

Luffarschack

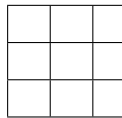
Programmering 1

2023/2024

1 Beskrivning

Luffarschack är ett spel som spelas av två personer på ett bräde med nio rutor — tre rader och tre kolumner — se figur 1. Båda spelarna har var sin uppsättning brickor, ofta antingen **o** eller **x**, och målet med spelet är att få tre av sina egna brickor i rad. Under spelets gång turas spelarna om med att lägga ut sina brickor antingen tills någon har vunnit eller tills spelplanen är full.

En spelare har vunnit när tre av dennes brickor formar en rät linje. Det kan vara tre i rad horisontellt, vertikalt eller diagonalt. Se figur 2 för ett exempel där spelare **x** har vunnit.



Figur 1: Ett 3×3 bräde

1.1 Varianter

Det finns i alla fall två till varianter på spelet. I både de andra varianterna av spelet så har de båda spelarna bara vars tre brickor de får lägga ut. I den första varianten så får man — efter att ha lagt ut alla sina tre brickor — plocka upp valfri av ens egna brickor för att lägga på en ny position. I den andra varianten så måste man plocka upp den bricka som legat på brädet längst. Båda varianterna slutar först när någon har vunnit.

X		O
O	X	
O	X	X

Figur 2: Ett exempel där **X** har vunnit

2 Uppgiften

Din uppgift är att programmera spelet *Luffarschack* med hjälp av den pseudokod som är angiven längre ner. Du kommer att ha fram till och med vecka 4 på dig att lösa uppgiften.

Utöver att implementera pseudokoden kommer du att behöva kommentera *varje* rad med kod i ditt program.

2.1 Syfte

Syftet med uppgiften är att vänja sig vid större projekt. Det här kommer att bli ett program på 86 rader kod, och framtida inlämningar kommer att vara minst så här stora. Det är även en övning i att hantera två-dimensionella listor, vilket också kommer att dyka upp i senare projekt, att kommentera sin kod är också ett viktigt steg för att förstå vad koden faktiskt gör. Ytterligare ett syfte med projektet är att öva på struktur och god programmeringssed. En strukturerad kod är väldigt användbar när du gör större projekt.

3 Pseudokod

```
1 function draw_board(board) :
2     print " ____"
3     for row in board:
4         print "|" + row[0] + row[1] + row[2] + "|"
5     print " ---"
6
7 function check_victory(board, player) :
8     for i := 0 to 3:
9         if row_victory(board, player, i) :
10             return True
11         if col_victory(board, player, i) :
12             return True
13     if dia_victory_1(board, player) :
14         return True
15     if dia_victory_2(board, player) :
16         return True
17     return False
18
19 function row_victory(board, player, row) :
20     return board[row][0] = board[row][1] = board[row][2] =
        player
21
22 function col_victory(board, player, col) :
23     return board[0][col] = board[1][col] = board[2][col] =
        player
24
25 function dia_victory_1(board, player) :
```

```

26     return board[0][0] = board[1][1] = board[2][2] = player
27
28 function dia_victory_2(board, player):
29     return board[2][0] = board[1][1] = board[0][2] = player
30
31 function create_board():
32     board := []
33     for i := 0 to 3:
34         board append []
35         for j := 0 to 3:
36             board append " "
37     return board
38
39 function take_position():
40     valid := False
41     while not valid:
42         print "Choose row: "
43         row := input
44         if row is integer and 1 <= row < 4:
45             valid := True
46     valid := False
47     while not valid:
48         print "Choose column: "
49         col := input
50         if col is integer and 1 <= col < 4:
51             valid := True
52     return (row-1, col-1)
53
54 board := create_board()
55 player := "X"
56 playing := True
57 turn := 0
58 draw := False
59
60 while playing:
61     turn := turn + 1
62     draw_board(board)
63     print "Player "+player+"'s turn"
64     row, col := take_position()
65     while board[row][col] != " ":
66         print "Position taken, choose again"
67         row, col := take_position()
68     board[row][col] := player
69     if check_victory(board, player):
70         playing := False
71     else:
72         if player = "X":
73             player := "O"
74         else:

```

```
75         player := "X"
76     if turn = 9:
77         draw := True
78         playing := False
79
80 draw_board(board)
81 if not draw:
82     print "Player "+player+" has won!"
83 else:
84     print("It's a draw")
85
86 input
```