

Scrabble játék

A program felépítése

A program fordításához a szabványos könyvtárakon kívül semmi nem szükséges. A program forráskódja a következő modulokra van bontva: szótár, játék, játékos, pálya és main. A szótár, játékos és pálya modulok feladata az adatok alacsonyabb szintű kezelése. Ehhez a következő struktúrákat használják:

'betukeszlet': Ez a struktúra a szótár modulban található, feladata, hogy a hagyományos játék során használt „betű zsák” szerepét betöltse. Tehát ebben tárolódik, hogy milyen betűk vannak, azoknak mennyi az értéke, és hogy mennyi van még az egyes betűkből. Az adatokat dinamikus tömbben tárolja. A struktúrához tartozik fájlból beolvasó és kiíró függvény, és törlő függvény, a dinamikus memóiafoglalás miatt. Van egy függvény, ami egy adott betűhöz rendeli az eltárolt értékét és egy olyan, ami véletlenszerűen visszaad egy betűt a készletből. Ehhez az egyes betűk valószínűségét aszerint súlyozza, hogy mennyi van még a készletben.

'szotar': A szótár modul része. Itt tárolja a program azokat a szavakat, amiket elfogad, mint értelmes szavak, amiket le lehet rakni. A struktúrához tartozik fájlból beolvasó és törlő függvény. Ezeket az adatokat nem kell fájlba kiírni, mert a program nem változtatja meg őket. Ehhez tartozik egy függvény, ami megmondja egy adott szóról, hogy az benne van-e a szótárban. Ehhez bináris keresést használ, tehát a fájlban rendezetten, betűrendben kell, hogy legyenek a szavak. Egy függvény azt is kiszámolja, a 'betukeszlet' struktúrát felhasználva, hogy egy teljes szó hány pontot ér (az akciómezők beleszámolása nélkül).

'vektor': A pálya modul része. Ez egy 2 dimenziós vektor struktúra, a pálya struktúra egyes függvényei használják, hogy egyszerűbb legyen a 2 dimenziós tömbök indexelése, valamint irányvektorként használja a program, hogy a vízszintes és függőleges kirakásokat közös függvények tudják kezelni. Tartozik hozzá összeadó, -1 -el szorzó és összehasonlító függvény.

'palya': A pálya modul része. A pálya összes adatát tárolja, 2 dimenziós tömbökben. Tehát a pálya méretét, a betűket, és az akciómezők értékét. Az akciómezők adatai két tömbben vannak eltárolva, az egyik a szavakra vonatkozó szorzókat, a másik a betűkre vonatkozókat tárolja. Tartozik hozzá fájlból beolvasó, kiíró és törlő függvény is. A 2 dimenziós tömbök létrehozására és törlésére külön függvényeket használ. Egy függvény megmondja, hogy egy adott vektor helyén milyen betű van. Egy másik eldönti, hogy egy vektor a pálya korlátain belül van-e. Egy harmadik pedig egy kapott helyen és irányban mondja meg, hogy mi az ott lévő leghosszabb szó. A fenti függvényeket a vektor struktúra felhasználásával oldja meg.

'jatekos': Ez a struktúra egy játékos adatait tárolja, azaz a nevét, a pontszámát, és hogy éppen milyen betűk vannak a betűtartójában. Egy függvény képes az adott tömböt feltölteni játékosokkal egy adott fájlból. Egy függvény pedig kiírja azokat.

A játék modul feladata, hogy az előző struktúrákat használva oldjon meg bonyolultabb feladatokat. Itt találhatóak a következők:

'parancs': Ez nem egy struktúra, hanem egy enumeráció, ami az egy körben választható opciókat képviseli.

'kirakas': Egy adott körben kirakott szó adatait tartalmazza, tehát a betűket, a helyét az irányát és joker használata esetére, a konkrét szót.

'palya': A fő struktúra, amely a fentieket struktúrákat egybefoglalja, és magasabb szintű függvényeket valósít meg. Van szokványos fájlból beolvasó, kiíró és törlő függvény, azonban itt egyetlen fájl helyett mappákról van szó. Képes kiírni a játék adatait, kirajzolni a pályát, bekérni a kirakást, ellenőrizni azt, és kezeli a játékosokat.

A main modulban csak a main függvény található, ez a 'palya' struktúrát használva irányítja a játék menetét.

A függvények részletesebb leírása

szotar.h:

- **bk_beolvas:** A betűkészlet beolvasása. A beolvasott adatokat a paraméterlistán adja vissza. Visszatérési értéke 0, ha nem volt hiba és 1, ha volt. A beolvasott betűkészletet a hívónak kell törölnie.
- **bk_kiir:** A betűkészlet kiírása fájlba. Argumentumai a fájl neve és a betűkészlet maga.
- **bk_torol:** A betűkészlet törlése. Argumentuma a betűkészlet.
- **betu_ertek:** Argumentumként egy betűt és egy betűkészletet kap. A betűkészletben tárolt értéket adja vissza.
- **random_betu_kivesz:** Visszaad egy véletlen betűt a készletből, amit kivesz onnan. A betűszám alapján határozza meg a betűk valószínűségét.
- **szotar_beolvas:** Argumentumként a fájl nevét és a feltöltendő szótárat kapja. A szótárat a hívónak kell törölni. Visszatérési értéke 0, ha sikeres volt, különben 1.
- **szotar_benne_van:** Eldönti, hogy egy kapott szó benne van-e a kapott szótárban. Logikai értékkel tér vissza.
- **szotar_torol:** Törli a szótárat.
- **szo_ertek:** Egy szót és egy betűkészletet kap, meghatározza a szó értékét a benne lévő betűk alapján. Nem veszi figyelembe az akciómezőket. A betűk és a szavak értéke mindig egész szám.

palya.h:

- **vekt_osszead:** A kapott két vektor összegével tér vissza. (Ami szintén vektor).
- **vekt_negativ:** A kapott vektor ellentétvektorát adja vissza.
- **vekt_egyenlo:** Eldönti, hogy a két vektor egyenlő-e. Logikai értéket ad vissza.
- **tomb2d_letrehoz:** Létrehoz egy két dimenziós dinamikus tömböt, Az argumentumai a tömb szélessége, magassága és az elemeinek a mérete. (Bármilyen típusra használható, visszatérési értéke void ** típusú). A hívó feladata törölni.
- **tomb2d_torol:** Törli a tomb2d_letrehoz által létrehozott tömböt. Ehhez a tömb elejére mutató pointer és a tömb szélességét kell megadni.
- **palya_beolvas:** Argumentumként a fájl nevét és az üres pályát kapja meg, amibe beolvassa az adatokat. Visszatérési értéke 0, ha sikeres volt, különben 1. A pályát a hívónak kell törölni.
- **palya_kiir:** A kapott nevu fájlba írja a pálya adatait.
- **palya_torol:** A pálya törlése.
- **palya_betu:** Argumentumként a pályát és egy vektort kap. A pályán a vektor által meghatározott helyen lévő betűt adja vissza. A programban sok ciklus vektorokat használ, hogy a vízszintes és a függőleges irány egyben legyen megvalósítva, ezekhez hasznos ez a függvény.
- **palya_belul_van:** Eldönti a kapott vektorról, hogy bent van-e a kapott pálya korlátai közt. Logikai értékkel tér vissza.
- **palya_szo_itt:** Argumentumként egy pályát, egy helyvektort, egy irányvektort és egy sztringet kap. Eldönti, hogy a pályán a kapott helyen és irányban mi a leghosszabb szó. Ezt a sztringet tölti.

jatekos.h:

- **jatekosok_beolvas:** Argumentumai a fájl neve, a feltöltendő tömb, egy pointer a játékosok számára és egy a soron következő játékos indexére, valamint a tömb maximális mérete. A fájlból beolvassa a játékosok adatait, hogy mennyi van belőlük és hogy hanyadik következik. Nem olvas be több játékost, mint a tömb maximális mérete. Visszatérési értéke 0, ha sikeres, különben 1.

- `jatekosok_kiir`: A kapott nevű fájlba írja az adatokat a játékosokat tartalmazó tömbből. Meg kell adni a tömb méretét és a soron következő játékos indexét.

jatek.h:

- `kormenu`: A kör elején lévő menüt kezeli. Argumentuma egy cím szerint átadott játék. Visszatérési értéke a felhasználó által választott opció. (Típusa `enum parancs`).
- `jatek_print_info`: A cím szerint átadott játékban lévő betűkészletről és a játékosokról ír ki adatokat a konzolra.
- `jatek_kirakas_beker`: Cím szerint kap egy játékot. Bekéri egy kirakás adatait a felhasználótól, majd visszaadja.
- `jatek_kirakas_vegrehajt`: Ellenőrzi, hogy szabályos-e a kirakás. Ha igen, akkor lerakja a pályára, elveszi a betűket a játékosról, és a következő játékos jön. Ha nem, akkor kiírja a hibaüzenetet.
- `jatek_beolvas`: A mappa nevét és az üres játékra mutató pointert kapja meg, majd a mappából beolvassa a játék adatait. Visszatérési értéke 0, ha sikeres, különben 1. A játékot a hívónak kell törölnie.
- `jatek_torol`: A játék összes adatát törli, tehát a benne lévő egyéb struktúrákat is.
- `jatek_kiir`: A kapott mappába írja a cím szerint átadott játék adatait.
- `jatek_kirajzol`: Kirajzolja a pályát, és kiírja, hogy ki jön és neki milyen betűi vannak. A pályát koordinátákkal együtt jeleníti meg. Ahol nincs még betű lerakva, azaz az akciómező még érvényes, ott azt jelzi is. Kétszeres szóérték: „*2”, háromszoros szóérték: „*3”, kétszeres betűérték: „+2”, háromszoros betűérték: „+3”. Mivel a hagyományos játékban csak ilyen akciómezők fordulnak elő, a függvény csak ezeket képes megjeleníteni.. (A fájlformátum és az adatstruktúra lehetővé teszi az egyéb értékű akciómezőket is).
- `jatek_vege`: Cím szerint vesz át egy játékot, és eldönti, hogy befejeződött-e. (Akkor fejeződött be, ha nincs több betű a betűkészletben). Logikai értéket ad vissza.
- `jatek_befejez`: Cím szerint vesz át egy játékot. Kiírja a győztest.
- `jatek_mentes_parbeszed`: Cím szerint vesz át egy játékot. Bekéri a felhasználótól a mentési mappát és elmenti a játékot.
- `jatekos_feltolt_betuk`: Cím szerint vesz át egy játékot. A soron következő játékos betűit feltölti random betűkkel a `random_betu_kivesz` függvény hívásával. Tehát a betűk el is vesznek a betűkészletből.
- `palya_kifer`: Egy pályát és egy kirakást vesz át. Eldönti, hogy a kirakás betűi kiférnének-e a pályára. Azt is figyelembe veszi, hogy vannak-e olyan betűk lerakva, amik miatt már nem férne oda a szó. Logikai értéket ad vissza.
- `van_uj_betu`: A pályát és a kirakást kapja argumentumként. Ellenőrzi, hogy a a kirakásban kerülne-e le új betű a pályára. Logikai értéket ad vissza.
- `jo_meroleges_szavak`: Átveszi a pályát, a szótárát és a kirakást. Negyedik argumentuma egy sztring, ahova az esetleges hibás szó fog kerülni. Eldönti, hogy a kirakás során keletkező merőleges szavak mind benne vannak-e a betűkészletben. Logikai értéket ad vissza.
- `van_szomszed`: A pályát és a kirakást vesz át. Eldönti, hogy van-e legalább egy szomszédos betű a kirakáshoz, ugyanis az csak akkor szabályos. (Üres területre csak az első körben kerülhet szó, az is csak a pálya közepét érintve). Logikai értéket ad vissza.
- `jatekos_vannak_betuk`: Átveszi a játékot cím szerint és a kirakást. Eldönti, hogy a soron következő játékosnak vannak-e a kirakáshoz szükséges betűi.
- `szabalyos`: A fenti függvények felhasználásával eldönti, hogy egy kirakás szabályos-e. Argumentumként a pályát kapja cím szerint, a kirakást és egy sztringet, ahova hiba esetén egy hibaüzenetet ír. Logikai értéket ad vissza.
- `jatekos_levon_betuk`: Átveszi cím szerint a játékot és egy kirakást. A játékos tartójából kiveszi a kirakáshoz szükséges betűket.

A programozás alapjai 1 – Házi feladat

- `jatek_kovetkezo_jatekos`: Beállítja a cím szerint átvett játékban, hogy a következő játékos következzen.
- `pontszam`: Átveszi a pályát, a betűkészletet és a kirakást. Kiszámolja, hogy a keletkező merőleges szavak és az akciómezők figyelembevételével a kirakás hány pontot ér. Ezt a pontszámot adja vissza.
- `palya_kirak`: A pályát és a kirakást kapja argumentumként. Lerakja a betűket a pályára.

Gegő Levente 2021. 11. 12.