

Programozás alapjai 1

Czap Balázs

Nagy házi feladat

Tesztelési Dokumentáció

Indítás:

Például windows intézőből indítva hibátlanul működik, vagy parancssorból bárhány paraméterrel elindítva hibátlanul működik (azaz elindul a program. mivel nem parancssori program, ezért ennél többre nincs is szükség).

A program elindul, és mivel a forráskódban 100x100-as méret van, egy ekkora élethez kirajzol (átírtam a forráskódban ezt 10x10-re, és így tényleg megszámlálhatóan egy 10x10-es élethez rajzolt ki). Teljesen fehér az egész, tehát tényleg kinullázottan indul.

Teljesen helyesnek mondható a program indulása.

Futás:

Menü:

A **Reset**, **Run**, és **Stop** gombok hibátlanul működnek.

Az **1-step** gomb a szimuláció futtatása közben aktív marad, de ez nem akadályozza a program helyes működését. A Simul.speed gombok mellett látható a szimuláció léptetésének sebességére vonatkozó érték, ha ez elég alacsony (4-5 körül) akkor tökéletesen funkcionál a gomb, viszont ha magasabb, akkor nagyon rövid kattintások esetén a gomb nem csinál semmit. Ha hosszabban lenyomva tartjuk, helyesen működik.

A **Save** gomb a helyes fájl névvel ment a helyes helyre, és a Load gomb alatt (megfelelő mennyiségű lapozás után) egyből meg is jelenik.

A **Load** gomb (azaz a bemeneti fájlok tesztelése):

- Ha a saves könyvtárban nincsen semmi, a Load gomb nem csinál semmit (helyes működés)
- Ha a saves könyvtárban csak tökéletes formátumú fájlok vannak, akkor azokat helyesen betölti.
- Ha a saves könyvtárban ezen kívül van más .txt kiterjesztésű fájl is (a nem .txt-ket figyelmen kívül hagyja):
 - pl. asd.txt, nem a szokásos „szám, szám\n” adatpárral kezdődik.
Mindenképpen egy furcsa, két vonalból álló rajzot mutat. Ebből az állapotból úgy lehet kilépni, hogy megnöveljük az élethez méretét valamilyen irányba, vagy ha van helyes formátumú fájl, akkor annak a betöltésével.
 - pl. asd.txt, és a szokásos „szám, szám\n” adatpárral kezdődik.
Ilyenkor az első számot jól értelmezi, a másodikat automatikusan 255-nek. Létrehoz egy üres élethez ezekkel az értékekkel.

- pl. Helyes formátum, de a fájlba kevesebb sor van írva mint ahányat az első szám indokolna: a számpár által megadott méretű élettér létrejön, és helyesen beleíródik a fájlban található adat. A többi, hiányzó sor 0-val töltődik fel.
- pl. Helyes formátum, de a fájlba több sor van írva, mint ahányat az első szám indokolna: azok a sorok figyelmen kívül vannak hagyva
- pl. Helyes formátum, de a sorok rövidebbek mint ahányat a második szám indokolna: helyesen betöltődik, a sorok rövidege figyelmen kívül van hagyva, a maradék 0-val töltődik fel.
- pl. Helyes formátum, de a sorok hosszabbak, mint ahányat a második szám indokolna: a második szám által megadott hosszúságig tölt, a többit figyelmen kívül hagyja.

A **Cellsize** nyilak helyesen növelik illetve csökkentik a cellák méreteit. Ha már nem fér ki az élettér, automatikusan nem engedi tovább növelni.

A **Simul. speed** nyilak egyetlen hibája, hogy kissé nehéz precíz sebességet beállítani, viszont helyesen működnek.

A **Ver. size** és **Hor. size** nyilak helyesen működnek.

Szimuláció:

A rajzolás és törlés az élettéren helyesen működik. Ha az élettéren kívül, de még az annak fenntartott helyen belül klikkelek, akkor az élettér ahhoz legközelebbi pontjára rajzol. Ez nem feltétlenül hibátlan dolog, de nem akadályozza a program működését igazából.

Maga a szimuláció helyesen működik. Az egyetlen hibás működést az adhatja, hogyha az élettér szélső sávjában (3 cella alapértelmezetten ami nincs kirajzolva), van valami, de feltételezzük hogy nincs (pl ezt úgy lehet elérni, hogy telerajzoljuk az életteret, futtatjuk pár körig, majd az egérrel a látható részeket kitöröljük, így általában marad valami a szélén), akkor sem feltétlenül hibás, de nem várt eredményeket produkálhat a szimuláció.