mx), és azt hogy hanyadik körben járunk (integer). Ezután ellenőrzi, hogy a tábla koordinátával jelölt helyén nem üres mező található, valamint hogy megfelelő színű bábút "fogunk meg" (a kör száma alapján : páratlan számú kör -> fehér köre). Amennyiben nem üres mezőre vagy ellenfeles színű bábra mutat a koordináta, akkor a függvény 0-t ad vissza, egyébként 1-et és hibaüzenetet.

dekodolas függvény

A függvény paraméterként kap egy sztringet és egy integert (ez utóbbi 0 vagy 1 kell legyen). Az integert indexként használva vizsgálja meg a sztring első vagy második karakterét. A procedura során ha a vizsgált karakter A és H közé esik, akkor 7 és 0 közötti számot ad vissza, oly módon, hogy A -> 7, B -> 6 és így tovább, ha viszont a vizsgált karakter szám, akkor annál egyel kisebbet ad vissza (minthogy a táblán 1-el kezdjük a számlálást itt viszont 0-val)

mezo_tamadott függvény

A bemeneti paraméterek ezúttal egy bábu helyzete (Vektor), színe (int) és a tabla. A függvény feladata megmondani azt, hogy az adott bábu helyére bármelyik ellenfeles bábu szabályosan lépni tud-e, azaz a bábunk támadásban van-e. Ebben segít neki a lepes_vizsgalat függvény. Ha támadva van 1-et ad vissza, különben pedig 0-t.

sakk függvény

A bejövő paraméterei az egyik fél színe (int) és a tabla. Feladata hogy megmondja az adott színű király sakkban van-e. Ehhez először megkeresi a királyt a táblán, majd meghívja a mezo_tamadott függvényt. Ha a király támadva van (mezo_tamadott 1-et ad vissza), akkor ő is 1-et ad vissza, egyébként pedig 0-t.

patt függvény

A bejövő paraméterei az egyik fél színe (int) és a tabla. Megmondja, hogy adott színű bábuk közül tud-e valamelyik szabályos lépést tenni. Ehhez segítségül hívja a lepes_vizsgalat függvényt. Amennyiben egyik bábu sem tud lépni 1-et, különben pedig 0-t ad vissza.

gyalog_atvaltozas függvény

A bejövő paramétere egy darab Babu-ra mutató pointer. Az a feladata, hogy megkérdezze a felhasználót, milyen egységre váltsa le az éppen átváltozó gyalogot, majd a pointer által mutatott Babu (a gyalog) megjelenés attribútumát megváltoztatja a kapott bemenet függvényében.

lepes_vizsgalat függvény

A függvény paraméterként két mezőt (Vektor), a táblát (Babu mx) és még egy integert kap sanc néven. A függvény feladata, hogy megállapítsa a különböző lépésekről, hogy helyesek e. Ezt úgy éri el, hogy egy Babu megjelenés attribútuma alapján tovább küldi egy bábuspecifikus alfüggvénynek (gyalog_lep, huszar_lep stb.). Amennyiben helyesek visszaad egy nullánál nagyobb számot, hogy pontosan mit az attól függ történt-e speciális lépés (pl.: gyalog kettőt lép, -átváltozik stb.). A speciális lépéseket szintén az alfüggvények vizsgálják. (Megjegyzés: a vezér lépéseinek ellenőrzéséhez nem készült külön alfüggvény, mert az csak olyan, mintha egyszerre hívnánk meg a futó és a bástya lépéseire tartozó alfüggvényeket.)

gyalog_lep (al)függvény

Paraméterként két mezőt (Vektor) és a táblát (Babu mx) kapja. Feladata ellenőrizni, hogy az adott gyalog bábuval helyesen lépünk-e. Ha a lépés helyes és kettőt lépünk előre, akkor a függvény 2-t ad vissza (ekkor a jatek függvény a át fogja állítani a bábu en_passant attribútumát 1-re). Ha a lépés helyes és elérünk a gyaloggal az utolsó sorba, 4-et ad vissza, (ekkor a jatek függvény meg fogja hívni a gyalog_atvaltozas függvényt), azonban ha csak egyet lépünk előre és nem az utolsó sorba, akkor 5-öt ad vissza. Ha a gyaloggal en-passant szabály szerint ütünk, akkor 3-at ad vissza a függvény (ekkor a jatek függvény "leveszi a tábláról" a felesleges darabot (üres mezőt rak a helyére), viszont ha csak egy átlagos ütésről van szó, akkor 1-et ad vissza. Különben 0-t.

huszar_lep (al)függvény

Paraméterként két mezőt (Vektor) és a táblát (Babu mx) kapja. Feladata ellenőrizni, hogy az adott huszar bábuval helyesen lépünk-e. Amennyiben a pozíció- és a célmező távolsága az egyik tengelyen 2 és a másikon 1, akkor a lépés helyes és 1-et ad vissza a függvényünk. Különben 0-t.

futo_lep (al)függvény

Paraméterként két mezőt (Vektor) és a táblát (Babu mx) kapja. Feladata ellenőrizni, hogy az adott futó bábuval helyesen lépünk-e. Amennyiben a futónk átlósan lép és nem ugrik át egy egységet sem, (valamint ahogy az összes többinél is a célmezőben nem egy azonos színű egység áll) akkor a lépés helyes és 1-et ad vissza a függvényünk. Különben 0-t.

bastyalep (al)függvény

Paraméterként két mezőt (Vektor) és a táblát (Babu mx) kapja. Feladata ellenőrizni, hogy az adott bástya bábuval helyesen lépünk-e. Amennyiben a bástyánk csak az egyik tengely szerint mozog és nem ugrik át egy egységet sem, (valamint ahogy az összes többinél is a célmező egyezik meg a pozíciómezővel) akkor a lépés helyes és 1-et ad vissza a függvényünk. Különben 0-t.

kiraly_lep (al)függvény

Paraméterként két mezőt (Vektor), a táblát (Babu mx) és még egy integert kap sanc néven. Feladata ellenőrizni, hogy a királlyal helyesen lépünk-e. Amennyiben a királyunk csak egyet lép, bármilyen irányban is legyen az, akkor a lépés helyes és 1-et ad vissza a függvényünk (hogy a királyunk lépés után sakkban van-e itt nem ellenőrizzük, csak a jatek függvénybe való visszakerülés után lesz meghívva a sakk függvény, és abban az esetben, ha tényleg sakkban lesz a királyunk mozgás után, akkor lépésünket semmisnek tekinti és visszaállítja az eredeti állást.) Amennyiben sáncolni szeretnénk a királlyal (kettőt lépve valamelyik bástya felé alapállásból), akkor a függvény még ellenőrzi a sanc integert is, ami akkor 1, ha a királyunk sakkban van (ezt az integert még a jatek függvényben deklaráljuk). Ha tudunk sáncolni akkor a függvény 6-t ad vissza (ilyenkor a jatek

függvény még elhelyezi a bástyát is a király másik oldalára). Helytelen lépés esetén 0-t ad vissza a függvény.

matt függvény

A matt függvény a királyunk színét (int) és a táblát (Babu mx) kapja meg paraméterként. Feladata hogy meg tudja mondani, van-e olyan lépés, amelyet ha elvégzünk királyunk nem lesz sakkban. Ehhez a sakk és a lépés függvényeket is segítségül hívja. Amennyiben nincs olyan lépés, amivel megvédehetnénk királyunkat a függvény 1-et ad vissza.

[visuals.c](#) - A MEGJELENÍTŐ

Feladata, hogy a Babu mátrixokat tényleges tábla formájában kirajzolja a konzolablakba

rajzol függvény

Bejövő paraméterként egy táblát (Babu mx) és egy módot (integer) kap. Feladata a tábla kirajzolása a bábukkal és az oszlopok és sorok számozásával együtt úgy, hogy ha a mód egy páros szám, akkor a sorok számozása felfele növekszik, ellenkező esetben pedig csökken, és ehhez igazodnak a bábuk is (azaz mindig ahhoz fordul a tábla akinek éppen a köre van).

bejatszas függvény

Bejövő paraméterként egy táblát (Babu mx) és a kör számát (integer) kapja meg. Feladata a tábla kirajzolása a bábukkal és az oszlopok és sorok számozásával együtt úgy, hogy alá kiírja, hanyadik körben járunk, itt nem fordul a tábla körönként. (Ezt a függvényt csak visszajátszás esetén hívjuk meg).

[saves.c](#) - A MENTŐ EGYSÉG

Feladata a játékállások fájlba mentése.

allas_mentese függvény

Paraméterként kapja a táblát (Babu mx) és a módot (int). Ugyanabba a fájlba menti el az egymást követő játékállásokat append módon. Minden kör végén meg van hívva a függvény a game.c-ből, azonban a fájl tartalma, amiben dolgozik ("mentett_allas.txt") olykor törlődik is (új játék kezdésénél vagy egy játék feladásánál). A függvény először azt írja fájlba, hogy hanyadik körnél járunk, majd egymás alá az összes bábu helyzetét és attribútumait, majd a kör végeztével az utolsó sorba még rak egy '.' karaktert.

lepesek_mentese függvény

Feladata átmásolni a "mentett_allas.txt" tartalmát a "mentett_lepesek.txt"-be. Ezt azért kell megtennie, a betölthető és visszajátszható játékokat el kell választani egymástól. Amíg egy befejezett játékot betölteni már nem lehet visszajátszani még igen, azonban olykor ennek a fájlnak a tartalmát is törölli a program amikor új játékot kezdünk.

[replay.c](#) - A VISSZAJÁTSZÓ EGYSÉG

Feladata hogy egy elkezdett játék összes lépését vissza tudja tölteni fájlból, és megjeleníteni ezeket az állásokat képernyőn.

visszajatszas függvény

Paraméterként kapja a táblát (Babu mx). Az initialize.c modulból hívódik meg, eleinte csak egy üres táblát kap, amit meg is jelenít a bejatszas függvény segítségével. Megszámolja hány kör ment le a

legutolsó játék alkalmával (ez maximum 999 lehet) Feladata, hogy beolvassa a visszajátszásban az előző vagy a következő körre szeretnénk lépni, és ezt a kérést továbbítja a megjelenítőnek (amennyiben ez lehetséges).

atvalt függvény

A `kovetkezo_kor` segédfüggvénye. Bemeneti paraméternek egy darab 16 bites sztringet kap. Feladata, hogy a sztring elején lévő (maximum 3 jegyű) számot integer formában adja vissza. Ha a szám rövidebb akkor a függvény el fogja érni a lezáró '.' karaktert, és ezzel vissza is adja a számunkat.

kovetkezo_kor függvény

Bemenetként csupán a kör számát (int) kapja meg. Feladata kikeresni a "mentett_lepesek.txt" fájlból a következő kört, legyen az korábbi vagy későbbi lépés következménye. Majd a kikeresett állást Babu mátrix formában továbbítani a `bejatszas` függvénynek.