

Imperatív Programozás Labor ZH – 2023.11.10.

Elvárások a programmal szemben

- A megoldáshoz semmilyen segédeszköz nem használható, kivéve a C referenciát.
- A program végleges verziójának működő képesnek kell lennie. Forduljon és fusson!

A nem forduló kód 0 pontot ér!

- Ne használj globális változókat! Csak a makrók megengedettek!
- Logikusan tagold a megoldást! A megoldás részeit külön függvényekben valósítsd meg!
- Kerüld a nem definiált viselkedést okozó utasításokat!

A végleges programban ezek megléte pontlevonással jár!

Egyszerű Snake játék

A feladatod a Snake játék leegyszerűsített verziójának elkészítése lesz. A játékos lépésenként irányíthatja a kígyót a játék mezőn, hogy a játék elején véletlenszerű pozíciókban megjelent almákat összegyűjtse, miközben igyekszik nem lefutni a pályáról illetve nem összeütközni önmagával.



A feladat összesen **50** pontot ér.

Legalább **10** pontot kell gyűjteni a tárgy sikeres teljesítéséhez.

A megoldásra **180 perc** áll rendelkezésre.

A végén csak az egy darab forráskódot (`<neptun_kód>.c` állomány) kell feltölteni. (Az utoljára feltöltött megoldást pontozzuk.)

A kötelezően megvalósítandó alprogramok:

Játéktér inicializálása – `init_field()` (7 pont)

A játéktérre reprezentálja egy **10 x 20**-as karaktereket tároló kétdimenziós tömb. (A dimenziókat tároljuk makrókban.) Helyezzünk el véletlenszerűen **10** almát a játéktéren. Figyeljünk arra, hogy minden alma felkerüljön, ne tudjon a program ugyanarra a mezőre több almát generálni! Az üres játék mezőket ' '(SPACE), az almákat 'a' karakter jelölje.

A függvény megkapja a játéktérre tároló mátrixot és az almák számát paraméterben.

Kígyó inicializálása – `init_snake()` (5 pont)

A kígyót reprezentálja egy 9x2-es egészeket tároló tömb, amely a kígyó testének koordinátáit (indexeit) jelenti a játék mezőn. A (0,0) és (0,1) jelentse a kígyó fejének pozícióját. A kígyó legyen 9 hosszú és a játék során nem változtatja a méretét. A kígyó testszegmenseinek adjunk érvényes koordinátákat. (Tipp: Tetszőlegesen elhelyezhetjük a kígyót a játéktéren, csak ne lógjon le. Azzal nem kell foglalkozni, hogy almára tettük-e.)
A függvény megkapja a kígyót tároló mátrixot paraméterben.

Játék aktuális állapotának kirajzolása – `print_game()` (11 pont)

A játék vizualizációjáért felelős alprogram. A kirajzoláshoz másoljuk le a játéktér mátrixot egy munka mátrixra. Erre a munka mátrixra fogjuk rárakni a kígyót minden kirajzolás előtt. (Egyszerűbb, mint a kígyó régi pozícióját törölni.) A kígyó fejét jelöljük '8'-cal a teste által elfoglalt mezőket jelöljük '0'-val.

Rajzoljuk ki a játéktérre valamilyen kerettel (pl.: '#') a sarkokat is beleértve, amin belül látszódnak a begyűjtendő almák és a kígyó.

A függvény megkapja a játéktérre és a kígyót tároló mátrixot paraméterben.

Kígyó frissítése – update_snake() (17 pont)

Az alprogram frissíti a kígyó testének koordinátáit. Siker esetén visszaadja, hogy történt-e alma begyűjtés(1) vagy nem(0), sikertelenség esetén jelzi, hogy a pálya szélével(-1) vagy a kígyó önmagával(-2) ütközött-e. A kapott karakter függvényében meghatározzuk, hogy merre mozdul el a kígyó feje: 'a' – balra, 's' – lefelé, 'w' – felfelé, 'd' – jobbra. Az új fejpozíciónál ellenőrizzük, hogy nem futottunk-e le valamelyik irányban a pályáról. Ha nem, akkor ellenőrizzük, hogy alma mezőre léptünk-e. Alma mező esetén, az eredeti játéktér mátrixról töröljük az almát (' ' mező lesz). A kígyó koordinátáit frissítsük értelemszerűen. (Tipp: másoljuk eggyel nagyobb indexű pozícióba a test szegmensek koordinátáit (hátról előre haladva), a legeleső azaz a (0,0) és (0,1) indexű helyekre kerül az új sor és oszlop indexe a fejnek.) Ellenőrizzük, hogy a kígyó nem ütközött-e önmagával. (Tipp: nincs-e a fej és egy másik szegmens ugyanabban a pozícióban. Következmény: ha a kígyó feje a „haladási iránnyal” ellentétesen, azaz a teste felé mozdulna el az ütközést jelentene.)

A függvény megkapja a játéktér, a kígyót tároló mátrixot és az elmozdulás irányát tároló karaktert paraméterben.

Főprogram – main() (10 pont)

A program összefogása és kommunikáció a felhasználóval. Köszöntsük a játékost és röviden ismertessük a játékot. (Saját szavakkal, pár mondatban elegendő.) Deklaráljuk a játékhoz szükséges változókat, tárolókat. Hívjuk meg a játék inicializálását végző alprogramokat, majd írassuk ki a kezdő állapotot. A játékos az 'aswd' karakterekkel vezérelheti a kígyót. Egy vagy több karakter beütése után az <ENTER> (newline karakter) beütésére lép a kígyó egyet vagy többet. Magát a newline karaktert ignoráljuk. A játékból az EOF megadásával léphetünk ki. Az érvénytelen karaktereket vesszük el, ne akadályozzák a játék futását. Érvényes karakter esetén frissítsük a kígyó pozíció értékeit. Sikertelen lépés (ütközés) esetén a játék leáll. Sikeres lépés esetén rajzoljuk ki az új állapotot. Ha történt alma begyűjtés, akkor frissítsük a megmaradt almák számát. Ha minden almát begyűjtöttünk, akkor gratuláljunk a játékosnak (üzenet a standard kimenetre) és érjen véget a játék.