

Specifikáció

Memóriajáték

Adott méretű táblán elem-párok vannak elrendezve a játékos számára nem látható módon. A játékos egy lépésben kiválaszthat két cellát, amelynek a tartalmát megtekintheti. Cél a párok megtalálása. Ha egy párt megtalál a játékos, azok eltűnnek a tábláról.

Program felépítése

Felhasználói dokumentáció

A program parancssorról bekéri a felhasználótól, hogy mit szeretne csinálni (megnézni a rekordokat és/vagy játszani, esetlegesen kilépni). Rekordok megtekintése esetén beolvassa soronként a records.txt szövegfájl adatait, és kiírja a képernyőre. A játék elindítása után bekér egy pályaméretet, amely csak 2 és 10 közötti páros szám lehet, mivel az ASCII kódtáblában nem lenne elég karakter 10x10-esnél nagyobb pályára.

Programozói dokumentáció

Bekérések

A program figyel arra, hogy a bekért adatok egyezzenek azzal, amit kér. Ilyen például a méretmegadásnál, hogy számot lehet csak beadni, mivel betű méretű pálya nem létezik, ezt egy *is_digitstr* nevű függvénnyel ellenőrzi is.

A koordináták megadásánál figyel arra, hogy mindenképp „a1” formátumú legyen, így az „1a”, „a”, „1” megadásokat nem fogadja el, és újra kéri. Arra is figyel, hogy a már megtaláltakat ne kérhesse be újra, valamint hogy a pályáról kimutató koordináták helyett is újat kér be.

Fájlkezelés

A program minden megnyert játék után bekér egy nevet, majd kiírja azt egy szövegfájlba (*records.txt*, ami ha nem létezik, akkor létrehozza azt) a pályamérettel és a lépésszámmal együtt. Ezen kívül a főmenüből elérhető a rekordok bekérése funkció, mely beolvassa az említett szövegfájl tartalmát, és kiírja a képernyőre a felhasználó számára.

A főprogram

A *main.c*-ben hozom létre a tömböt, melyben a karaktereket tárolom. A méretmegadás után (mely csak 2-10 közötti páros szám lehet) itt foglallok neki memóriaterületet a *calloc*-cal, mert az a tömb minden elemét 0-ra állítja. Ezek után feltölteti a tömböt, majd amíg a játékos meg nem találja a párokat ($\text{meret}^2/2$ db). A *main.c*-ben kezelem még a fájlt is, mind a kiíratást, mind a beolvasást egyaránt.

A többi függvény

Ezeket a *memory.c*-ben valamint a *memory.h*-ban hoztam létre. Összesen 4 függvénnyel dolgozom.

Az *is_digitstr* megvizsgálja, hogy a beadott karaktersorozat minden eleme számból áll-e, avagy sem. Ezzel szűrtem ki, hogy a bekért pályaméret ne lehessen betű.

A *feltoltes* rögtön összekeverve feltölti az ASCII kódtáblából karakterekkel a dinamikusan lefoglalt tömböt. Random helyre kéri be, figyelve arra, hogy minden karakter pontosan kétszer szerepeljen a tömbben.

A *kiiratas* nevű függvény szimplán kiírja a képernyőre a felhasználónak a pályát. Azokat az elemeket, amelyek már szerepeltek, ott az 1-es kódú ASCII karaktert írja ki, ami jelen esetben egy ☺. Természetesen e helyett kiírathatunk ' ' -t is, ilyenkor csak meg kell cserélni a kiíratás részben a kikommentezett `//printf("%c ", ' ');` és a `printf("%c ", tomb[sor*meret+j]);` részt.

A *vizsgalas* is tartalmaz kiíratás részt, mivel ott kiíratjuk, hogy a bekért két elem egyrészt hol is van a pályán, másrészt hogy melyik karakterek azok. Alapvetően ez a függvény bekér két koordinátát, majd megvizsgálja azokat, hogy ne legyenek egyformák, se olyanok, amelyeket már megtaláltunk korábban, valamint hogy pályán belüli elemekre gondoljunk. Ezek után átírja az eredeti tömbben a karaktereket 1-re, ha a két megtalált karakter egyforma.

Visszatérése egyforma esetén 1, nem egyforma esetén pedig 0. Visszatérési értéke alapján döntöm el a *main.c*-ben, hogy találtunk-e újabb elemet. Ha igen, akkor a megtaláltak számát növelem.