

Život

Vášim úkolem je implementovat simulátor **Conwayovy hry života**. Váš program dostane jako argument příkazové řádky cestu k souboru, ve kterém jsou popsány rozměry mřížky a počáteční stav všech buněk. Program poté simuluje vývoj všech buněk v mřížce podle předem zadaných pravidel a výslednou mřížku v pravidelných intervalech vykresluje do terminálu.

Pravidla

Hra život je hra nula hráčů, která se odehrává v obdélníkové mřížce o šířce w a výšce h . Políčka mřížky se nazývají *buněk* a mohou v každém okamžiku být *živé*, nebo *mrtvé*.

Jaké buňky jsou na začátku hry živé a jaké mrtvé, zadává uživatel na vstupu. Poté se v každém kole vypočítá nový stav buněk podle následujících pravidel:

- Každá živá buňka, která má dva nebo tři živé sousedy, zůstane živá. Pokud má živá buňka jiný počet živých sousedů, zemře (buď samotou, nebo přelidněním).
- Každá mrtvá buňka, která má právě tři živé sousedy, obživne (reprodukcí). Pokud má mrtvá buňka jiný počet živých sousedů, zůstává mrtvá.

Při výpočtu se uvažují i sousedi diagonálním směrem. Tedy každá buňka má celkem 8 sousedů.

Dejte si pozor na to, že nový stav všech buněk se počítá *zároveň*. Tedy pokud počítáte pro nový stav počet živých sousedů buňky, je potřeba brát do úvahy jejich stav v předchozím kroku, nezávisle na tom, zda v novém stavu někteří ze sousedů ožili a někteří zemřeli.

Formát vstupu

Rozměry mřížky a počáteční stav všech buněk jsou popsány ve vstupním souboru, jehož tvar je následující. První dva řádky souboru obsahují kladná čísla, která reprezentují šířku w a výšku h mřížky (v tomto pořadí). Následuje graficky reprezentovaná mřížka, kde živé buňky jsou reprezentovány znakem '#' a mrtvé znakem ' ' (mezera). Soubor může obsahovat i méně řádků mřížky než h a řádky mohou být kratší než w znaků. V takovém případě se neuvedené buňky považují za *mrtvé*.

Například následující soubor reprezentuje mřížku širokou 40 políček, vysokou 50 políček, ve které na začátku žije právě 5 zakreslených buněk. Tento soubor najdete i ve studijních materiálech jako `examples/glider.txt`.

```
40
50
#
#
###
```

Příklady

Kluzák

Při zavolání `./life examples/glider.txt` program nejprve smaže terminál a vykreslí do něj počáteční stav.

```
1 #
2 #
3 ###
```

Program může nebo nemusí vykreslit zbývajících 47 prázdných řádků; to necháváme na vás. V dalším kroce¹ smaže terminál a vykreslí následující stav (s prvním prázdným řádkem)

```
1
2 # #
3 ##
4 #
```

¹Časový interval mezi dvěma následujícími kroky necháváme na vás. Rozumná perioda může být například 200 ms.

V dalším kroce smaže terminál a vykreslí následující stav (s prvním prázdným řádkem)

```
1
2 #
3 # #
4 ##
```

V dalších krocích kluzák postupně putuje doprava dolů.

Další počáteční stavy

Další zajímavé počáteční stavy najdete na [anglické Wikipedii](#). Pro jednodušší testování některé z nich najdete ve studijních materiálech jako:

- Nehybné `examples/block.txt` a `examples/loaf.txt`.
- Oscilátory `examples/blinker.txt` a `examples/pulsar.txt`.
- Vesmírné lodě `examples/glider.txt` a `examples/lwss.txt`.

Rady na závěr

Přesný formát výstupu je na vás. Pokud se rozhodnete udělat výstup hezčí, rozhodně vám v tom nebudeme bránit. Pokud se cítíte odvážně, můžete místo do terminálu vykreslovat buňky graficky pomocí OpenGL. K tomu by se vám mohla hodit například knihovna [Gloss](#), a zejména pak její funkce `simulate`.²

Odevzdáváte program, který jde přeložit na spustitelný soubor. Tedy musí obsahovat funkci `main :: IO ()`. Samozřejmě nejen můžete, ale také byste měli, definovat vlastní pomocné funkce.

Na reprezentaci aktuálních živých buněk by se vám mohla hodit datová struktura `Set` z modulu `Data.Set`.

Pro vykreslování do terminálu se vám můžou hodit funkce z modulu `System.Console.ANSI`³. Zejména se podívejte na funkce `setCursorPosition`, `hideCursor` a `clearScreen`. Také se vám může hodit funkce `threadDelay` z modulu `Control.Concurrent`.

Nebojte se často kontrolovat dokumentaci používaných modulů na [Hackage](#) či využívat vyhledávač [Hoogle](#). V případě problémů a dotazů se nebojte obrátit se na nás například na diskuzním fóru.

²Je potřeba nainstalovat balíček `gloss` pomocí `cabal` a knihovnu `freeglut` pomocí správce balíčků vašeho operačního systému.

³Je potřeba nainstalovat balíček `ansi-terminal` pomocí `cabal`.