

Takács Bálint

B8I2BV

Programozói dokumentáció

Felépítés

A programom összesen 5 modulból áll, felhasználok ezek mellé még a debugmalloc header fájlt. Ezekben a mentés modulnál a dirent könyvtárat használom ahhoz, hogy mappát is tudjak kezelni. Ezen kívül a string könyvtárat használom a string kezeléshez.

Modulok:

Main: Létrehozza a játékot és elindítja a menü nézetet.

Nezet: 2 függvényt tartalmaz, a kezdő menü betöltését végzi el, illetve a játék alap adatainak a beolvasását végzi el.

Mentes: 3 függvényt tartalmaz, a játék mentését, a mentett fájlok felsorolását, majd ebből kiválasztva az adott mentett játék betöltését végzi el.

Jatek: 13 függvényt tartalmaz, a modul a játék menet vezénylését végzi el, felállítja a kezdő pályát, megadja a következő játékos nevét, kirajzolja a pályát a konzolra, beolvassa a játékos lépését majd ezt meglépi és elhelyezi a pályán, majd kivizsgálja, hogy ez egy nyerő lépés volt e, ha igen akkor véget vet a játéknak, clearli a pályát és visszaállítja az értékeket. Ha nincs vége váltja a játékost és a bábút, kiértékeli a legjobb értéket a gépnek majd ezt meglépi. A kiértékelés folyamatában ellenőrzi, hogy van a játékosnak 2 vagy 3 bábuja egymás mellett. Minden játékos lépés után felkínálja a lehetőséget, hogy mentsünk, vagy kilépünk a menübe. Ellenőrzi azt is, hogy ha be telt a pálya akkor döntetlen. Itt hozom létre a főbb adatszerkezeteket is melyek:

Jatek: Struktúra melyben 12 értéket tárolok. A kétszeres indirekciót alkalmazó Babuk típusú pályát, a következő Bábut, azt, hogy vége van a játéknak, illetve a játékosok nevét. Ezekon kívül tartalmazza még a pálya méreteit, tehát a sort, és oszlop hosszt. Eltárolom azt is hogy ki kezdett, illetve értéket adok a gépnek és a játékosnak, és az adott súlyozást is itt tárolok. A játék végét pedig bool értékben tárolok. Tehát a játék összes fontos tulajdonságát tartalmazza.

Pozicio: Struktúra melyben 2 integer típusú értéket tárolok. Ezek adják a pályán a koordinátákat.

Babuk: Enum típusú struktúra, elemei az üres, kör, x enumok.

Memoriakezeles: 1 felszabadító függvényt tartalmaz, ami a pályát szabadítja fel.

Függvények:

Nezet-ben:

Kezdo_nezet: paraméterei: Jatek struktúra, sor , hossz

Először is kirajzolja a menüt,melyben lehetőségünk van:

- 1. Korábbi játékok betöltésére
- 2. Új játékra
- 9. Kilépni

Switch-et használva van megvalósítva.

1-est nyomva meghívja a betolt_mentes függvényt.

2-est nyomva bekéri a pálya szélességét és magasságát, majd dinamikus memóriakezeléssel lefoglalja a pályának a memóriát. Ezután meghívja a jatek_nezet függvényt.

9-est nyomva exit(0)-t hív meg.

Jatek_nezet: paraméterei: Jatek struktúra

Bekéri, a játékos nevét, illetve melyik bábuval szeretne játszani, ezután bemásolja a játékos nevét a jatek struktúrába, meghívja ezután a jatek függvényt.

Mentes-ben:

Betolt_mentes: paramétere a Jatek struktúra.

Dirent.h-t megnyitja a mappát amiben a mentések vannak, ennek felsorolja elemeit majd bekéri a felhasználótól hogy melyik játékot akarja megnyitni. Ezt továbbadva meghívja a jatek_betolt függvényt.

Jatek_betolt: paraméterei a Jatek struktúra, fajlnev string.

Mappát vált először a chdir() függvénnyel, ezzel bemegy a mentések mappájába, majd megkeresi az átadott fajlnevet és megnyitja. Ebből kiolvassa az elmentett játék adatait majd ezt betölti a paraméterként adott Jatek strukturaba, dinamikusan memóriát foglal és elindítja a játékot.

Mentes: paramétere a Jatek struktúra.

Generál 1-100 között egy random számot, melyet az amoba stringhez hozzáillesztve egy új egyedi fájlnevet hoz létre, ezt a filet megnyitja és beleírja a menteni kívánt játék tulajdonságait. Ezután felszabadítja a foglalt memóriát és visszalép a kezdő menübe.

Jatek-ban:

Kezdes: A következő bábut körre állítja be majd végig iterál a pálya celláin és üresre állítja őket.

Jatek_kovetkezo_neve: Megadja annak a játékosnak a nevét, aki a következő lépést teszi majd.

* @return A játékos neve.

Következo: Babukjparaméter. J től függően kör és x közt váltogatja a következő Bábut, visszatérési értéke 1 vagy 2.

Nyert_e: paraméterei: Jatek struktúra, P pozíció, Babu c.

Létrehoz egy db számlálót melyet 1-re állít az elején, mert feltételezi hogy le tudta tenni a bábut a meghívott pozícióra.

Ezután a 4 vonal mindegyikén a két ellentétes irányba addig fut amíg olyan bábut talál mint az övé, ha talál növeli a db változót 1-el, ha az ellentétes irányok után ez a db nem éri el az 5 öt, akkor visszaállítja a darabot 1 re és megy a következő vonalra. Ha a db 5 akkor visszatérési érték true, ha nem akkor false.

Jatek_lep: paraméterei: Jatek struktúra, P pozíció.

Lép a megadott játékos a megadott pozícióra.

* @return true, ha ezzel a lépésével megnyeri a játékot.

Jatek_allas: szokásos Jatek struktúra paraméterrel.

Megjeleníti a pálya összes celláját és elemeit a képernyőn.

Pozicio_beolvas: Beolvassa a koordinátákat a billentyűzetről

* @return A beolvasott koordináták pozícióját adja vissza.

Dontetlen_e: paraméter a Jatek struktrúra

Végigfut a pálya összes celláján, ha talál üreset return false ot ad, ha nem akkor return truet, tehát ellenőrzi hogy betelt e minden cella.

Egymas_mellett_harom: paraéterek: Jatek struktúra, **érték tábla, y ,x koordináták, keresett ertek.

A megkapott koordinátákból 4 irányba indulva keresi , hogy van 3 keresett Babu, ha van akkor ennek a hármasnak a két oldalára értéket ad az érték táblán.

Egymas_mellett_ketto: Tulajdonképpen ugyan az mint az egymás mellett három, csak itt kettőre keres és kisebb értéket ad a szélekre.

Legjobb_lepes: Jatek struktúra a paraméter.

A legkomplexebb függvénye a játéknak. Ez számolja ki hogy melyik a pályán a legjobb lépés. Először a pálya méretével hasonló nagyságú segéd táblát hoz létre, melyet feltölt 0 értékekkel. Ezután bármikor mikor megvan hívva, átfut a segéd táblán, és ahol az igazi pályán Bábu van letéve, ott az értéket egy nagy - számra állítja, körülötte pedig egyre kisebb értékeket oszt ki a cellákra, 1 es környezetében 10- ez a szám és minnél messzebb megy annál kisebb lesz. Ha a játékot a gép kezdte, akkor súlyozással nagyobb értékeket ad a gép lépéseire így inkább támadó taktikákat fog alkalmazni ilyenkor a gép. Kikeresi ezek után a maximum értéket a segéd táblán, és visszatérési értéknek adja a legnagyobb érték koordinátáit.

Gep_lep: Jatek struktúra a paraméter.

Beolvas a legjobb_lepes függvény segítségével egy pozíciót, majd erre beteszi a gép bábuját. Ha ezzel a lépéssel megnyeri a játékot akkor a visszatérési érték true, ha nem akkor átállítja a következő Bábut majd false-t térít vissza.

Jatek:

Jatek_kovetkezo_neve függvény segítségével meghatározza a következő játékost, majd ezután meghívja az adott játékos lépés függvényét, ezt ezután ellenőrzi, ha a visszatérési érték true akkor megszakítja a játékot, ha nem a játék folytatódik. Játékos lépés előtt felhossa az opciókat, a játékos kiléphet a kezdőmenübe, vagy mentheti is az adott játékát.