Rush Hour

Concurrency Opgave 3, Fall 2013[1]

Achtergrond Rush Hour is een schuifpuzzel waarbij je door middel van het schuiven van andere auto's op het terrein je eigen auto moet proberen vrij te spelen.

Opdracht Voor deze opdracht moet je een parallelle Rush Hour solver maken. Je programma moet niet alleen de standaard 6x6 configuratie aan kunnen, maar ook rechthoeken van andere formaten. Bij het zoeken van een oplossing moet je bijhouden welke posities in de puzzel je al hebt bezocht, om herhaling van zetten te voorkomen. Er zullen voor dit programma twee uitvoermodi zijn:

- Telmodus: Geef het aantal stappen dat minimaal nodig is om de puzzel op te lossen. Indien de puzzel niet oplosbaar is wordt -1 gegeven.
- Oplosmodus: Geef als antwoord een mogelijke kortste oplossing door middel van een stap voor stap beschrijving. Elke stap wordt in de uitvoer gescheiden door een komma en bestaat uit 3 characters: (1) het character van de auto, (2) de richting van de stap: u, d, l of r voor respectievelijk up, down, left en right, (3) het aantal vakjes dat de auto verschuift in die richting. Als er geen oplossing voor de puzzel is wordt "Geen oplossing gevonden" als uitvoer gegeven.

Een auto meer dan 1 hokje verplaatsen wordt ook gerekend als 1 stap. Let er op dat je tijdens de Telmodus geen tijd en geheugen besteedt aan het onnodig bijhouden van de bijbehorende oplossing.

Het doel van het spel is om je eigen auto te bevrijden. Voor deze opdracht zal je eigen auto altijd aangeduid worden met de letter x. Omdat we met variabele borden werken wordt bij de invoer de targetspace gedefiniëerd (zie Invoer). Dit zijn de coordinaten van het vakje, geteld vanuit een 0-based grid, waar de linker bovenhoek van onze auto zich moet bevinden om de puzzel als opgelost te beschouwen. Het targetvak ligt altijd binnen het bord.

De Invoer De invoer begint met 4 regels met daarop:

- De uitvoermodus. Voor Telmodus staat hier een 0, voor Oplosmodus een 1.
- Het aantal regels dat het bord hoog is (y-richting). $0 < bordhoogte < 120$
- De y-coordinaat van het targetvakje (zie target).
- De x-coordinaat van het targetvakje (zie target).

[1] Versie: 21 oktober 2013.

Daarna volgt het bord regel voor regel. Regels hebben een gelijke lengte, en elke regel is maximaal 120 characters. Een '.' geeft een leeg vakje aan, een 'x' geeft je eigen auto aan, en elk ander character is een andere auto. Auto's zijn minstens 2 vakjes lang, en zijn altijd 1 dimensionaal. Ze staan altijd horizontaal of verticaal. Elk character geeft een unieke auto aan. Er zit geen limiet aan hoe veel auto's er op het bord staan, anders dan de limiet van wat er fysiek op het bord past.

- Het character van een auto heeft niets met de auto te maken, behalve auto x.
- Ook je eigen auto kan groter zijn dan 2 vakjes.
- Je eigen auto kan ook verticaal staan.
- Hoe meer extra tekst je print in je programma, hoe langzamer hij is in de wedstrijd.

Punten en Extra Punten Bij het maken van de opdracht mag je gebruik maken van System.Collections.Concurrent. Het implementeren van een eigen datastructuur levert meer punten op. Kijk bijvoorbeeld naar een Trie, of naar wachtvrije implementaties. Ook zijn er extra punten te halen door de opdracht uit te breiden, bijvoorbeeld door een implementatie te maken die via de GPU de oplossing berekent. Uitbreidingen dienen wel altijd met Concurrency te maken te hebben om in aanmerking te komen voor extra punten, en moeten vermeld worden in het verslag (zie Kort Verslag). Het maximale cijfer voor deze opdracht is een 10. Er valt een voldoende te behalen zonder uitbreiding of zelf geïmplementeerde datastructuur.

Kort Verslag Bovenaan je opdracht moet je kort vertellen welke datastructuren je hebt gebruikt, waar de bottlenecks zitten in je programma, en hoe je er voor hebt gezorgd dat je programma zo snel mogelijk is. Ook bottlenecks die je hebt vermeden moeten hier genoemd worden. De richtlijn voor dit verslag is tussen de 5 en de 10 regels.

Voorbeelden Hier volgen enkele voorbeelden van invoer en bijbehorende uitvoer. Je kunt ze vinden op de praktikumpagina, waarbij voorbeeld VVV bestaat uit bestanden VVV.in en VVV.uit.
Voorbeeld `st`:

```
1                                   bd1, cl1, pu3, fr1, ku1, hr4, dd1, kd1,
6                                   fl1, pd1, cr1, el1, od3, bu1, xr3, ou3,
2                                   er1, du3, el1, od3, ar1, du1, xl3, ou1,
4                                   bd1, cl1, pu1, fr1, ku1, hl4, kd1, fl1,
aaobcc                              pd3, cr1, bu1, od1, xr4
..ob..
xxo...
deeffp
d..k.p
hh.k.p
```

Voorbeeld `drie`: Deze kan in drie stappen.

```
1                                   ad3, bd2, xr4
6
2
4
......
..a...
xxa.b.
....b.
......
......
```

Voorbeeld `vol`: Oei die staat behoorlijk vol.

```
1                                   xl1, bd2, ar1, eu1, xl1, hu1, jl2, dd2,
6                                   cd2, fr2, ar2, hu2, xr1, gr1, ed4, xl1,
2                                   gl1, hd2, al2, du2, fl3, bu1, cu2, hu1,
4                                   gr4, jr2, bd2, hd2, fr2, xr2, eu4, iu3,
aaabcd                              kl2, bd1, hd1, gl4, cd1, ar1, iu1, xl2,
effbcd                              hu2, bu2, jl4, ll2, cd2, bd1, hd1, dd3,
e.xxcd                              xr4
ggh...
.ih.jj
.ikkll
```

Voorbeeld `fifty`: Fifty shades of sweat.

```
1                                   au1, xl1, cd1, dl2, iu3, ur1, tr2, rd1,
6                                   yr1, ed3, yl2, id1, dr2, cu1, xr3, cd1,
2                                   dl1, iu1, yr2, br2, gu4, bl2, yl1, id1,
4                                   dr1, cu1, xl3, cd2, dl1, iu1, yr1, eu3,
..cddd                              yl1, id1, dr1, cu2, ru2, tl4, ul4, rd2,
a.ce..                              cd1, dl1, iu1, yr1, ed3, yl2, id3, dr1,
axxe..                              cu1, xr4
bbryyi
.gruui
.gtt.i
```

Voorbeeld `qwerty`:

```
1                                   ed1, wr1, ru1, pu3, yr1, td1, il1, ou2,
6                                   ur3, od2, rd1, wl1, eu1, ir3, tu1, yl1,
2                                   qd3, xr1, tu3, xl1, qu3, il3, ed1, wr1,
4                                   ru1, ou2, ul4, od2, rd1, wl1, eu1, ir3,
..qwwe                              qd3, wl2, ru1, pu1, xr3, td1, wl1, qu3,
..qr.e                              il1, ed3, xr1
xxqr.e
tiii..
tuuop.
yy.op.
```

Voorbeeld `rondje`: Autodraaikolkje.

```
1                                   zr1, qd3, dr6, eu1, xr9
6
2
10
..cddd...q..
a.ce.....q..
axxe.....q..
bbryyi......
.gruui......
.gtt.i...zz.
```

Voorbeeld `kannie`: Deze kan echt niet.

```
1                                   Geen oplossing gevonden
2
0
10
.xx......q..
.........q..
```