Tic-tac-toe

Tic-tac-toe (also called *Xs and Os* or *noughts and crosses*) is a simple game for two players. It is played with paper and pencil on a 3×3 grid. Initially, all spaces in the grid are empty. One player marks spaces with crosses (here represented by the letter `X`) and the other player marks spaces with noughts (here represented by the letter `O`). Player in turn mark an empty space in the grid with their own symbol. The player who marks crosses always begins the game. The player who succeeds in placing three respective marks in a horizontal, vertical or diagonal row wins the game. If all places are filled up without any player winning, the game ends in a draw. The following game is won by the player marking crosses.

Players soon discovered that best play for both parties leads to a draw. Hence, tic-tac-toe is most often played by young children. Especially for beginners, the middle square is the most important square in the grid. In total there are 764 essentially different states of the grid, without taking into account the empty grid and without taking into account positions that are the same after rotation and/or mirroring the grid. There are 2096 valid ways players can mark empty spaces starting from the first mark and successive marks after that. As such, there are three possible opening marks: a mark in the middle, a mark in one of the corners and a mark in the middle of one of the borders.

Assignment

A given text file contains the configuration of a tic-tac-toe game that has just finished. The file contains three lines, each containing three characters. The possible characters are *i*) a space representing an empty space, *ii*) a letter `X` used as a marker by the first player, or *iii*) a letter `O` used as a marker by the second player. Your task is to determine which player has won the game. In order to do so, you need to define a class TicTacToe, that at least supports the following three methods:

- An initialization method that takes the location of a text file containing the configuration of a tic-tac-toe game. The method must read the information from the file, store it in the attribute `grid` of the newly created object. The attribute grid must refer to a list containing three strings, where each string contains the three characters from the corresponding line in the text file.
- An implementation of the built-in method `__str__` that returns a string representation of the grid of the tic-tac-toe game. This string representation must contain a formatted representation of the grid. The formatted string must contain three lines, representing the rows of the grid. The columns of the grid are separated by a single space in the formatted string. Empty space are represented by a dot (`.`).
- A method `winner` that returns who won the tic-tac-toe game. The method must return the string `X wins` if the player marking crosses has won the game, the string `O wins` if the player marking noughts has won the game, and the string `draw` if the game has ended in a draw.

Example

In the following interactive session we assume that the text files tictactoe1.txt, tictactoe2.txt and

tictactoe3.txt are located in the current directory.

```
>>> game = TicTacToe('tictactoe1.txt')
>>> game.grid
['OX ', 'OX ', ' X ']
>>> print(game)
O X .
O X .
. X .
>>> game.winner()
'X wins'

>>> game = TicTacToe('tictactoe2.txt')
>>> game.grid
['OOO', ' XX', 'X  ']
>>> print(game)
O O O
. X X
X . .
>>> game.winner()
'O wins'

>>> game = TicTacToe('tictactoe3.txt')
>>> game.grid
['XXO', 'OOX', 'XOX']
>>> print(game)
X X O
O O X
X O X
>>> game.winner()
'draw'
```

Boter-kaas-en-eieren (ook wel *oo maal oo* of *kruisje nulletje* genoemd) is een eenvoudig spel voor twee spelers. Het wordt met potlood en papier gespeeld op een 3×3 rooster. Bij aanvang zijn alle velden leeg. Een speler zet kruisjes (hier voorgesteld door de letter X) en de andere speler zet rondjes (hier voorgesteld door de letter O). De spelers zetten om beurt elk hun eigen symbool op één van de lege velden van het rooster. De speler die kruisjes zet mag altijd beginnen. Degene die drie van zijn eigen symbolen op een rij heeft (diagonaal, verticaal of horizontaal) heeft gewonnen. Als alle veldjes ingevuld zijn zonder dat één van de spelers wint, dan eindigt het spel in een gelijkspel. Het volgende spelletje werd bijvoorbeeld gewonnen door de speler die kruisjes zet.

Voor iemand die het spel goed genoeg kent, is het eenvoudig om ieder spel in een gelijkspel te laten eindigen, ongeacht wat de tegenstander doet. Zeker voor beginners is het veld in het midden het belangrijkste veld. Er zijn in totaal 764 mogelijke bordposities, zonder daarbij het lege bord mee te rekenen en zonder posities dubbel te rekenen die door roteren of spiegelen hetzelfde zijn. Er zijn in totaal 2096 zetten toegestaan vanaf de eerste zet en daarna tussen de opeenvolgende bordposities. Zo zijn er drie verschillende openingszetten mogelijk: in het midden, in een hoek of in het midden van een rand.

Opgave

Een gegeven tekstbestand bevat de configuratie van een spelletje boter-kaas-en-eieren dat net is afgelopen. Het bestand bestaat uit drie regels, die elk drie karakters bevatten. De mogelijke karakters zijn *i*) een spatie die staat voor een leeg veld, *ii*) een letter `X` die is neergezet door de eerste speler, of *iii*) een letter `O` die is neergezet door de tweede speler. Aan jou de vraag om te bepalen welke speler gewonnen heeft. Hiervoor definieer je een klasse `BoterKaasEieren`, die minstens de volgende drie methoden ondersteunt:

- Een initialisatiemethode waaraan de locatie van een tekstbestand met de configuratie van een spelletje boter-kaas-en-eieren moet doorgegeven worden. De methode moet de informatie uit het bestand inlezen, en opslaan in een attribuut `bord` van het nieuw aangemaakte object. Het attribuut `bord` moet hierbij verwijzen naar een lijst van strings, waarbij elke string bestaat uit de drie karakters uit de overeenkomstige regel in het gegeven tekstbestand.
- Een implementatie van de ingebouwde `__str__` methode die een stringvoorstelling teruggeeft van het rooster waarop het spelletje gespeeld wordt. Deze stringvoorstelling moet een opgemaakte weergave van het rooster voorstellen. Hiervoor moet de string bestaan uit drie regels, die de rijen van het rooster voorstellen. De kolommen van het rooster worden in de string van elkaar gescheiden door een spatie. Lege veldjes moeten voorgesteld worden door een punt (.).
- Een methode `winnaar` die de winnaar van het spelletje boter-kaas-en-eieren teruggeeft. De methode moet de string `X wint` teruggeven als de speler die kruisjes zet het spel heeft gewonnen, de string `O wint` als de speler die rondjes zet het spel heeft gewonnen, en de string `gelijkspel` als het spel geëindigd is op een gelijkspel.

Voorbeeld

Bij onderstaande voorbeeldsessie gaan we ervan uit dat de tekstbestanden tictactoe1.txt, tictactoe2.txt en tictactoe3.txt zich in de huidige directory bevinden.

```
>>> spel = BoterKaasEieren('tictactoe1.txt')
>>> spel.bord
['OX ', 'OX ', ' X ']
>>> print(spel)
O X .
O X .
. X .
>>> spel.winnaar()
'X wint'

>>> spel = BoterKaasEieren('tictactoe2.txt')
>>> spel.bord
['OOO', ' XX', 'X  ']
>>> print(spel)
O O O
. X X
X . .
>>> spel.winnaar()
'O wint'

>>> spel = BoterKaasEieren('tictactoe3.txt')
>>> spel.bord
['XXO', 'OOX', 'XOX']
>>> print(spel)
X X O
```

```
O O X
X O X
>>> spel.winnaar()
'gelijkspel'
```