

NHF dokumentáció

Tóth Mihály Balázs – OAYOF1

A program egy amőbajáték menetét valósítja meg.

A forrásfájlok a *main.c*, a *hatter.c* és a *pvpjatek.c*.

A *main.c* tartalmazza a játék alapvető működését, a másik két fájl a *main*-ben használt függvényeket definiálja. A *main* egy keretrendszert biztosít és a főmenü első szintjét tartalmazza. Itt van deklarálva a *map* globális változó is, ez tárolja az összes adatot a játék állásáról és feltételeiről.

A *map* egy *palya* típusú struktúra, felépítése a következő:

- *hely*: egy *char* típusú duplapointer, a dinamikusán foglalt mátrixra mutat
- *meret*: egy integer, a pálya oldalhosszát tárolja
- *gyozelem*: egy integer, azt tárolja, hogy hány elemet kell elhelyeznünk egy sorban a győzelemhez
- *pvp*: egy logikai változó, azt tárolja, hogy játékosok közti vagy gép elleni játékról van-e szó (ha igaz, akkor játékosok, ha hamis, akkor egy játékos és a gép játszik egymás ellen)
- *soron*: azt tartja számon, hogy ki van éppen soron lépésben (ha 1, akkor player 1, ha 2 akkor player 2)
- *szabad*: egy integer, azt számolja, hogy hány szabad mező van még a pályán

A *hatter.c* függvényei a következők

- *kilep*: nem kap paramétert, visszakérdez a felhasználónak, hogy biztos ki akar-e lépni, ha a menüből a kilépés opciót választotta
- *palyafoglal*: paraméterül kap egy *palya* struktúrát annak oldalhossza szerint dinamikusán lefoglal egy pointertömböt, aminek elemeit szintén dinamikusán foglalt karaktertömbökre állítja
A dinamikus területen lévő dinamikus cím értéke kétszeres indexeléssel érhető el
A függvény kijelzi, hogy sikeres volt-e a memória foglálás és visszaadja az új struktúrát
- *palyafree*: felszabadítja a paraméterként kapott pálya által foglalt memóriát
- *ujjatek*: nem kap paramétert, létrehoz egy *palya* típusú változót, beállítja a struktúra megfelelő adatait a felhasználó által megadott bemenetek alapján, majd a *palyafoglal* függvény segítségével lefoglalja a pályának a memóriát és a kész struktúrát visszaadja
- *palyakiir*: paraméterül kap egy *palya* struktúrát és kiírja a képernyőre az aktuális táblát
- *palyament*: paraméterül kap egy *palya* struktúrát és elmenti azt egy felhasználó által megadott nevű txt fájlba a specifikációban leírtak szerint
- *palyabetolt*: nem kap paramétert, futása során bekér egy fájlnevet és abból a fájlból betölti egy *palya* struktúrába az adatokat, ezt adja vissza

A *pvpjatek.c* függvényei a következők:

- *soron_valt*: cím szerint átvesz egy *palya* struktúrát, híváskor átváltja a struktúra *soron* adatsorát

- *szabad*: cím szerint átvesz egy palya struktúrát és két egész számot, ellenőrzi, hogy az adott pályán a megadott koordináták szerinti lépés érvényes-e, ha igen, igazat, egyébként hamist ad vissza
- *place_x*: cím szerint átvesz egy palya struktúrát és két egész számot. A *szabad* függvény segítségével ellenőrzi, hogy szabad-e oda karaktert elhelyezni, és ha igen, akkor elhelyez egy X-et és a *soron_valt* függvény segítségével átváltja a soron lévő játékost az ellenfélre
- *place_o*: működése a *place_x*-el azonos, azzal a különbséggel, hogy nem X-et hanem O-t helyez el a pályán
- *playertochar*: cím szerint átvesz egy palya struktúrát, az aktuálisan soron lévő játékos karakterét (X vagy O) adja vissza
- *kelet*: cím szerint átvesz egy palya struktúrát, ezt bejárva megkeresi az első karaktert, amit az aktuálisan soron lévő játékos rakott le, majd ettől a tömbben „keletre” haladva (vízszintesen jobbra) megszámolja, elég azonos karakter van-e már egy sorban a játék megnyeréséhez, ha igen, akkor igaz értékkel tér vissza, ha nem akkor hamissal
- *delkelet*: működése a *kelet*-tel azonos, azzal a különbséggel, hogy „dékeletre” (átlósan jobbra lefelé) keresi az elemeket
- *del*: működése az előbbiekhöz hasonlít, ez „délre” (függőlegesen lefelé) keres
- *delnyugat*: működése az előbbiekhöz hasonlít, ez „délnyugatra” (átlósan balra lefelé) keres (ezekkel a függvényekkel az összes irányt lefedtük, mivel az egész tömböt bejárjuk vele)
- *randomkezd*: cím szerint átvesz egy palya struktúrát, pseudorandom módon meghatározza, hogy ki kezdjen
- *nyert*: cím szerint átvesz egy palya struktúrát, meghatározza, hogy az aktuális állásban van-e nyertes, ha igen, akkor igaz értékkel tér vissza, ha nem akkor hamissal
- *betelt*: cím szerint átvesz egy palya struktúrát, meghatározza, hogy betelt-e az összes mező, ha igen, akkor igaz értékkel tér vissza, ha nem akkor hamissal
- *vege*: cím szerint átvesz egy palya struktúrát, eldönti, hogy vége van-e a játéknak, és ha igen akkor hogyan:
 - ha még nincs vége, akkor -1-et ad vissza
 - ha döntetlen (betelt a tábla) akkor 0-t
 - ha player 1 nyert játékos ellen akkor 1-et
 - ha player 1 nyert gép ellen akkor 11-et
 - ha player 2 nyert játékos ellen akkor 2-t
 - ha player 2 nyert gép ellen akkor 12-t
- *pvp*: cím szerint átvesz egy palya struktúrát, a függvény feladata a játékosok között zajló játék lebonyolítása:
A *randomkezd* függvény segítségével randomizálja, hogy ki kezdi a játékot, a struktúra *pvp* értékét igazra állítja, ezután kiírja az üres pályát, és hogy ki van soron. A játékos a lépésének koordinátáit beolvassa, majd ennek megfelelően elhelyezi a karaktert a kívánt helyen (ha szabad). A lépéseket a kiírástól a karakter elhelyezésig addig ismétli, amíg be nem telik a pálya, vagy valaki nem nyer.