Article
Danny Wijnschenk · Nov 18, 2017 4m read

Advent of Code 2016 Day18: Like a Rogue

This is a series of programming challenges for beginners and experienced Caché programmers.

For an introduction : go to article https://community.intersystems.com/post/advent-code-2016-day1-no-time-ta...

You walk into a room, step on a tile and hear a loud click...

Traps !

The challenge is about avoiding traps.

You can recognise if a tile is a trap by following scheme :

The first row is a given : a safe tile is shown as . and a trap is shown as ^
For the next rows, the state of a tile is calculated related to the tiles in the previous row (same position, tile left and tile right).
If the tile is at an edge, the non-existing tile at the previous row outside the edge is considered safe.

This is how you can calculated if a tile is safe or trap while looking at the tiles on the previous row:

```
Its left and center tiles are traps, but its right tile is not.
Its center and right tiles are traps, but its left tile is not.
Only its left tile is a trap.
Only its right tile is a trap.
In any other situation, the new tile is safe.
```

As an example, here is how the room could look like after 10 rows :

```
.^^.^.^^^^
^^^...^..^
^.^^.^.^^.
..^^...^^^
.^^^^.^^.^
^^..^.^^..
^^^^..^^^.
^..^^^^.^^
.^^^..^.^^
^^.^^^..^^
```

In this case, there are 38 safe tiles.

You have to take the puzzle input, and calculate 40 rows (start row including). How many safe tiles are there ?

Go for a full explanation to http://adventofcode.com/2016/day/18, it contains your puzzle input to start.

The second part of the challenge is just a bigger room : instead of 40 rows, you have to calculate 400.000 rows.

The code we wrote for the first part can handle also the second part, so here are our solutions :
(remember, you are welcome to add your own solution to these articles !)

The routine that Bert wrote :

```
Start() PUBLIC {
  s input=
".^.^..^......^^^^^...^^^...^...^....^^.^...^.^^^^....^...^^.^^^...^^^^.^^.^.^^..^.^^
^..^^^^^^.^^^..^"
  s safeTiles=$LENGTH(input,".")-1
  s prevLine=input

  s nextLineId=2
  s goal=400000

  while goal>=nextLineId {
    s nextLine=""
    for i=1:1:$LENGTh(input) {
      if i=1 {
        s left="."
      } else {
        s left=$EXTRACT(prevLine,i-1)
      }
      s center=$EXTRACT(prevLine,i)
      if i=$LENGTH(input) {
        s right="."
      } else {
        s right=$EXTRACT(prevLine,i+1)
      }
      if $$isTrap(left,center,right) {
        s nextLine=nextLine_"^"
      } else {
        s nextLine=nextLine_"."
      }
    }
    s nextLineId=nextLineId+1
    s safeTiles=safeTiles+$LENGTH(nextLine,".")-1
    s prevLine=nextLine
  }
  w !,"safetiles: ",safeTiles
}

isTrap(Left,Center,Right) {
  s isTrap=0
  if ((Left="^") && (Center="^") && (Right=".")) {
    s isTrap=1
  }
  if ((Left=".") && (Center="^") && (Right="^")) {
    s isTrap=1
  }
  if ((Left="^") && (Center=".") && (Right=".")) {
    s isTrap=1
  }
  if ((Left=".") && (Center=".") && (Right="^")) {
    s isTrap=1
  }
  q isTrap
}
```

Here is my code (a bit shorter, but functionally the same, the question is alwasy if shorter code is better,

especially if you have to change it after a few years !)

```
Class AOC2016.Day18
{

ClassMethod Part(part As %Integer = 1)
{
  #Dim row as %String =
"^.^^^.^..^....^^....^^^^.^^.^...^^.^.^^.^^.^^..^.^...^.^..^.^^.^..^.....^^^.^.^^^..^
^...^^^...^...^."
  #Dim safe as %Integer = 0
  #Dim iInput, iRow, iCol as %Integer
  #Dim prevRow, Row as %String
  #Dim newTile as %Char
  #Dim maxRows as %Integer = $Select(part=1:40,1:400000)

  //don't forget to count the safe tiles on first row
  For iInput=1:1:$l(row) {
    If $Extract(row,iInput)="." set safe = safe + 1
  }

  For iRow=2:1:maxRows {
    Set prevRow = row
    Set row=""
    For iCol=1:1:$Length(prevRow) {
      Set newTile = $$TrapOrTile($S(iCol=1:".",1:"")_$E(prevRow,iCol-1,iCol+1)_$S(
iCol=$length(prevRow):".",1:""))
      Set row = row _ newTile
      If newTile = "." set safe = safe + 1
    }
    If part = 1 Write row,!
  }
  Write "Safe tiles : ",safe,!
  Quit

TrapOrTile(s) Quit $Case(s,"^^.":"^",".^^":"^","^..":"^","..^":"^",:".")

}
}
```

Look here for all our solutions so far : https://bitbucket.org/bertsarens/advent2016 and https://github.com/DannyWijnschenk/AdventOfCode2016

Here is the list of all Advent of Code 2016 articles :

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

#Caché #Code Snippet #Contest #ObjectScript

Source URL:https://community.intersystems.com/post/advent-code-2016-day18-rogue