

Sokoban

A programozás alapjai 3.

Házi feladat specifikáció

Radnai Bálint

Áttekintés

A házi feladatom a jól ismert klasszikus Sokoban játék elkészítése Java nyelven.

A játék szabályai

A játék során a felhasználó különféle pályákat oldhat meg. Minden pálya egy speciális négyzetrácsos raktárépületben játszódik, amelyre a felhasználó felülnézetben, stilizáltan lát rá. A raktár négyzet alakú mezőkből/cellákból épül fel. A raktár bizonyos mezőin dobozok vannak, illetve egy raktáros (japánul *sokoban*) is tartózkodik az épületben. A játékos és minden egyes doboz pontosan egy-egy, egymástól különböző mezőt foglal el. A raktáros feladata a dobozok meghatározott célmezőkre juttatása.

A raktarat kívülről mozdíthatatlan és áthatolhatatlan falak határolják, de a raktáron belül is lehetnek ugyanilyen falak. A falak által határolt belső területet padlómezőkből áll. A padlók lehetnek célmezők vagy nem célmezők. Lehetnek üresek, de állhat rajtuk doboz vagy játékos is.

A raktáros egy lépésben egy oldalszomszédos mezőre léphet a négy irány (fel, le, balra, vagy jobbra) valamelyikébe, a következő egyszerű szabály szerint:

- Ha az a mező, amelyre lépne, üres, akkor a lépés gond nélkül megtehető.
- Ha az adott irányban levő szomszédos mezőn doboz található, de a dobozon túl üres cella van, akkor a lépés során nemcsak a játékos lép, hanem a dobozt is egy egységgel eltolja az adott irányba.
- Ha viszont a dobozon túl fal vagy másik doboz van, akkor a játékos abba az irányba nem léphet.

A dobozok teljesen egyformák, így nincs megkötés arra, hogy melyiket melyik célmezőre toljuk. A felhasználó akkor teljesített egy pályát, ha megengedett lépések egy véges sorozatával mindegyik dobozt célmezőre sikerült tolnia.

A játék pályáinak megoldása akkor az igazi kihívás, ha minimalizálni próbáljuk a megtett lépések, vagy a megtett doboztolások számát, esetleg az ehhez szükséges időt is.

A programmal kapcsolatos további követelmények

A programnak képesnek kell lennie 0.16 verziójú **.sok** formátumú fájlokból olvasni a pályák adatait. (Ez a formátum pályák listái, azaz pályacsomagok leírásához használatos.)

Az eredményeket XML formátumban kell tárolni, ahol első szinten a pályák, második szinten a pályán elért eredménybejegyzések találhatók. Az egyes eredményeknél szerepeljen mezőként a helyezés, a játékos által beírt név, a lépések száma, a tolások száma, az időeredmény és a játék dátuma, ideje.

Use case-ek

Az alkalmazás fő funkciói

A program a **soko.jar** fájl elindításával nyitható meg. Fontos, hogy ezzel a fájlal egy mappában legyen egy **levels** nevű mappa, amely a pályacsomagokat-, és egy **results** mappa, amelyik az eredményfájlokat tartalmazza. A program indításakor egy tetszőleges alapértelmezett pályacsomag legelső pályája töltődik be, és jelenik meg a képernyőn.

Ha másik pályacsomagra szeretnénk váltani, azt a **Pályacsomag betöltése** menüvel tehetjük meg. Az előugró ablakban tallózhatunk a megfelelő **.sok** kiterjesztésű fájlok között. Megnyitás után a megnyitott fájl első pályája töltődik be a képernyőre. A **<pályacsomag>.sok** pályacsomagban levő pályákhoz tartozó eredmények a **<pályacsomag>.xml** fájlban találhatók meg, induláskor ennek a tartalma is betöltődik.

Ezután navigálhatunk a pályacsomag pályái között, és játszhatunk az aktuális pályán.

A pályán elért eredményünket elmenthetjük, és a pálya eredménylistájában megtekinthetjük, az „Eredmények” gombra kattintva. Az eredménylista új ablakban jelenik meg. Az eredmények rendezhetők

- lépésszám szerint növekvően, azon belül idő szerint csökkenően, vagy pedig
- tolások száma szerint növekvően, és idő szerint csökkenően.

A program a pálya legjobb 15 eredményét jeleníti meg. Az eredménylista ablaka a jobb felső piros „X”-szel zárható be.

A játékból ki lehet lépni az Exit gombbal, vagy itt is az ablak jobb felső piros „X” gombjára kattintva. Kilépéskor frissül az eredményfájl.

Játék egy adott pályán

Amikor egy adott pályára navigálunk, akkor annak a kezdőpozíciója rajzolódik ki a képernyőre. A raktáros a billentyűzetről a Joystick-gombokkal vagy a WASD gombokkal, egérrel pedig a játékalap megfelelő fel/le/balra/jobbra gombjaira kattintva irányítható. A pálya rajza automatikusan frissül a szabályosan végrehajtható lépések során.

Lehetőség van sorban, egyesével visszavonni a legutóbb megtett lépéseket a „Vissza” gombra kattintva, vagy az U billentyűt leütve (Undo). A pályát újra is kezdhethetjük az „Újra” gombbal vagy az R billentyűvel (Restart).

Kezdetben a lépések és a tolások számlálója, valamint az idő is nullázva van. Minden lépésben eggyel nő a megtett lépések számlálója, és ha a játékos dobozt is tolt a lépésével, akkor a doboztolások számát is növeljük. Az U billentyű hatására ezeknek a száma a megfelelő módon csökken. Az R billentyű lenyomásakor mindkét számláló nullázódik.

Kezdetben a mért idő is nulla. Az időmérés az első lépés megtételével indul. Ha a játékos

- sikeresen megoldotta a pályát,
- vagy újrakezdi azt,
- vagy ha másik pályára navigál,

akkor a futó időmérés leáll. Az utóbbi két esetben az időadat is nullázódik.

Sikeres megoldás esetén az alkalmazás egy felugró ablakban gratulál a játékosnak, aki megadhatja a nevét. Ha az „Eredmény mentése” gombra kattint, akkor az eredményét el kell tárolni az eredménylistában. Ha a „Vissza a játékhoz” gombra kattint, vagy bezárja az ablakot, akkor ne történjen mentés.

A játékos, ha szeretné, megtekintheti az aktuális pálya mintamegoldását, ha van. Ezt a „Megoldás” gombbal kérheti. Ilyenkor újramegoldódik a pálya, és deaktiválódnak a játékot irányító gombok. A program lejátssza az animált megoldást a felhasználónak. Ezt a lejátszást meg lehet szakítani másik pályára navigálva.

Tervezett megvalósítás

A házi feladat követelményeit az alábbi módon valósítanám meg.

Grafika

A grafikát Java Swing-gel tervezem kivitelezni, a tanult widget-ek, elsősorban gombok használatával. A játék képernyőjének tetején helyezkedik el a menürendszer, ahonnan a pályacsomagok megnyithatók, és az aktuális pályacsomag bezárható. A képernyő közepén található a játékpálya, ahol a játékos lépked. A mezőket - fajtájuktól függően - előre megadott képkockák lenyomataként jelenítem meg minden lépésben. Alul helyezkedik el az időmérő, és a lépésszámlálók.

Gyűjtemény

A pályákat kétdimenziós generikus tömbmegvalósításban tárolnám, például ArrayList-ekben, az eredménylistát pedig fésűs listában (pl. LinkedList kollekció).

Fájlkezelés

A függelékben megtalálható a használt 0.16 verziójú **.sok** formátum¹ specifikációja. A fájlt egyszerű parse-olással fel lehet dolgozni, kiolvassa belőle az adattagokat. Itt nincs szükség XML használatára.

¹ A formátum leírásának forrása: http://www.sokobano.de/wiki/index.php?title=Sok_format (Legutóbbi hozzáférés: 2020.11.01.)

Függelék

A .sok formátum és a fejléce

A pályatérkép karakteres formátumát kétféleképpen lehet megadni. Mindkettőben mindenfajta mezőt különböző karakterrel kódolunk. A pályák mintamegoldásait a LURD jelölésrendszer alapján kell megadni.

A 0.16 verziójú formátum fejléce az alábbi:

```

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::          Sokoban (c) by Falcon Co., Ltd., Japan          ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::          File Format 0.16                                ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::                                                         ::
:: File Notes                                             Optional  ::
:: Puzzle 1                                             Required  ::
::   Title                                           Optional*  ::
::   Board                                           See legend  ::
::   Puzzle Notes                                     Optional  ::
::   Saved Game or Solution 1                       Optional  ::
::     Title                                           Optional*  ::
::     Moves                                           See legend  ::
::     Notes                                           Optional  ::
::   Saved Game or Solution 2                       Optional  ::
::   ... (more saved games and solutions)             ::
:: Puzzle 2                                             Optional  ::
:: ... (more puzzles)                                   ::
::                                                         ::
:: Remarks:                                             ::
::                                                         ::
:: File Notes                                             ::
::   File notes consist of unstructured text and         ::
::   key/value properties, such as "Author: Name". Lines ::
::   beginning with ":@" are comments meant to be read  ::
::   only by a person examining the file in a text      ::
::   editor, and should not be displayed by the Sokoban ::
::   program.                                             ::
::                                                         ::
::   The optional but recommended property              ::
::   "Collection: Name" assigns a name to the puzzle    ::
::   collection. When a collection is copied from the   ::
::   internet, for example, and pasted into a Sokoban  ::
::   program, this information allows the collection to  ::
::   be saved with the proper name.                     ::
::                                                         ::
:: Titles                                             ::
::   A title line is the last non-blank text line before ::
::   a puzzle, a saved game, or a solution, provided the ::
::   line is preceded by a blank line or it is the only ::
::   text line at this position in the file.           ::
::                                                         ::
::   Title lines are optional unless a single or a last  ::
::   text line from a preceding puzzle, saved game,     ::
::   solution, or file header can be mistaken for a title ::

```

```

:: line. ::
::
:: Puzzle Notes ::
:: Two special key/value pairs are supported in puzzle ::
:: notes: "Title" and "Author", hence, titles can ::
:: either come from a title line or from a key/value ::
:: pair. ::
::
::: Board :::
:: Legend.....: .....Legend ::
:::
:: Wall.....: # # .....Wall ::
:: Pusher.....: p @ .....Pusher ::
:: Pusher on goal square...: P + ...Pusher on goal square ::
:: Box.....: b $ .....Box ::
:: Box on goal square.....: B * .....Box on goal square ::
:: Goal square.....: . . .....Goal square ::
:: Floor.....: .....Floor ::
:: Floor.....: - _ .....Floor ::
::
:: Remarks: ::
::
:: The first and the last non-empty square in each row ::
:: must be a wall or a box on a goal. A board cannot ::
:: have empty rows. ::
::
:: Boards may be run-length encoded (RLE), e.g., ::
:: "###---p.#" may be encoded as "3#4-p.#", and ::
:: "#-#-#-#-#-#-#" may be encoded as "2(3(#-)#)". ::
:: A row cannot be split over multiple physical lines. ::
::
:: Rows may be combined on a single physical line by ::
:: using "|" as a row separator, e.g., ::
:: "--3#|3#-#|#pb.#|5#". A "|" at the end of a physical ::
:: line is optional and may be omitted. ::
::
::: Moves :::
:: Legend.....: .....Legend ::
:::
:: Move pusher up.....: u U .....Push/pull box up ::
:: Move pusher down.....: d D .....Push/pull box down ::
:: Move pusher left.....: l L .....Push/pull box left ::
:: Move pusher right.....: r R .....Push/pull box right ::
:: Begin jump.....: [ ] .....End jump ::
:: Begin pusher change.....: { } .....End pusher change ::
:: Current position.....: * * .....Current position ::
::
:: Remarks: ::
::
:: Moves may be run-length encoded, e.g., "3r4U" means ::
:: "rrrUUUU", and "2(3(dr)R)" means "drdrdrRdrdrdrR". ::
:: Each line must, however, have at least one proper ::
:: non-digit character. Spaces between moves are ::
:: allowed. ::
::
:: Jumps and pulls: Only in reverse mode saved games ::

```

```
:: and solutions. ::
:: ::
:: Reverse mode saved games and solutions must begin ::
:: with a jump, even if it is empty. An example: ::
:: "[]U[rr]d". ::
:: ::
:: Pusher changes: Only in puzzles with multiple ::
:: pushers, e.g., Multiban. Moves inside the braces ::
:: depict the relative movement to get from the ::
:: currently active pusher to the next active pusher. ::
:: At game start, a "{...}" sequence activates the ::
:: pusher relative to the top-left pusher. An example: ::
:: "{rddd}Urr{uul}uLU". If the top-left pusher is the ::
:: first active pusher, then the empty "{}" can be ::
:: omitted. ::
:: ::
:: The current position is optional and defaults to the ::
:: position after the last move. ::
:: ::
:: ::
```