



Tre på rad mot datam

Introduksjon

I dag skal vi prøve å skrive kode slik at datamaskinen kan spille tre på begynnelsen, men etterhvert som den lærer noen triks vil den kanskje

Steg 1: Vi fortsetter fra for

I leksjon 6 skrev vi et tre-på-rad spill for to spillere. Vi brukte *Tk* *lerret* oss se på hva vi allerede har før vi begynner å skrive ny kode.



Sjekkliste



Åpne IDLE. Åpne filen fra forrige leksjon og lagre den med et ny kopiere inn følgende:

```
from tkinter import *  
  
main = Tk()  
  
c = Canvas(main, width=600, height=600)  
c.pack()  
  
c.create_line(200, 0, 200, 600)  
c.create_line(400, 0, 400, 600)  
  
c.create_line(0, 200, 600, 200)  
c.create_line(0, 400, 600, 400)
```

```

grid = [
    "0", "1", "2",
    "3", "4", "5",
    "6", "7", "8",
]

def click(event):
    shape = choose_shape()
    across = int(c.canvasx(event.x) / 200)
    down = int(c.canvasy(event.y) / 200)
    square = across + (down * 3)

    if grid[square] == "X" or grid[square] == "0":
        return

    if winner():
        return

    if shape == "0":
        c.create_oval(across * 200, down * 200,
                      (across+1) * 200, (down+1) * 200)
        grid[square] = "0"
    else:
        c.create_line(across * 200, down * 200,
                      (across+1) * 200, (down+1) * 200)
        c.create_line(across * 200, (down+1) * 200,
                      (across+1) * 200, down * 200)
        grid[square] = "X"

def choose_shape():
    if grid.count("0") > grid.count("X"):
        return "X"
    else:
        return "0"

def winner():
    for across in range(3):
        row = across * 3

```

```

        line = grid[row] + grid[row+1] + grid[row+2]
        if line == "XXX" or line == "000":
            return True

    for down in range(3):
        line = grid[down] + grid[down+3] + grid[down+6]
        if line == "XXX" or line == "000":
            return True

    line = grid[0] + grid[4] + grid[8]
    if line == "XXX" or line == "000":
        return True

    line = grid[2] + grid[4] + grid[6]
    if line == "XXX" or line == "000":
        return True

c.bind("<Button-1>", click)

mainloop()

```

- ☐ Lagre, og kjør programmet, slik at du er sikker på at det virker!
Du skal kunne klikke i rutene for å plassere sirkler og kryss inntil
- ☐ Før vi begynner med dagens kode vil vi gjøre en liten oppryddning i prosedyren `click`. Vi flytter koden som tegner sirkler og med disse to prosedyrene:

```

def click(event):
    shape = choose_shape()
    across = int(c.canvasx(event.x) / 200)
    down = int(c.canvasy(event.y) / 200)
    square = across + (down * 3)

    if grid[square] == "X" or grid[square] == "0":
        return

```

```

    if winner():
        return

    grid[square] = shape
    draw_shape(shape, across, down)

def draw_shape(shape, across, down):
    if shape == "O":
        c.create_oval(across * 200, down * 200,
                      (across+1) * 200, (down+1) * 200)
    else:
        c.create_line(across * 200, down * 200,
                      (across+1) * 200, (down+1) * 200)
        c.create_line(across * 200, (down+1) * 200,
                      (across+1) * 200, down * 200)

```

Kjør koden og test at den fortsatt fungerer på samme måte som refaktorering. Vi har endret på selve koden, men ikke endret hva

Steg 2: Spill tilfeldig

Før vi kan lære datamaskinen hvordan den gjør gode trekk vil vi lære med å la datamaskinen finne en tilfeldig ledig rute, og deretter spille

Husk at vi har en variabel som heter `grid` som kan fortelle oss hvordan `["0", "1", "2", ...]`, hvor vi putter inn `"X"` og `"O"` etterhvert som denne listen for deretter å spille en slik rute.

Sjekkliste

- ☐ Vi vil først lage en ny prosedyre, `free_squares`, som kan finne ledige ruter i `grid`, men over linjen `c.bind(...)`:

```
def free_squares():
```

```

output = []
for position, square in enumerate(grid):
    if square != "X" and square != "0":
        output.append(position)
return output

```

Denne prosedyren lager en tom liste. Deretter går den gjennom Kommandoen `enumerate` kan fortelle oss posisjonen til hvert element om en liste `['A', 'B', 'C']` til parene `(0, 'A'), (1, 'B'), (2,`

- På toppen av filen vil vi importere `random`-biblioteket, som vi vil

```

from tkinter import *
import random

```

Du husker kanskje at vi brukte `random.choice` i en tidligere lekse.

- Nå skriver vi en prosedyre `play_move()` som kan spille i en tilfeldig `free_squares` men før linjen `c.bind(...)`

```

def play_move():
    moves = free_squares()
    square = random.choice(moves)

    across = square % 3
    down = square // 3

    grid[square] = "X"
    draw_shape("X", across, down)

```

Først bruker vi `free_squares` til å lage en liste over de tomme rutene. Vi vil nå oversette dette rutenummeret til rad- og kolonne-numr. La oss se litt nærmere på hvordan dette virker:

```

0 1 2
- - - -

```

```
0 | 0 1 2
1 | 3 4 5
2 | 6 7 8
```

For eksempel er rute nummer 5 i rad 1 og kolonne 2. Hvis vi deler $5 // 3$ er 1, $6 // 3$ er 2, og så videre. Operatoren `//` forteller oss bortsett fra resten. Siden vi har 3 kolonner forteller $5 // 3$ oss at $5 // 3$ er 1, $6 // 3$ er 2, og så videre. Operatoren `//` forteller oss bortsett fra resten. Siden vi har 3 kolonner forteller $5 // 3$ oss at $5 // 3$ er 1, $6 // 3$ er 2, og så videre. Operatoren `%` forteller oss hva resten er. $5 \% 3$ er 2, $6 \% 3$ er 0. Operatoren `%` forteller oss hva resten er. kolonnennummeret.

Legg merke til at de to linjene

```
across = square % 3
down = square // 3
```

gjør den motsatte utregningen av

```
square = across + (down * 3)
```

som vi allerede har brukt i `click`.



Til slutt endrer vi `click`-prosedyren slik at den kaller `play_move` deretter gjør datamaskinen sitt trekk.

```
def click(event):
    across = int(c.canvasx(event.x) / 200)
    down = int(c.cansasy(event.y) / 200)
    square = across + (down * 3)

    if grid[square] == "X" or grid[square] == "0":
        return

    if winner():
        return

    grid[square] = "0"
    draw_shape("0", across, down)
```

```
if winner():  
    return  
  
play_move()
```

Vi sjekker først om spilleren har vunnet, og hvis ikke lar vi datar

- ☐ Lagre programmet og kjør det. Datamaskinen vil nå trekke etter gjør tilfeldige trekk.

Steg 3: Velg et trekk som v

Datamaskinen spiller nå tre på rad, men den er ikke spesielt flink. La den velge trekk tilfeldig, la datamaskinen velge trekk som gjør at den vinner om de fire trekkene til datamaskinen, og om ett av disse vil vinne spillet lar vi datamaskinen

Sjekkliste

- ☐ Endre prosedyren `winner` slik at den tar et argument `grid`:

```
def winner(grid):  
    for across in range(3):  
        row = across * 3  
        line = grid[row] + grid[row+1] + grid[row+2]  
        if line == "XXX" or line == "000":  
            return True  
  
    for down in range(3):  
        line = grid[down] + grid[down+3] + grid[down+6]  
        if line == "XXX" or line == "000":  
            return True  
  
    line = grid[0] + grid[4] + grid[8]
```

```
if line == "XXX" or line == "000":  
    return True  
  
line = grid[2] + grid[4] + grid[6]  
if line == "XXX" or line == "000":  
    return True
```

Du trenger bare å endre den første linjen i prosedyren. Dette be-
stedet for `grid` som husker hvordan dette spillet ser ut. Dermed
spilt enda.

- ☐ Nå må vi forandre `click` så den sender inn riktig liste.

```
def click(event):  
    across = int(c.canvasx(event.x) / 200)  
    down = int(c.canvasy(event.y) / 200)  
    square = across + (down * 3)  
  
    if grid[square] == "X" or grid[square] == "0":  
        return  
  
    if winner(grid):  
        return  
  
    grid[square] = "0"  
    draw_shape("0", across, down)  
  
    if winner(grid):  
        return  
  
    play_move()
```

Alle steder vi har `winner()` i koden bytter vi det ut med `winner`

- ☐ Kjør koden, den skal fortsatt virke akkurat som før for vi har end-
- ☐ La oss hjelpe datamaskinen ved å legge til noen linjer i `play_mo`


```
def play_move():
    moves = free_squares()
    square = random.choice(moves)

    # Bruk et vinnende trekk om det eksisterer
    for possible in moves:
        new_grid = list(grid)
        new_grid[possible] = "X"
        if winner(new_grid):
            square = possible
            break

    across = square % 3
    down = square // 3

    grid[square] = "X"
    draw_shape("X", across, down)
```

For hver ledige rute lager vi en kopi av `grid`-listen med komma ledige ruten og bruker `winner` for å undersøke om dette vil være

- ☐ Kjør programmet ditt og test det flere ganger. Datamaskinen sk

Steg 4: Velg et trekk som k

Den andre strategien vi vil lære datamaskinen er å blokkere trekk som måte, men nå ser vi hva som skjer om vi plasserer ut O i de ledige rut

Sjekkliste

- ☐ Legg til litt mer kode i `play_move` som blokkerer trekk som gjør

```
def play_move():
```

```

moves = free_squares()
square = random.choice(moves)

# Bruk et blokkerende trekk om det eksisterer
for possible in moves:
    new_grid = list(grid)
    new_grid[possible] = "O"
    if winner(new_grid):
        square = possible
        break

# Bruk et vinnende trekk om det eksisterer
for possible in moves:
    new_grid = list(grid)
    new_grid[possible] = "X"
    if winner(new_grid):
        square = possible
        break

across = square % 3
down = square // 3

grid[square] = "X"
draw_shape("X", across, down)

```

Legg merke til at datamaskinen først plukker en tilfeldig ledig ruten, så bestemmer den seg. Til slutt sjekker den om den bestemmer seg en gang til!



Kjør koden og se om du klarer å vinne mot datamaskinen! Det h

Hele programmet

Det ferdige programmet ditt vil nå se omtrent ut som dette!

```
from tkinter import *
import random

main = Tk()

c = Canvas(main, width=600, height=600)
c.pack()

c.create_line(200, 0, 200, 600)
c.create_line(400, 0, 400, 600)

c.create_line(0, 200, 600, 200)
c.create_line(0, 400, 600, 400)

grid = [
    "0", "1", "2",
    "3", "4", "5",
    "6", "7", "8",
]

def click(event):
    across = int(c.canvasx(event.x) / 200)
    down = int(c.canvasy(event.y) / 200)
    square = across + (down * 3)

    if grid[square] == "X" or grid[square] == "0":
        return

    if winner(grid):
        return

    grid[square] = "0"
    draw_shape("0", across, down)

    if winner(grid):
        return

    play_move()
```

```

def draw_shape(shape, across, down):
    if shape == "O":
        c.create_oval(across * 200, down * 200,
                      (across+1) * 200, (down+1) * 200)
    else:
        c.create_line(across * 200, down * 200,
                      (across+1) * 200, (down+1) * 200)
        c.create_line(across * 200, (down+1) * 200,
                      (across+1) * 200, down * 200)

def winner(grid):
    for across in range(3):
        row = across * 3
        line = grid[row] + grid[row+1] + grid[row+2]
        if line == "XXX" or line == "000":
            return True

    for down in range(3):
        line = grid[down] + grid[down+3] + grid[down+6]
        if line == "XXX" or line == "000":
            return True

    line = grid[0] + grid[4] + grid[8]
    if line == "XXX" or line == "000":
        return True

    line = grid[2] + grid[4] + grid[6]
    if line == "XXX" or line == "000":
        return True

def free_squares():
    output = []
    for position, square in enumerate(grid):
        if square != "X" and square != "O":
            output.append(position)
    return output

def play_move():

```

```

moves = free_squares()
square = random.choice(moves)

# Bruk et blokkerende trekk om det eksisterer
for possible in moves:
    new_grid = list(grid)
    new_grid[possible] = "O"
    if winner(new_grid):
        square = possible
        break

# Bruk et vinnende trekk om det eksisterer
for possible in moves:
    new_grid = list(grid)
    new_grid[possible] = "X"
    if winner(new_grid):
        square = possible
        break

down = square // 3
across = square % 3

grid[square] = "X"
draw_shape("X", across, down)

c.bind("<Button-1>", click)

mainloop()

```

Utfordring

Det er fortsatt mulig å vinne mot datamaskinen. Kan du gjøre endringer slik at du kan lære datamaskinen å spille perfekt?

Lisens: [Code Club World Limited Terms of Service](#) **Forfatter:** O
Øversetter: Geir Arne Hjelle